FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Man-Machine Symbiosis - EEG and other biosignal man-machine interaction

Vitor Daniel Veigas Minhoto

Mestrado Integrado em Bioengenharia

Supervisor: João Paulo Trigueiros da Silva Cunha

July 22, 2019

© Vitor Daniel Veigas Minhoto, 2019

Man-Machine Symbiosis - EEG and other biosignal man-machine interaction

Vitor Daniel Veigas Minhoto

Mestrado Integrado em Bioengenharia

July 22, 2019

Resumo

Ao longo das últimas décadas, veículos aéreos não tripulados (VAT), drones, têm crescido bastante em popularidade. Estas máquinas são versáteis o suficiente para, num momento, serem usados como assistentes em missão de busca e salvamento, e no outro serem apenas veículos recreacionais. No entanto, normalmente, estas máquinas operam por controlo remoto de maneira autónoma, não havendo uma interação natural entre o drone e o humano. O principal objetivo deste trabalho era, então, desenvolver um módulo de inteligência que permita ao VAT ter a habilidade de sentir o humano e comunicar com ele tanto explícita como implicitamente através de biosinais, alcançando, então, o patamar de simbiose.

Para o módulo de comunicação explícita, o humano tem a habilidade de mandar comandos para o drone servindo-se uma Interface Cérebro Maquina usando um Wearable EEG. O drone pode implicitamente aperceber-se da presença do seu humano e do seu estado fisiológico através de um fluxo de dados ECG vindos do humano. No entanto esta comunicação deve funcionar de maneira bidireccional. O drone pode comunicar com o humano através de som ou fazendo formas no ar enquanto voo.

Durante a realização deste trabalho, para além da implementação destes módulos, um simulador foto-realista foi construído, permitindo tanto o humano como o drone treinar e desenvolver este sistema de interação dentro da simulação e depois trazer este conhecimento para o mundo real.

Uma prova de conceito do sistema foi desenvolvida e testada no evento "Joint Interagency Field Experimentation" promovido pelo Departamento da defesa e Departamento da marinha americana. A prova foi um sucesso tendo ganho um prémio de "Develop now".

Abstract

Over the past few decades, Unmanned Aerial Vehicle has risen both in popularity and in use. These machines are versatile enough to, in one moment, be used as an assistant in search and rescue missions, and the other be merely recreational. However, usually, these drones operate either by the use of remote control or autonomously, lacking a more natural interaction with the human.

The main goal of this dissertation was to add an augmented intelligent component to the drone, giving it the ability to sense a human user and interact with him both explicitly or implicitly, using biosignals, therefore achieving Symbiosis.

For the explicit communication part, the human can issue commands to a drone with a Brain Computer Interface by using an EEG wearable. The drone can also implicitly perceive who is its human by making use of a streaming ECG data from its human. However, communication should work both ways! The drone can also communicate with its human by sound or follow paths in the sky while flying.

Throughout this work, these modules were implemented as well as a photo-realistic simulation framework where both humans and drones can further develop these Interactions without the fear of crashing, and then transfer all learning developed to the real world.

A proof of concept was developed and tested at the US Department of Defence (DoD) & Department of Navy "Joint Interagency Field Experimentation" event, hosted by the Naval Postgraduate School in Camp Roberts, California, amusing the other participants and earning a *Develop Now* token.

Agradecimentos

Chegámos ao fim. E que viagem. Quando olho para trás e relembro a quantidade de pessoas que me fizeram o que sou hoje é com grande alegria que agradeço pela amizade e companhia ao longo destes anos.

Começo por agradecer aos meu 2 mais antigos amigos, Fernando e Francisco, por me terem acompanhado desde que éramos crianças até ao dia de hoje.

Agradeço também a todas as pessoas da SPRU que nos primeiros anos que vive cá no Porto, nunca me fizeram sentir sozinho. Deixo um abraço e um beijinho especial ao gang do 247: Pedro, Daniela, Maria, Ricardo e mais tarde as lituanas!

Deixo também um agradecimento aos membros do IEEE UP SB por todos os momentos na J201 e pelas amizades e oportunidades que me proporcionaram. Um especial ao carinho à crew do SYP por me terem dado uma das semanas mais cansativas, mas divertidas ao longo destes anos. Que continuemos a crashar maquinas Andante!

Deixo um agradecimento especial às pessoas com quem convivo e interajo mais diariamente estando sempre lá pra quando for preciso: Luísa, Hugo, Moreira, João Mota, Filipe, Gil, Tiago e Paulo.

A todo o grupo BRAIN por este último ano que passei por lá. Ao Duarte por sempre ajudar se alguma coisa for precisa, ao pessoal do -1 Bino e Ana pelas gargalhadas ao final do dia. Agradeço ainda ao Professor João Paulo Cunha por todas as oportunidades ao longo deste ano letivo, por sempre questionar, motivar a discussão e por ir mais além sempre em busca de desafios novos.

Um agradecimento muito especial à minha família! Aos meus avós por me ensinarem sempre algo novo, ao meu pai por estar sempre lá e especialmente à minha Mãe pelo apoio, por acreditar sempre em mim em me imbuir o espírito questionar e acreditar que sempre consigo ser e fazer melhor!

Por fim agradeço à minha namorada Inês por todo a companhia, suporte, carinho e amor ao longo destes últimos anos. Que venham muitos mais!

Vitor Daniel Veigas Minhoto

“The only true wisdom
is in knowing you know nothing.”

Socrates

Contents

List of Figures

List of Tables

Abreviaturas e Símbolos

AEP	Auditory Evoked Potentials
AI	Artificial Intelligence
API	Application programming interface
BCI	Brain Computer Interface
BLE	Bluetooth Low Energy
CACTF	Combined Armed Collective Training Facility
CMU	Carnegie Mellon University:
CMU-SV	Carnegie Mellon University - Silicon Valley
CPS	Common Spatial Patterns
DTG	Digital Fourier Transform
DOD	Deportment of Defense
DWT	Discrete Wavelet Transform
ECG	Electrocardiography
EEG	Electroencephalography
EMG	Electromyography
EOG	Electrooculogram
ERD	Event Related De-Synchronization
ERS	Event Related Synchronization
FAA	Federal Aviation Administration
FBCSP	Filter Bank Common Spatial Pattern
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FP	False Positive
GPS	The Global Positioning System
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HOC	Higer Order Crossing
IEEE	Institute of Electrical and Electronics Engineers
ITR	Information Transfer Rate
JIFX	Joint Interagency Field Experimentation
KNN	K-Nearest Neighbors
LDA	Linear Discriminate Analysis
lsl	Lab Streaming Layer
MEG	Magnetoencephalography
MIBIF	Mutual Information Best Individual Features
ML	Machine Learning

MNE	MEG+EEG Analysis and Visualization Toolbox
NZC	Number of Zero Crossings
OVR	One versus Rest
PTBD	
RF	Radio Frequency
RGB	Red, Green and Blue
SVM	Support Vector Machine
TCP/IP	Transmission Control Protocol/Internet Protocol
TN	True Negative
TP	True Positive
TP	True Positive
UAV	Unmanned Aerial Vehicle
VEP	Visual Evoked Potential
VIPER	Virtual Image Processing Environment for Research

Chapter 1

Introduction

1.1 Man-Machine Symbiosis

The concept of "Man-Machine Symbiosis" was introduced in 1960 by J.C.R. Licklider [1], at a paper with the same title in IRE Transactions on Human Factors in Electronics, now IEEE Transactions on Human-Machine Systems and IEEE Transactions on Systems, Man and Cybernetics. The core idea was that intelligent technology should augment the human factor instead of replacing it. This idea pursues the concept of "Symbiosis", that is defined as "Interaction between two different organisms living in a close physical association, typically to the advantage of both" by the Oxford dictionary[1]. However, this idea did not have many followers as user interfaces raised in popularity and the technology at the time did not have the capacity to perform the needed computations.

This dissertation aims to explore this concept and develop it in an already existing partnership with CMU-SV (Carnegie Mellon University Silicon Valley). Due to the extensive work in UAVs (Unmanned Aerial Vehicles) by this institution, the target machines were drones (more exactly quadcopters), although almost all results obtained in this work can be replicated with almost any Machine.

1.2 UAVs: a Brief History

UAVs already exist since the 1950s for military purposes, but it was not until 2006 that the FAA (Federal Aviation Administration) started issuing commercial drone licenses [2].

European market research from 2018 [3], made by Drone Industry Insights, evaluated the current industrial application of drones and its potential for growth. The data shows that almost 8 in 10 drones are being used for surveying activities and 33% for surveillance and monitoring. According to hardware and software manufacturers, police and emergency services has the potential for 26% of growth rate.

Due to the low price of this kind of vehicles in the last decade, several studies have been developed in order to make these machines as autonomous as possible with systems using computer

[1] https://www.lexico.com/en/definition/symbiosis

vision [4, 5, 6], environmental sensors [7, 8] or behaving as a swarm [9, 10] in applications like surveillance, landing on a target, environmental monitoring, disaster management and mission planning.

1.3 Motivation

1.3.1 UAVs in Search and Rescue Scenarios

There are many types of natural and human-made disasters such as earthquakes, floods, wildfires, hurricanes, radiological emergencies, explosions, among others that cause different disaster areas like collapsed buildings, landslide or crater. In these situations, search and rescue operations are often characterized by a similar set of constraints: time is critical, and any delay can result in dramatic consequences – potentially human losses; severe injuries requiring extensive treatments, increased risk of communicable diseases, damage to the health facilities, damage to the water systems, food shortage, population movements [11]. A recent report from the World Health Organization [12] states that during the decade 2001–2010, an average of 700 disasters (floods, earthquakes and others) were reported on an annual basis. These disasters affected around 270 million people. Deaths reported exceeded 130,000 (13,000 per year).

Adding UAVs to help officers on these scenarios, and some cases have already been used and documented in the literature. In [13] this vehicles are used to locate victims. In [11] they use infrared cameras to locate humans in the water. In [14], they help officials with path planning and scouting. In these studies, UAVs have a major import role as they can reduce time by locating key targets or finding routes, therefore improving the success rate of this kind of missions.

1.3.2 Innovation

The common characteristic in the examples shown in the previous section is that although UAVs do possess some intelligent aspect, they do not interact directly with the users on the field. Usually, there is one user that can control the drone and assess the information given using a GUI (Graphical User Interface).

This "human in the middle' making the bridge between the vehicle and the first responders on the field creates a bottleneck of information that can lead to delays, loss of information and generates a dependency on the judging of an external agent. By adding to the drone an augmented intelligence component, generating a passive (or implicit) communication, using biosignals, in addition to having a method for explicitly communicating with the drone a more natural interaction between both can be developed. This idea goes back to the thesis of J.C.R. Licklider [1] as both actors can learn how to cooperate in different situations of the target scenarios resulting in the fulfilment of the concept of "Man-Machine Symbiosis". A quick search on this term in Google Academics search and on IEEE eXplore shows little or none activity in this theme, confirming its novelty.

1.4 Objectives

The main objective of this dissertation is to design, implement and study the impact of a system that is capable to have a symbiotic interaction between a human user and a machine using biosignals captured from wearable devices. This main goal was to divide it into smaller milestones listed below:

1. Design a Conceptual System Architecture for the problem at hands.
2. Implement a bio-identification system in order for the user to authenticate and be identified by the machine (based on previous papers and patents of the group).
3. Compare several EEG wearable devices in order to select the best for real-world use.
4. Develop a brain-computer interface to send commands to the machine explicitly.
5. Implement a proof of concept (with a UAV) to be tested in the JIFX field trial in conjunction with Carnegie Mellon University - Silicon Valley, held between the 29th of April and the 3rd of May in Camp Roberts, California, USA.

1.5 Achievements

The achievements completed during this dissertation are as follows:

1. Designed a flexible System Architecture that can be built upon adding modules that were not approached during this work (more information on Chapter 2).
2. Re-implemented a previously developed ECG based bio-identification module [15], presenting a proof of concept demonstration.
3. Performed a comparison between four wearable EEG systems.
4. Created a protocol and dataset on EEG from motor movement and imagery captured from wearable devices.
5. Developed a brain-computer interface using motor movement.
6. Traveled to Camp Roberts, California, USA to attend JIFX event and present a proof of concept of a Human-UAV symbiotic system receiving a *Develop now* token from the event organization (Naval Postgraduate School).
7. Developed a photo-realistic environment where a real-world scenario can be simulated for virtual learning environment. In this simulator, one can build a level where a real-world location can be replicated, any number of actors (humans or UAVs) can be simulated and inter-interact with each other. The goal of the system is to develop, train and test the drone augmented intelligence component and then transfer this knowledge to the real world, saving both time and resources.

8. Assisted INESC-TEC's BRAINLab in writing a CMU-Portugal proposal named **Man-UAV Symbiosis** – New Models of Man-Autonomous Vehicle Implicit Interaction for Search & Rescue Scenarios. The other consortium partners are Connect Robotics Sistemas Autónomos LDA, Instituto de telecomunicações and CMU-SV

9. Developed a multi-platform real-time web-based data visualization tool for data streams sent by Bluetooth Low Energy (BLE) devices using the Web Bluetooth API.

1.6 Structure

Besides the Introduction, this document has seven more chapters.

Chapter 2 presents the conceptual Architecture of the proposed symbiotic system with a detailed description of each subpart that composes it.

Chapter 3 is a review of State of the Art on techniques that perform biometry using ECG, brain-computer interfaces using EEG, ECG as a biometry system and simulation environments for UAV.

Chapter 4 exposes some machine learning and data handling methods used in biosignals classification.

Chapter 5 presents the algorithm used, and results on an EEG Brain-computer interface for explicit interaction.

Chapter 6 shows the algorithm used and the results of an ECG bio-identification module for implicit interaction.

Chapter 7 is a description of the proof of concept developed for the JFIX event and a report of the event itself.

Chapter 8 describes the simulator, presenting its architecture and uses with real events recreation examples.

Chapter 9 presents the conclusion and future work.

Chapter 2

Man-UAV Symbiosis Conceptual System Architecture

This chapter aims to give a birds-eye view of the Man-UAV Symbiosis Conceptual System Architecture. It will follow a top-down approach where the overall system is presented, and then each sub-part is described in more detail.

2.1 System Requirements

In order to develop the overall system, the first step was to identify its requirements. In order to do this, an Agile approach was employed.

Agile software development is a group of software development methodologies based on iterative development. Agile methods or processes do promote a disciplined project management process that encourages regular inspection and adaptation, being a philosophy that encourages teamwork, self-organization and accountability [16].

Although there are several agile methodologies used nowadays, e.g. Agile Scrum Methodology, Lean Software Development, Kanban, Extreme Programming (XP),Crystal, Dynamic Systems Development Method, Feature Driven Development, all of them start with the same step: the definition of the requirements and features to implement by developing user stories. During this work none specific methodologies used was employed, but this first step was adopted as it helps to shift the focus from writing about requirements to talking about them. These user stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability: **As a < *type of user* >, I want < *some goal* > so that < *some reason* >.)** [17].

A user is an actor in our system. Each can collect different types of information using their sensing capabilities and transmit only relevant pieces to its pair. In our architecture, we aim to have two types of actors: a Human and a Drone. The human user can be generalized to any person, from an everyday user that has a "pet drone" to a search and rescue officer performing a mission, however, in this work, we will focus mainly on search and rescue scenarios due to being a goal for the JFIX field test.

User Story 1 As a *Human user*, I want *to explicitly communicate with the drone* so that *I can issue him commands.*

User Story 2 As a *Human user*, I want *to be as mobile as possible* so that *I can freely perform my mission.*

User Story 3 As a *Human user*, I want *the drone to transmit me useful information* so that *I can better perform my mission.*

User Story 4 As a *Drone*, I want *to patrol the field* so that *I can inform my Human of what to expect.*

User Story 5 As a *Drone*, I want *to have information on my human condition* so that *I can help him if needed.*

User Story 6 As a *Drone*, I want *to quickly and easily communicate with my Human* so that *I can inform him if something catches my attention.*

User Story 7 As a *Drone*, I want *to be in contact with other humans* so that *if something goes wrong with my Human I can call for help.*

User Story 1 states that the Human must be able to communicate with the drone and **User Story 3** imposes that the user wants to retrieve information from the drone. From this, it can already be issued that two major components need to be further analyzed:

1. The actors: What roles do they have in the system and what data can they collect?
2. Interaction: How can the two actors interact and transmit information?

2.2 Actors

2.2.1 Human User

In search and rescue missions, the main role of the human user is to help people that might be trapped or injured: mainly in obstructed, damaged and hard to access areas. These are not only a danger to the victims but also agents on the field. With this in mind, continuously monitoring vitals and position would be beneficial and could prevent losing officers on the field. However, as required by **User Story 2**, these measurements should be the less intrusive way possible in order for this actor to maintain its mobility and effectiveness.

With this in mind, we must consider that our system must be ready to support any number of time-series data that translates any information related to the Human (ECG, EEG, GPS, or others) measured with non-intrusive wearable devices. However, this data should not be sent as it is collected to the UAV, so the Human should carry a module that aggregates the data and computes essential metrics from low-level features like Heart Rate to higher level features like commands from a Brain Computer Interface.

2.2.2 UAV

UAVs role in classic search and rescue scenario is to patrol between key points (User Story 4) and offer support to agents on the field by scouting terrain or finding victims. To do this, the drone is usually equipped with one or more sensing units, mainly cameras such as RGB or infrared in addition to the standard incorporated telemetry information. Moreover, our system should be ready to incorporate any other sensors to be placed on the drone.

Similar to the Human element, a unit that joins all sensor data (such as environment sensing), transform it into meaningful features and transmits it to the other agent when relevant is essential.

2.3 Interactions

2.3.1 Explicit Interaction

Explicit Interaction is when an consciously sends a command or information to the other element.

Examples on Human to drone interaction might be through gestures, movement, sound, GUIs, BCIs (Brain-computer Interfaces) or any other human-machine interface.

2.3.2 Implicit Interaction

Implicit Interaction is a key part of the symbiosis and where the true novelty of the work is. This concept resumes to an actor perceiving another actor intention or estate without the least performing a direct communication.

An example of this kind of interaction (on the human side) would be the human perceiving that the drone found something of interest because it lowered its altitude and started to rotate around the object of interest.

As for the drone, the most prominent example is **User Story 5**, where the UAV is continuously monitoring the Human's vitals and checking for any anomaly.

Achieving this back and forth interactions (both Explicit and Implicit) raises Human-machine interaction to a level never before seen where both parts can genuinely interact and learn from each other for the benefit of both, assuming a real symbiosis.

2.4 System Components and Information Flows

Figure 2.1 shows the proposed system architecture for the requirements identified. A Human user must carry a processing module that can aggregate and process any number of time-series data acquired by the sensing units (wearable devices). On this module, relevant features are computed using signal processing and/or machine learning. These results are then sent to the communications module to be transmitted to the drone wirelessly. The processing and communication unit can be on the same device, for example, integrated into a smartphone or a microcomputer, such as *raspberry pie*[1].

On the drone side, the information is received by the communications module and passed to a processing unit responsible for decoding and deciding what to do with that information (Implicit communication). This unit should have the capacity to learn in order for the drone to adapt to its Human. These two units can also be integrated on the same device and will form the augmented intelligent module described in prior sections. These devices should have sufficient computation power to perform all needed computations, for example, a board with integrated GPU like the *Nvidia Jetson*[2]. Note that the drone can and should have more than one processing unit in order to isolate the software that handles the drones internal mechanics such as stabilization and movement from this augmented intelligent system. The UAV also has several integrated sensors streaming data that might need processing. These data streams should be addressed by the *Jetson* as it might be useful for the decisions to take by the Augmented intelligence module.

In this dissertation, a particular case of this architecture is proposed, involving two sensing systems for the human (EEG and ECG wearable systems) and none for the drone. The EEG signal will allow explicit communication to the drone as a Brain Computer Interface. The ECG signal will serve as a way for the drone to evaluate the condition of its Human, react to specific conditions, e.g. stress or fall detection, as well as continually identifying the user (implicit communication).

[1] https://www.raspberrypi.org/

[2] https://developer.nvidia.com/buy-jetson

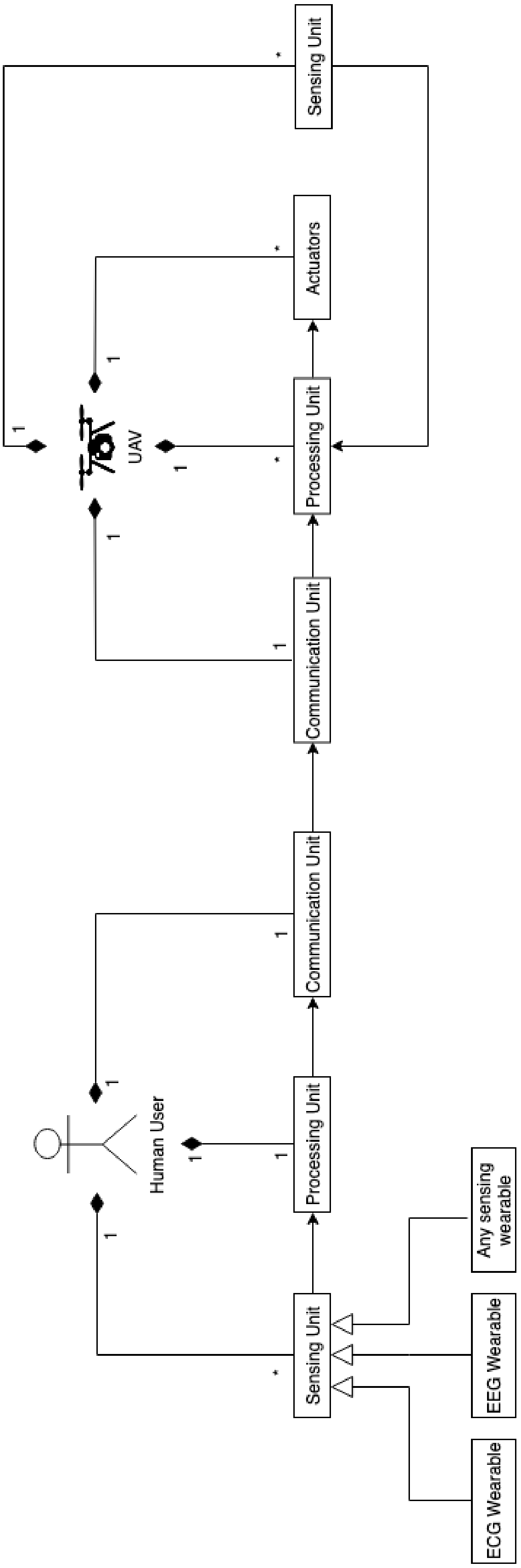


Figure 2.1: Man-UAV Symbiosis Abstract Arquitecture: Class Diagram.

Chapter 3

State of The Art

3.1 EEG and Brain-Computer Interfaces

This section focuses on EEG signal and its use in BCI. It describes its characteristics, acquisition method, paradigms, available datasets, main computed features and results reported in the literature.

3.1.1 EEG Signal: Characteristics

Surface EEG is a fast and easy way to acquire and record the electrical activity of the brain with easy to use, non-intrusive electrodes. The captured signal is the resultant electrical current of the joint polarization and depolarization of million of the underlying neurons in the several layers of the brain [19].

A typical healthy clinical acquired EEG can be seen in Figure 3.1. The amplitude range for this signal is usually between 5-300 μV and the useful information is on the 0 to 150 Hz frequency range.[19]

It can be decomposed in several frequency bands, Figure 3.2, being each one usually measured in a specific brain area and produced by or resulting from different psycho-physiological states, allowing an expert to take some conclusions or perform a diagnostic on a subject cognitive state by observing these specific characteristics on the signal. [20].

The five frequency bands are:

1. Delta Waves (1 – 4 Hz): Mainly studied in sleep laboratories, delta waves are analyzed to evaluate the depth of sleep. The stronger the wave rhythm, the deeper the sleep. An increase in delta power is also associated with increased concentration.[21]

2. Theta Waves(4 – 7 Hz) – Theta is associated with a wide range of cognitive processing from memory encoding and retrieval to cognitive workload [21]. Whenever a patient is confronted with a difficult task, these waves become more prominent. They are also associated with increased fatigue levels.[22].

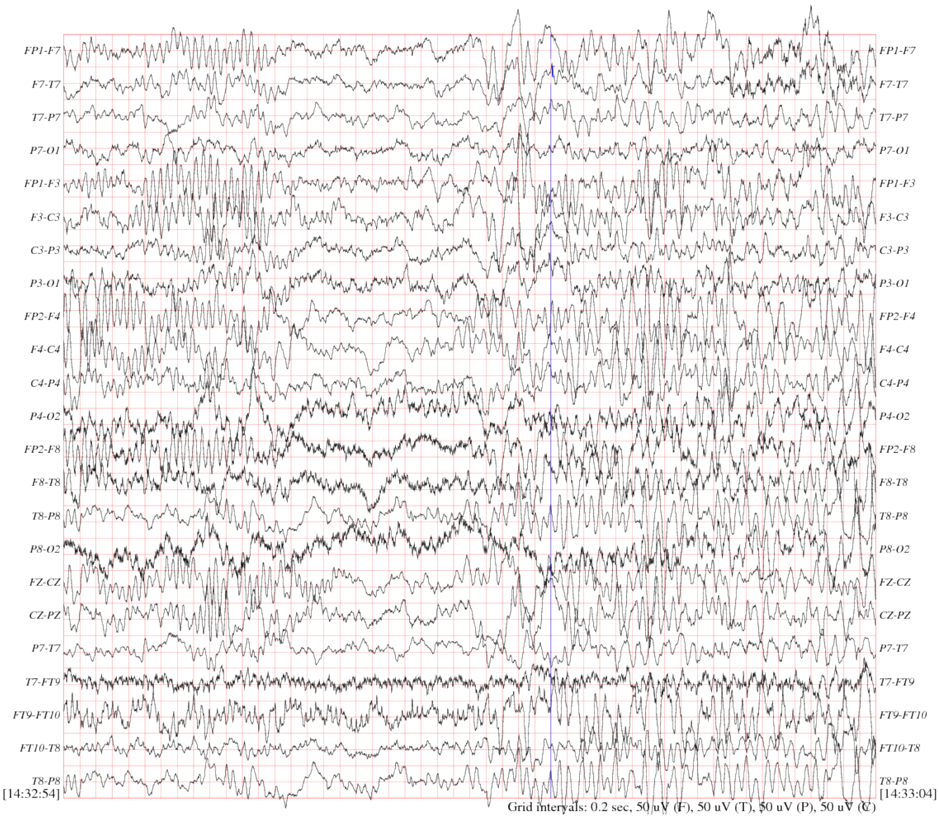


Figure 3.1: Clinical EEG Signal.[18]

3. Alpha Waves(7 – 12 Hz) – Alpha levels are the highest when in a state of relaxed wakefulness, being frequently used to monitor relaxation in *Biofeedback* training and also linked to inhibition and attention. They usually appear when the subject closes its eyes [23].

4. Beta Waves(12 – 30 Hz)- Normally observed on motor regions, beta frequencies become stronger as planning or executing movements with any body part. [24].

5. Gamma Waves(>30 Hz, typically 40 Hz) – Researchers are yet to reach an agreement on Gamma frequencies. Some argue that they reflect attentive focusing and serves as carrier frequency facilitating data exchange between brain regions[25], while others associate gamma with rapid eye movements, necessary for sensory processing and information uptake. [26].

3.1.2 Electrode location system

EEG acquisition, for clinical use, does have a pre-set of electrode placement rules in order for it to be invariant from center to center.

The first system proposed was the *10-20 system* by Jasper in 1958 [28], Figure 3.3. This system was the first to set a standard location and nomenclature for each of the 21 electrodes placed on the scalp. Using reference points such as the nasion, preauricular keypoints and inion, the head is

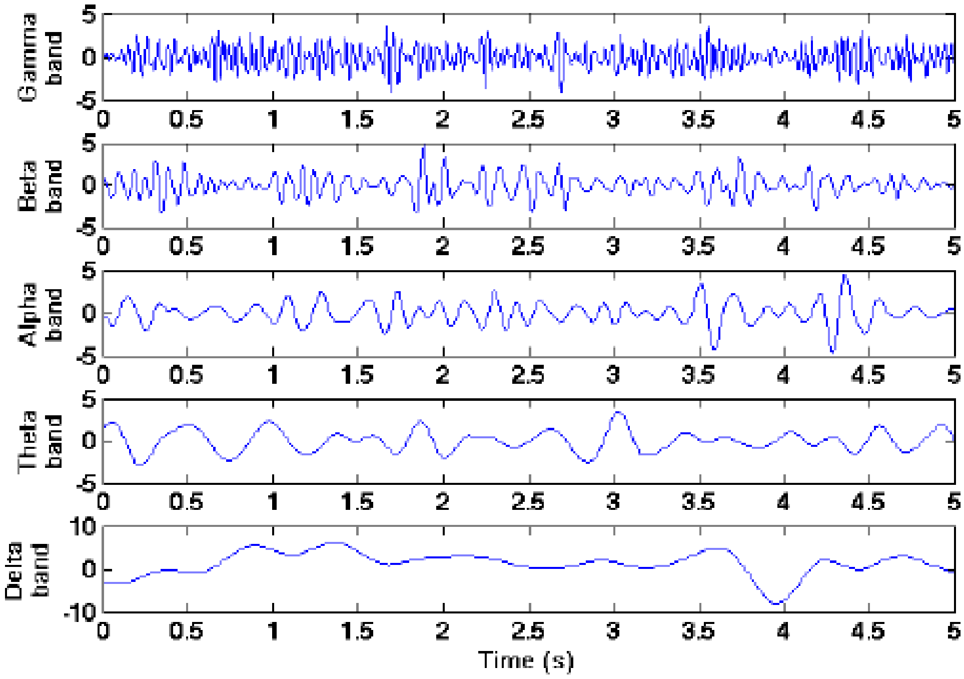


Figure 3.2: EEG frequency bands.[27]

divided into proportional positions to provide adequate coverage of all the brain regions. The name of each electrode consists of a letter and a number. The letter refers to the region of the brain where the electrode is positioned (F: frontal, C: central, T: temporal, P: posterior, and O: occipital), and the number is related to the cerebral hemisphere with even numbers in the right hemisphere, and odd numbers in the left as seen in Figure 3.3.

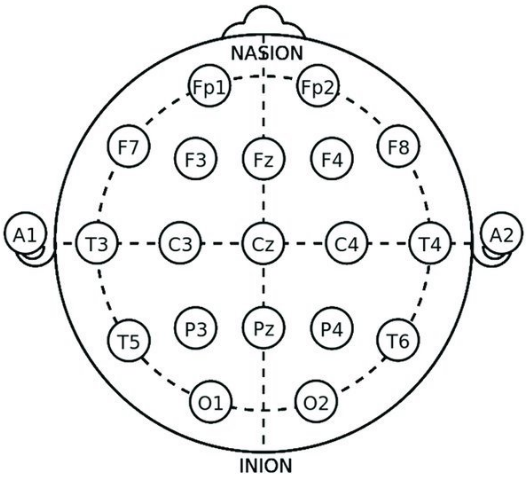


Figure 3.3: 10-20 System. [29]

In 1985, an extension to the original *10-20 system* was proposed involving an increase in the number of electrodes from 21 to 74 [30, 31, 29]. The *10-20 EEG electrode placement system* is mostly used in clinical application, while the *10-10 system*, represented by Figure 3.4 is mostly used in research.

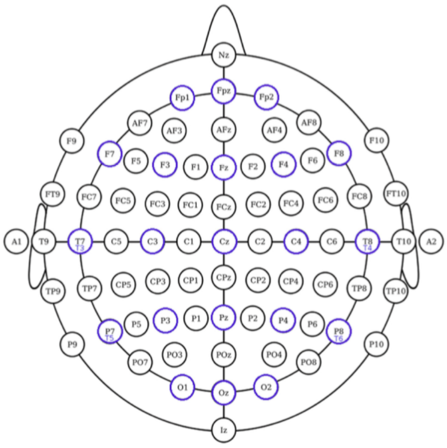


Figure 3.4: The 10-10 International system of EEG electrode placement. Blue circles represents the location of 10-20 EEG electrodes. Nodes T8, T7, P8, and P7 from 10-10 EEG placement are equivalent to nodes T4, T3, T6, T5 from 10-20 EEG placement. [29]

3.1.3 EEG paradigms

3.1.3.1 Evoked potentials

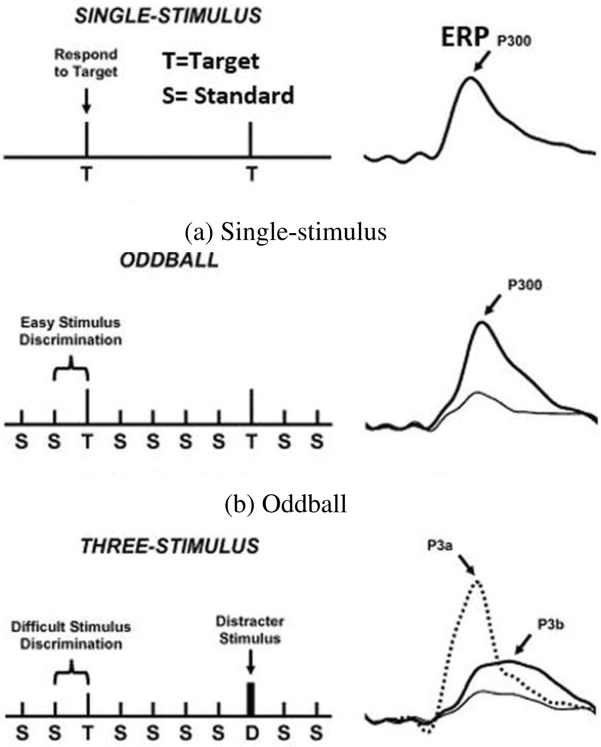
Evoked potentials are electrical potentials caused in response to some stimulation [32]. In most cases, these events must be compiled through several trials, amplified and averaged in order to ease its visualization and measurement.

There are two types of evoked potentials: [33, 34]:

- *Sensory evoked Potentials*: These potentials appear when the subject is stimulated by sight, sound, or touch. This stimulus is converted to signals that travel along the nerves to the brain, producing a peak on the resulting EEG. [33]
- *Motor evoked Potentials*: This term refers to the action potential elicited by non-invasive stimulation of the motor cortex. [35]

Sensory evoked Potential can also be further divided in:

- Visual evoked potentials (VEP). These are obtained in reaction to flashes or visual patterned stimulus.
- Auditory evoked potentials (AEP) are similar to VEP, but with an auditory stimulus
- Somatosensory evoked potentials are obtained by electrically stimulating the peripheral nerves.



(a) Single-stimulus

(b) Oddball

(c) Three-stimulus

Figure 3.5: Evoked potentials on three different paradigms. [36]

In Figure 3.5, three paradigms of evoked potentials with the corresponding alteration on EEG are shown.

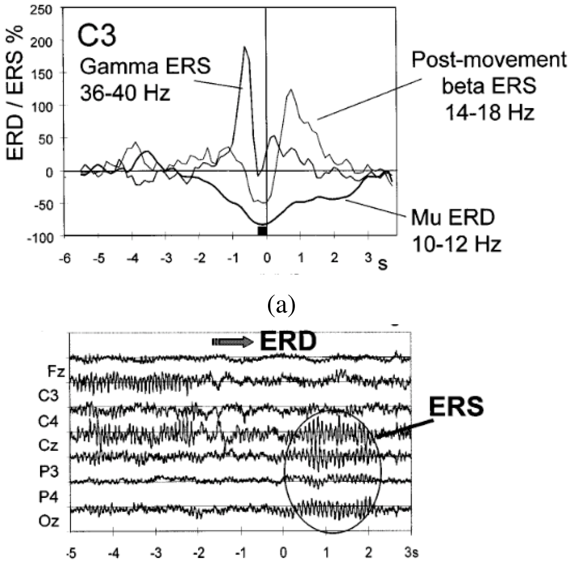
1. A single-stimulus paradigm, Figure 3.5a, is performed by irregularly presenting one type of stimuli to the subject. It results in a positive potential appearing 300 milliseconds after the stimuli, P300.

2. An oddball paradigm, Figure 3.5b, has two classes of simulation, displayed in sequence. The probability for each class to appear is not the same having one stimulus presented frequently, and the other rarely (oddball). Only the irregular event embosses the P300 peaks. [37]

3. The three-stimulus paradigm, Figure 3.5c, is a modified version of the oddball task where a non-target rare distractor stimulus is added to the standard and oddball class. In this case, a decomposition of the P300 into P3a and P3b can be observed. The distractor elicits a large P3a over the central area. In contrast, the oddball produces a P3b over the parietal area. [38]

3.1.3.2 Motor-imagery and Sensorimotor Rhythms

Primary sensory or motor cortical areas typically exhibit rhythmic activity at a frequency of approximately 8-12 Hz when they are not processing sensory information or producing motor output, called Mu rhythms. Computer-based analyses have demonstrated that this activity consists



(a)

(b)

Figure 3.6: Event Related DeSynchronization and Event Related Synchronization [39]

of a variety of different 8-12 Hz rhythms that are distinguished from each other by precise location, frequency, and/or typical relationship to concurrent sensory input or motor output, therefore performing a movements affects these rhythms.

Figure 3.6 shows the signal and band power signals during the execution of a right index finger lifting movement. The upper graph is the superimposition of band power time courses computed for three different frequency bands (10–12 Hz, 14–18 Hz, and 36–40 Hz) from EEG trials recorded from electrode C3. The vertical line corresponds to the start of the movement.

It can be noticed that about 3 seconds before the movement, there is a De-Synchronization of the Mu wave being associated with the brain preparing the movement.

Also immediately prior to the offset, there is a maximum of gamma ERS (Event Related Synchronization) and immediately after a maximum beta ERD (Event Related De-Synchronization).

On the lower graph, it can notice an EEG De-Synchronization at the central electrode locations prior to movement-onset and the enhanced alpha band activity over the posterior region (ERS) during movement.

This wave ERS and ERS can be ’learnt’ by a model and applied in a BCI as they are different according to each movement due to the spatially of the brain, Figure 3.7.

3.1.4 Available Datasets

For explicit communication, this work will focus on motor imagery. There are several Datasets available for training this type of BCI and all have protocols with different types of movement per session. The most important ones are presented below.

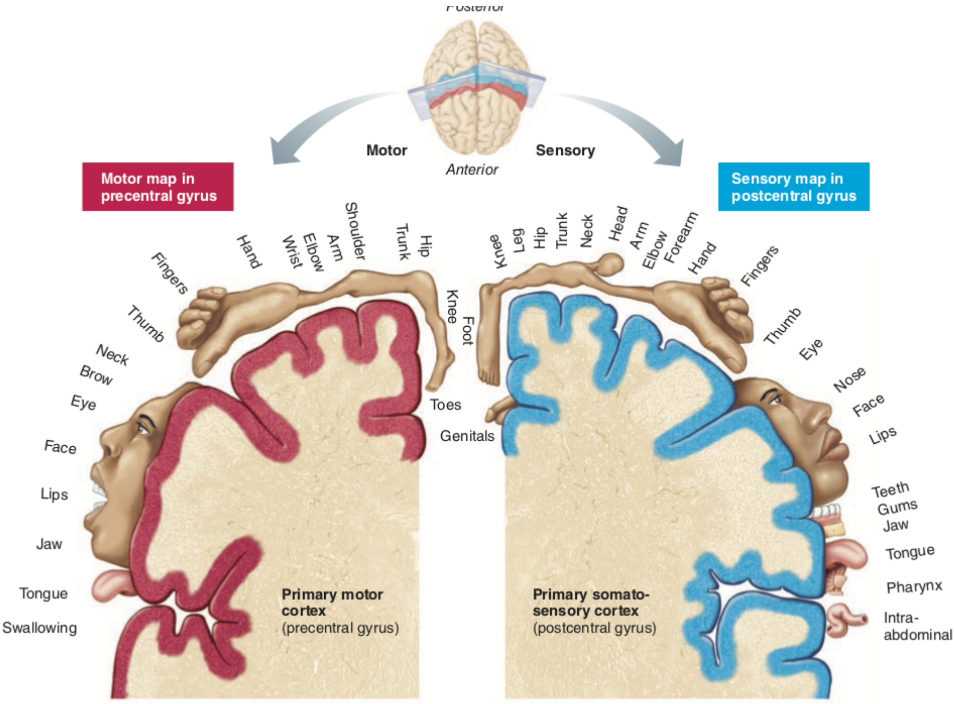


Figure 3.7: Body maps in the primary motor cortex and somatosensory cortex of the cerebrum. The relative amount and location of cortical tissue devoted to each function is proportional to the distorted body diagrams (homunculi). [40]

3.1.4.1 BCI Competition IV

This dataset is subdiveded in 3 different sub-dataset [41, 42, 43]

1. Data set 1 - Motor imagery, uncued classifier application: 2 classes of left hand, right hand, foot. 64 EEG channels (0.05-200Hz), 1000Hz sampling rate, 2 classes (+ idle state), 7 subjects;

2. Data set 2a - 4 class Motor Imagery: Left hand, right hand, feet, tongue. 22 EEG channels (0.5-100Hz; Notch filtered), 3 EOG channels, 250Hz sampling rate, 4 classes, 9 subjects;

3. Data sets 2b - Motor imagery: Left hand, right hand, feet, tongue. 3 bipolar EEG channels (0.5-100Hz; notch filtered), 3 EOG channels, 250Hz sampling rate, 2 classes, 9 subjects.

3.1.4.2 BCI 2000 EEG Motor Movement/Imagery Dataset

Available in Physionet this dataset contains recordings from 109 subjects of both hands and feet motor imagery data. Each person was subjected to 14 runs with two one-minute baseline runs and three two-minute runs of the tasks corresponding to each class. 64 EEG channels (unfiltered), 160Hz.

3.1.5 Main Features used in Motor Imagery BCI

3.1.5.1 Time-Domain Features

Several statistical features can be computed from Time-domain signal used by some authors.

1. Mean (μ_x);

2. Standard Deviation (σ_x);

3. Mean of the absolute values of the first differences (δ_x):

$$\delta_x = \frac{1}{N-1}\sum_{n=1}^{N-1} |x[n+1]-x[n]| \tag{3.1}$$

4. Mean of the normalized absolute values of the first differences ($\overline{\delta_x}$):

$$\overline{\delta_x} = \frac{\delta_x}{\sigma_x} \tag{3.2}$$

5. Mean of the absolute values of the second differences (γ_x):

$$\delta_x = \frac{1}{N-2}\sum_{n=1}^{N-2} |x[n+2]-x[n]| \tag{3.3}$$

6. Mean of the normalized absolute values of the second differences ($\overline{\gamma_x}$):

$$\overline{\gamma} = \frac{\gamma_x}{\sigma_x} \tag{3.4}$$

7. Fractal Dimension: Fractals are sequences that have details at tiny scales [44]. This feature gives information about space filling and self-similarity of a time series such as the EEG and can be computed using the Higuchi's algorithm [45]

8. High Order Crossings (HOC): A finite time series oscillating about zero level, such as EEG, will have a Number of Zero Crossings (NZC). When applying some filter to the signal, the oscillatory behaviour can change, therefore, also changing the NZC. If a sequence of pr-determined filters are applied to the time series, one can obtain a new sequence of NZCs, creating the HOC sequence. [46]

3.1.5.2 Frequency-Domain Features

Power Features can be calculated from several biosignals and in a EEG can be relevant for the classification of several characteristics [47]. The Digital Fourier Transform (DFT), Equation 3.5, can be used to compute the frequency components of the signal.

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-i2kn}{N}} \tag{3.5}$$

Due to the computation cost of the DFT, usually, a faster algorithm is applied: the Fast Fourier Transform (FFT). This quickness of the FFT allows computations in the frequency domain to be as feasible as those in the time domain. [48]

The Discrete Wavelet Transform (DWT) can also be used to compute Frequency Domain Features. This operation decomposes the signal into different approximation levels corresponding to different frequency ranges [49].

From both techniques, the standard statistical features, such as mean and standard deviation, can be computed and used as features [50].

3.1.5.3 FBCSP

FBCSP consists in a variation of another classical algorithm, Common Spatial Patterns (CSP). In CSP spatial filters are created that maximize the variance between two different motor images – it is, as so, adequate for binary classification. One of the biggest challenges in CSP is the selection of the frequency band to take into consideration – normally, a band between 8 and 30 Hz is selected, however, it has already been proven that results are improved when the selection of the frequency band is personalized to the user, which was what motivated the development of FBCSP [51]. The implementation process of the FBCSP can be seen in Figure 3.8.

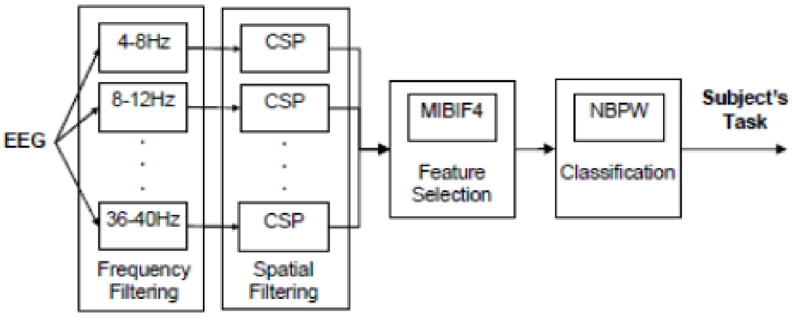


Figure 3.8: Stages implemented by the FBSCP algorithm.

In a few words, in FBCSP, a single trial EEG is classified using features extracted by spatial-temporal filters, personalized to the patient [52]. Similarly to what was mentioned to CSP, only binary classification is supported, so several multi-class extensions have been developed. The FBCSP algorithm [53] comprises the following stages:

1. **Filtering stage:** First, the EEG signal is filtered with a nine band-pass zero phase Chebyshev type II filters, with frequencies between 4 and 40 Hz;

2. **CSP algorithm:** applied individually to each one of the signal bands provided by the previous stage. It consists of the computation of spatial filters to detect Event-Related Desynchronization (ERD) or Event-Related Synchronization (ERS). By applying Equation 3.7,

features with an optimal variance to distinguish between two classes are obtained. In Equation 3.6, E is the raw EEG (NxT, being N the number of channels and T the number of samples per channel), W the projection matrix of CSP (the rows of W are stationary spatial filters and the columns of W-1are the common spatial patterns) and Z is the matrix that contains the computed features. Only the first and last *m* lines of Z are passed to the classifier. Each pair of bandpass and spatial filter yields CSP features that are specific to the frequency range of the bandpass filter, making this a patient-specific algorithm;

$$Z = WE \tag{3.6}$$

3. **Feature selection:** in this stage, the features that most discriminate two classes are selected. The selection of features occurs disregarding the classifier chosen and obliges a *a priori* selection of the number of features. The algorithm applied is the *Mutual Information Best Individual Features (MIBIF)*, in which the k features that maximize the mutual information are selected. Several subsets of features are tested. In order to select the best subset, Equation 3.7 is applied, in which ω is the class, X is the input features, obtained in the previous stage, $H(\Omega)$ represents the entropy of said class and $H(\Omega|X)$ represents the conditional entropy.

$$I(X;\Omega) = H(\Omega) - H(\Omega|X) \tag{3.7}$$

The final classification phase is not included as one can easily change it to the desired classifier.

As already mentioned, both CSP and FBCSP were designed for binary classification. As so, when the number of classes exceeds two, one of the following classification strategies is applied:

1. *One versus Rest (OVR)*: Train a classifier for each class training it versus the rest. The result is the classification result with the highest score.

2. *One vs One*: Train each class against another class, yielding, in the end, $\frac{N(N-1)}{2}$ classifiers, where N is the number of classes. Each binary classification result adds a vote to the class identified. In the end, the class with more votes is the output.

3. *Divide and Conquer*: This is similar to OVR but uses a tree, and successive classifications are done until converging to the result.

3.1.6 Performance of State of the Art examples

A search for the term "BCI motor imagery" yields a result of 11 300 papers, being impossible to review them all. This way for this section a search for BCI motor imagery review was developed and filtered to show only documents from 2015 or newer, reducing the number to 5 650. The most relevant review articles were then picked.

From this works a table with the best results, was compiled: Table 3.1

Table 3.1: Summary of most relevant literature reviewed for motor imagery classification.

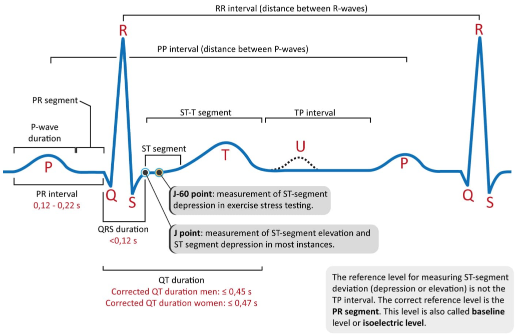
Used Features	Number of subjects	Classifier	Accuracy	Reference
Fractal Dimension	3	Adaptative LDA	80.2%	[54]
Band Power	6	Adaptative LDA	70.0%	[55]
CSP	10	Probablistic NN	83.8%	[56]
CSP	9	SVM	70.2%	[57]
Band Power	6	Gaussian Classifier	70.0%	[58]

3.2 ECG and Bio-identification

In this section, a short introduction of the ECG signal and some state of the art will be introduced. In this dissertation, the ECG module was implemented using an algorithm patented by the host group at INESC TEC.

3.2.1 ECG Signal Characteristics

ECG, represented in Figure 3.9, is the graphical recording of the electrical activity of the heart.


RR interval (distance between R-waves)
R
R
PP interval (distance between P-waves)
PR segment
P-wave duration
ST-T segment
TP interval
ST segment
P
T
U
P
Q
S
Q
S
PR interval
0,12 - 0,22 s
QRS duration
<0,12 s
QT duration
Corrected QT duration men: ≤ 0,45 s
Corrected QT duration women: ≤ 0,47 s

J-60 point: measurement of ST-segment depression in exercise stress testing.

J point: measurement of ST-segment elevation and ST segment depression in most instances.

The reference level for measuring ST-segment deviation (depression or elevation) is not the TP interval. The correct reference level is the **PR segment**. This level is also called **baseline** level or **isoelectric level**.

Figure 3.9: Classical ECG curve with its common waveforms, intervals and points. From Clinical ECG Interpretation

The electrical signal has its origin in the polarization and depolarization of cardiac cells. The P wave results from the atrial contraction and marks the beginning of the cardiac cycle. Shortly after, it follows the QRS complex due to the ventricle contraction. After depolarization the vehicle re-polarizes, resulting in a T wave [59].

3.2.2 State of the Art examples

ECG has been used for bio-identification in several literature works and submitted patents. Until 2014 there is already a comprehensive list presented at Paiva *et al.* [15], Table 3.2. However, a research on Google Patents on "ECG Biometric" [1] and filtering by data, from 2015 until today,

[1]

yields more recent results, showing the interest and potential of this type of technology. On table 3.3 the four most relevant ones, including the one patented by the group, are shown.

Table 3.2: Four most recent Patents in Biometric identification using ECG until 2014

Title	Authors	Description	Ref
Device and method for continuous biometric recognition based on electrocardiographic signals	H.Silva, A. Fred, A. Lourenço et al.	A finger-based ECG biometric device allowing identity recognition in an uninterrupted way—,technology applicable to clinical data protection, vehicles, tablet computers, etc.	[60]
System and method for continuous biometric recognition based on electrocardiographic signals	F. Agrafioti, F. Bui and D. Hatzinakos	Biometric security system and method able to authenticate one or more individuals using, physiological signals such as ECG, electroencephalogram (EEG), photoplethysmogram (PPG), and blood volume pressure (BVP).	[61]
Communication apparatus using biometrics	J. Hjelm and J. Soderberg	An apparatus allowing connection to a network after authentication of its user. This system, comprises a subscription module stored in memory and a controller that obtains biometric information of the user by using a sensor and compares this information to the identification information in the subscription module.	[62]
Electro-biometric identification	D. Lange	A processor based device that contains a pair of contacs that can be used to collect two different types of human biometric data. This data can therefore be processed and used to authenticate the user device.	[63]

Table 3.3: Four most relevant Patents in Biometric identification using ECG from 2014 to today.

Title	Authors/Compan	Description	Ref
System, terminal, and method for digital electrocardiogram authentication	Moon-Seog JUN	Electrocardiogram (ECG) biometric authentication system that performs biometric authentication using biometric, information having unique values for each client. Includes one or more client terminals and a key distribution centre (KDC) that uses ECG information of the one or more client terminals to issue a digital certificate	[64]
Method for biometric human identification on ECG and ptt smartwatch system using the same method	김인겸, 심소영	PTT(push-to-talk) smart watch system using a biometric personal authentication method.	[65]
System and method for biometric ECG verification of patient identity	Scott Baskerville, Sara England	A system including a heart monitor adapted to collect a baseline ECG signal from a patient and a monitored ECG signal from the patient; A memory to store a plurality of baseline statistical characteristics of the baseline ECG signal and a plurality of monitored statistical characteristics of the monitored ECG signal and processor adapted to compare the plurality of baseline statistical characteristics to the plurality of monitored statistical characteristics and determine a likelihood of patient match;	[66]
Biometric method and device for identifying a person through an electrocardiogram (ECG) waveform	João Paulo Cunha, Joana Paiva	Biometric method and device for identifying a person through electrocardiogram (ECG) morphology-derived waveform characteristics, using only one or a very few number of heartbeats.	[67]

3.2.3 Available datasets

There are several available datasets for developing and validating algorithms on ECG.

3.2.3.1 PTDB: Physikalisch-Technische Bundesanstalt database

Characterized by the following ECG Characteristics:

- 16 input channels, (14 for ECGs, 1 for respiration, 1 for line voltage)
- Input voltage: ±16 mV, compensated offset voltage up to ± 300 mV
- Input resistance: 100 Ω (DC)
- Resolution: 16 bit with $0.5\mu V/LSB$ (2000 A/D units per mV)
- Bandwidth: 0 - 1 kHz (synchronous sampling of all channels)
- Noise voltage: max. 10 $\mu V(pp)$, respectively 3 $\mu V(RMS)$ with input short circuit
- Online recording of skin resistance
- Noise level recording during signal collection
- Sampling Frequency: 1000Hz

It is composed of 549 records from 290 subjects (aged 17 to 87, with average of 57.2 years; 209 are men with an average age of age 55.5 years, and 81 women, with an average age 61.6 years; ages were not recorded for 1 female and 14 male subjects) with 15 simultaneously measured signals: the conventional 12 leads (i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5, v6) together with the 3 Frank lead ECGs (vx, vy, vz)[68, 69].

3.2.3.2 MIT-BIH Arrhythmia Database

This database has 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects. It serves as certification for validation of automatic QRS detection algorithms.

ECG Charateristics

- Sampling frequency: 360 samples per second
- Resolution11-bit resolution over a ten mV range
- two or more cardiologists independently annotated each record

3.3 Realistic UAV Simulators

In order to prevent wasting time and resource in field trials, a real-world simulation can help develop the proposed system in an accelerated way. On this section, two real-world simulators are presented.

3.3.1 VIPER: Virtual Image Processing Environment for Research

Viper is a project developed by the CMU team that aims to address the problem exposed at the beginning of the section. They defend that simulation can speed up learning methodologies that require repeated failure and success in the teaching process, such as deep learning if the simulation is close enough to reality.

The innovation in this work is to focus on the graphics in the, unlike classic robotic simulators that focuses on the accurate simulation of physics and sensors, making it a novelty for computer vision applications. Another innovative aspect of VIPER is the support for hardware, *i.e.* Embedded boards, on which the algorithms would run while on the UAV. The final objective would be a "truly unplug-and-go usability, with models trained in VIPER immediately usable in the real world."[2]

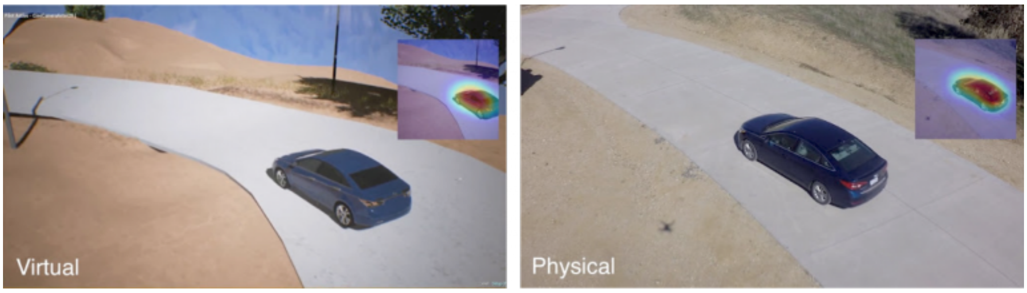


Figure 3.10: VIPER Environment

Meanwhile, the VIPER project was abandoned by the CMU-SV team.

3.3.2 AutoQuad quadcopter simulator

AutoQuad project was born at Berkeley University of California by group of students that needed a training environment for UAV autonomous navigation while running on low power machines [70].

This simulator was developed using the Unity editor [3] and is flexible enough for each user to build its own environment while maintain all UAV logic intact, allowing the vehicle to be testes in infinite scenarios.

However by building it in a way that runs in (almost) any machine, a disadvantage of this system is that it does not allow to have realistic simulations, Figure 3.11, and its not ready to have another controllable object other then an UAV, making it non ideal for the application where the human in the loop is required [71].

[2] Text based on and image taken from http://ccsg.ece.cmu.edu/wp/index.php/home/viper/

[3] https://unity.com/

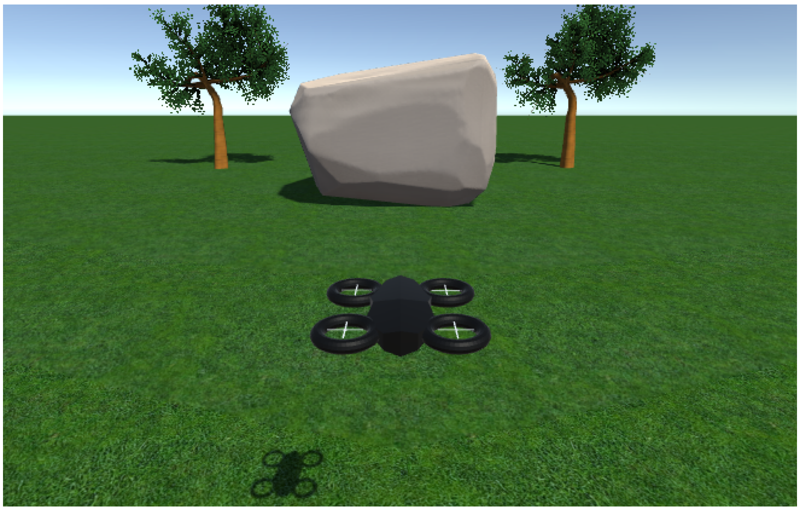


Figure 3.11: AutoQuad environment.
[70]

3.3.3 AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles

Airsim, by *Microsoft*, follows the same ideals as VIPER. It is open source[4] simulator for drones, cars and more, built on *Unreal Engine*.

Running on an excellence Engine like *Unreal* [5] allows the simulation to be photo-realistic, as seen in Figure 3.12. All vehicles have an exposed API and can be controlled outside of the simulation environment, therefore having also support for external hardware to communicate with the vehicle.

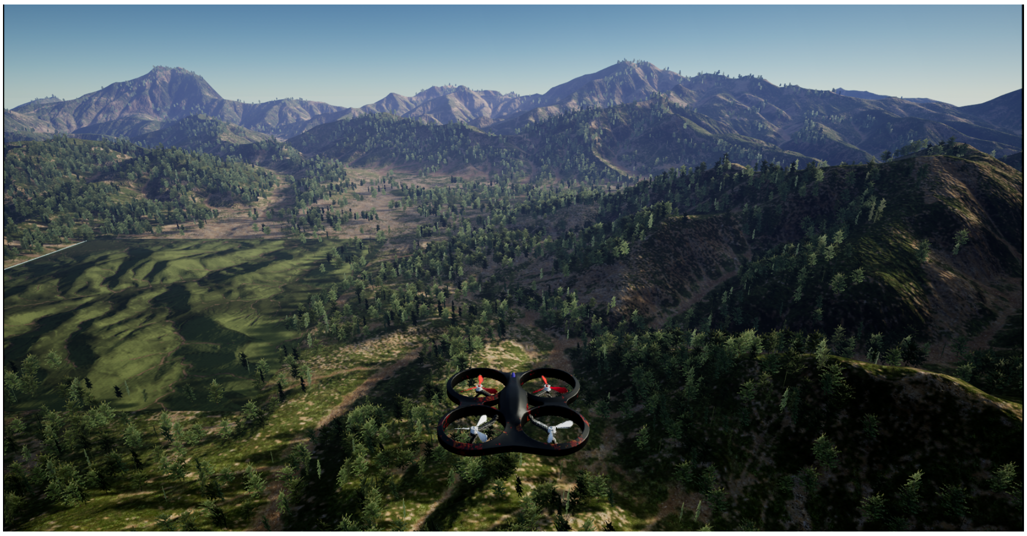


Figure 3.12: AirSim environment

[4] Available on https://github.com/Microsoft/AirSim

[5] https://www.unrealengine.com/en-US/

From the three exposed simulators, this was the one chosen. Contrary to VIPER, it is still being supported and continuously upgraded due to the wonderful community around it. In comparison with AutoQuad it has the advantage of supporting real photo-realistic environments, multiple UAV’s and, one of the most essential aspects of the project, the introduction of the "Human in the loop" within the level environment. Another reason to use airsim is that CMU-SV partners also do use it and in order to synchronize work and port any software developed it is better to have the same system running underneath. The negative aspect of it is that it requires a lot of computing power to run smoothly. For the simulation and physics engine itself is recommended to have a quad-core CPU and for a realistic environment an external graphics card is mandatory [72]. This requirements were not a blocker to the development of this system because BRAIN Lab did provide a machine capable of running it!

Chapter 4

Methodology used in biosignals classification

4.1 Machine Learning

Machine Learning (ML) is a sub-area of Artificial Intelligence (AI) that can identify patterns from given data. Generally speaking, it is a function that gives a discrete output by imputing features taken from data. This function, usually named *classifier*, must be tuned and adapted to each problem.

One of the significant challenges in this field is to have a generalist classifier that is not only able to correctly classify data already seen but also data not seen that has the same patterns as the training set.

4.2 Division in training and test Data

A good practice is, when training, to divide the Data available in the train set and test set. The training is fed to the classifier, and the test set is used to compute performance. This ensures the testing is performed in never before seen data, guaranteeing non-biased testing results.[73]

4.2.1 K-Fold Cross-Validation

A common technique used for hyper-parameters training is *K-fold cross-validation*. In this technique, the data (raw or transformed in feature vectors) is divided into several folds. One of the folds is then put aside for testing, and the remainders are used for training. On the next iterations, another fold is picked for testing, and the others are used for training. This process is repeated until all folds are used as a test set, as seen in Figure 4.1. In the end, the overall performance is the average of the performance of each iteration [73].

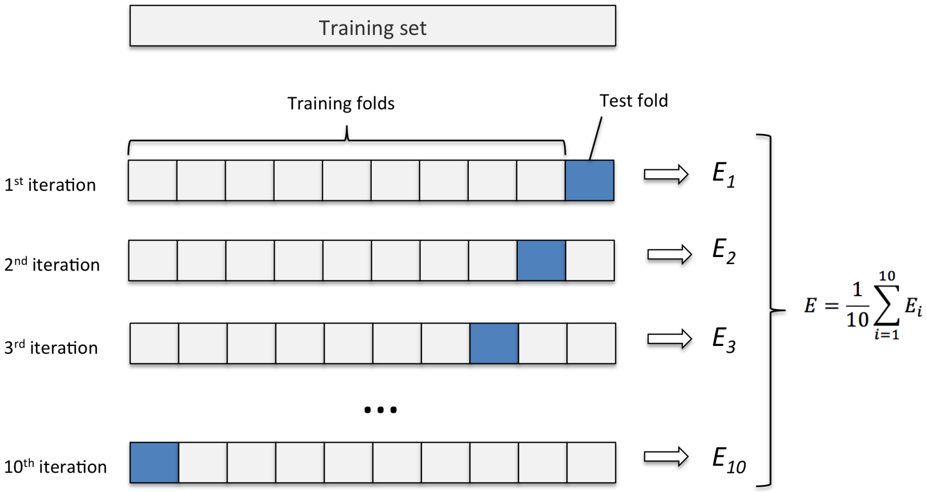
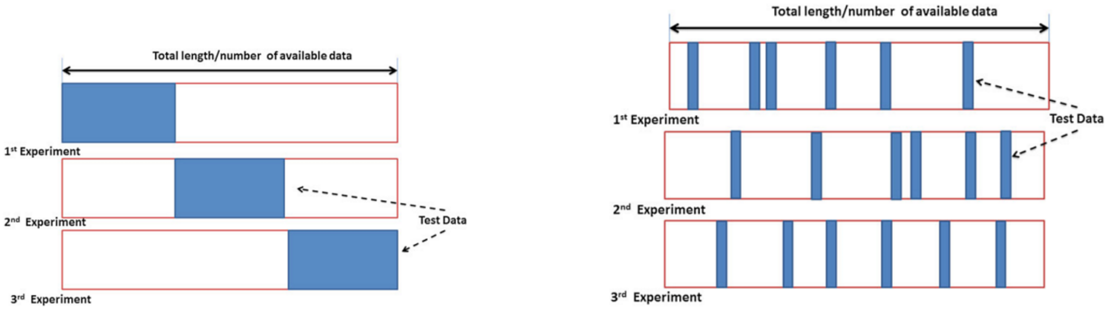


Figure 4.1: k-fold [74]

4.2.2 Monte-Carlo Cross-Validation

Another technique used training the classifier is Monte-Carlo Cross-Validation. It is performed by random selection (without replacement) some fraction of the data to form the training set, and then assigning the rest of the points to the test set, repeating this process multiple times. This way a new random training and test set partitions are generated each time. Since these partitions are done independently for each run, the same point can appear in the test or the training set multiple times 4.2.

The difference between these two techniques is that under *k-fold cross validation*, each point gets tested exactly once, which at a first glance seems fair. However, cross-validation only explores a few of the possible ways that the data could have been partitioned. Monte Carlo explores somewhat more possible partitions [75].



(a) K-Fold Cross Validation

(b) Monte Carlo Cross Validation

Figure 4.2: Monte Carlo vs K-Fold Cross Validation

4.3 Classifiers used in Biosignals

These are the actual functions used to predict an output given an input.

4.3.1 K-Nearest Neighbours

K-Nearest Neighbours, KNN, is told to be non-parametric. This is due to not be a ’true function’, but a model structure defined from data. It does not make a generalization of the data but uses all the ’training data’ to output a classification. The KNN algorithm is pretty simple. When new data reaches the classifier, it is compared to all the data used in training and classified according to the closest, as seen in Figure 4.3. Using one neighbour, the new data is identified as Class 1, but using three neighbours is identified as Class 2 [76].

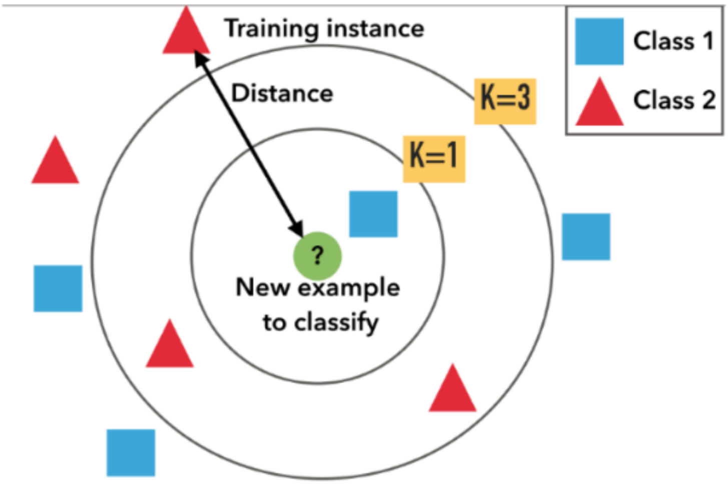


Figure 4.3: K-Nearest Neighbours [76]

4.3.2 Support Vector Machine

Support Vector Machine, SVM, uses a separating hyperplane to distinguish between classes [77]. The algorithm optimizes a plane to fit the given training data. In 2 Dimensions this hyperplane is a simple line, but in 3 Dimensions it might be a plane. This algorithm can expand dimensions to adapt to data that in a first look might not look correlated as seen in 4.4.

This classifier has three parameters that can be tuned: the Kernel, the Regularization parameter (C) and Gamma [77].

The Kernel is related to the expression of the hyperplane.

C defines how the boundary allows for misclassification. The bigger the C, the smaller the margins the hyperplane will have therefore better adapting the training that. Lower C will have a

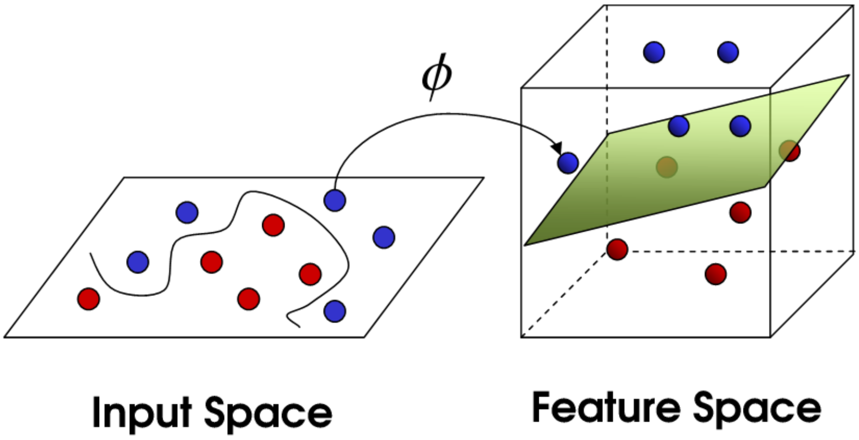


Figure 4.4: Support Vector Machine [78]

hyperplane with more significant margins even if it misclassifies some points. The problem with using large C is overfitting.

Gamma defines how fair a training point can influence the hyperparameter [77]. With low gamma points far away from the separating plane are to be considered. In the other hand, with high Gamma, only closer points to the boundary are used to compute the classifier expression.

4.3.3 Naive Bayes

Naive Bayes is a classifier that, in motor imagery BCI, usually assumes that the data follows a Gaussian distribution [79]. One can calculate the class posterior probability using equation 4.1 and then using the decision function 4.2 to calculate the predicted class.

$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)} \tag{4.1}$$

$$\begin{cases} \text{if } P(\omega_1|x) > P(\omega_2|x) \text{ if } x = \text{class } 1 \\ \text{if } P(\omega_1|x) < P(\omega_2|x) \text{ if } x = \text{class } 2 \end{cases} \tag{4.2}$$

4.3.4 Linear Discriminate Analysis

Linear Discriminate Analysis (LDA) is a classifier that assumes that both classes follow a normal distribution, with the same covariance matrix [80]. The hyperplane separating both classes is given by the projection, which maximizes the distance between the mean value of both classes and minimizes the interclass variance [80]. An example of LDA for a 2D classification problem is shown on 4.5.

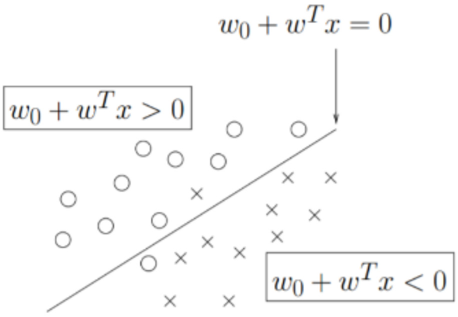


Figure 4.5: Linear Discriminate Analysis [81]

Due to its simplicity and fast training speed, LDA can easily be used in online classifications. It is suitable for tasks such as motor image classification [82], but it has the drawback of being linear, which can result in a lousy classification when the problem is not linearly separable.

4.4 Evaluation Metrics

In this section several metrics used over the next chapters are defined. Accuracy is both used in the identification module and in the BCI module.

Sensitivity, Specificity, Positive Prediction and Detection Error rate are metrics used the feature extraction component of the implicit ECG communication. They are used to measure the performance of the algorithm.

4.4.1 Concepts

For performance evaluation in ECG fiducials assessment, one of four cases can occur : **True positive - TP**: Point correctly detected.

False positive - FP: Point wrongly identified as a fiducial.

True negative - TN: Point correctly identified as a non-fiducial.

False negative - FN: Fiducial point not identified as one.

4.4.2 Accuracy

Accuracy, Equation 4.3, is the fraction of predictions the model gets right, both positives and negatives.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.3}$$

4.4.3 Sensitivity

Sensitivity, Equation 4.4 (only defined for binary cases), Equation 4.3, is the ability that the model has to identify a positive case correctly. In the ECG module case it is the ability that the model has to identify a true fiducial.

$$Se = \frac{TP}{TP+FN} \times 100\% \tag{4.4}$$

4.4.4 Positive prediction

Positive prediction,Equation 4.5, is the proportion of positive results that are true positive results. In the fiducial detector module, it is the proportion of points identified as fidutials that are true fiducials.

$$+P = \frac{TP}{TP+FP} \times 100\% \tag{4.5}$$

4.4.5 Detection Error Rate

This metric outputs the ratio between classes wrongly labeled and classes correctly classified.

$$Der = \frac{FP+FN}{TP+TN} \times 100\% \tag{4.6}$$

4.4.6 Information Transfer Rate

Information Transfer Rate is a metric proposed by [83] and measures the information transfer rate in a BCI.

$$B = \log(N) + P\log(P) + (1-P)\log[(1-P)/(N-1)] \tag{4.7}$$

B is Bit rate (bits/symbol), N is the number of possible class and P is the accuracy of the classification.

Chapter 5

EEG signal in Human-Drone Symbiosis

As stated on Chapter1 EEG signal can be used to interact with the drone both in an explicit or implicit way. Over the next sections, all work developed to establish this interaction is described.

5.1 Wearable EEG System

As a requirement of the project, all vital signals should be collected using wearable devices on the less intrusive way possible. This study was done between three different systems, sharing the same acquisition board but changing the electrode types.

The system used for EEG acquisition was the cyton board from openBCI shown in Figure 5.1. This board can be used to sample several physiological signals such as brain activity (EEG), muscle activity (EMG), and heart activity (ECG), however, during this dissertation, it was used in conjunction with an electrode system to capture just the brain signal. The main characteristics of the board are presented in Table 7.1 [1]

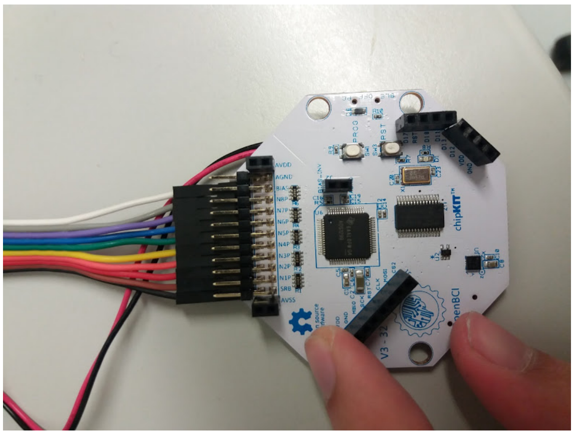


Figure 5.1: Cyton Board

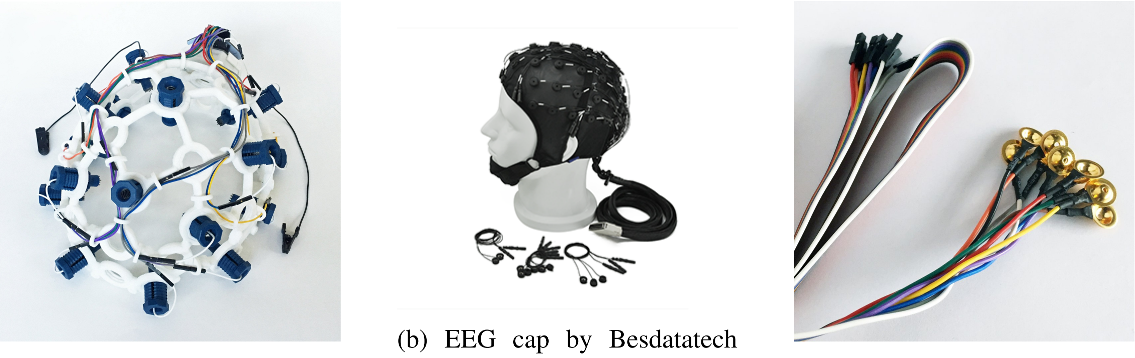
[1] Taken from https://docs.openbci.com/Hardware/02-Cyton

Table 5.1: OpenCBI Characteristics [84]

Specifications/Systems	OpenBCI
EEG sensors	8 channels that can be placed by the user at will
Connectivity	Bluetooth Low Energy
Sampling method	Eight High-Resolution Simultaneous-Sampling ADC
Sampling rate	250 Samples per Second
Resolution	24-bit channel data
Bandwidth	1 Hz - 50 Hz
Coupling mode	DC
Power Source	4 AA Batteries

The sampling rate is enough for us to use frequencies up to 125Hz(Accordingly to Nyquist theorem), being superior to the frequencies of EEG approached in BCI, as discussed in Chapter 3. It is powered by 4 AA Batteries and is small enough(6.1cm to 6.1cm in octagonal shape) to be portable.

The different electrodes used are *Ultracortex "Mark IV" EEG Headset with dry EEG comb electrodes*, a *EEG cap* or *Standard Gold Cup Electrodes* and compared using the same protocol. Each one was evaluated according to signal quality, electrode impedance, comfort to the user and set up duration on two different users.



(a) Ultracortex ”Mark IV”[85]

(b) EEG cap by Besdatatech [86]

(c) Gold Cup Electrodes[85]

Figure 5.2: EEG Electrodes Systems

5.1.1 Ultracortex "Mark IV" EEG Headset

Although the electrodes do not need any gel to capture signal, the setup of this system is not very user-friendly. The electrodes are held into the helmet structure by a screw-like piece that has to be adjusted to the user head shape. This has to be done before placing it in the head, or else the coating of the electrode might wear off due to friction with the scalp. This falls into an iterative process of placing the helmet and checking if the electrodes are in contact with the skin. If not, take the helmet off, adjust and try again. Besides, the electrodes are pressured against the head by a spring mechanism that often causes pain to the user.

The average electrode impedance measured was 20 KΩ, and might be the reason for a noisy signal. This wearable was already used in literature in BCI applications [87, 88, 50, 89, 90].

5.1.2 EEG cap

This setup is achieved by connecting the output pins from the EEG cap cord to the *cyton board* using the adaptor seen in Figure 5.3. The protocol to correctly place it on the user is as follows:

1. Use a *Skin Prep Gel*, such as the one from *Nuprep*, to strip away the top layer of skin and moisten the underlying layer.

2. Place the cap in the user's head by aligning key reference points like Fp1, Fp2, Cz, O1 and O2, making sure to places the odd channels symmetrical to the respective even ones.

3. Fill each electrode with electroconductive gel, e.g. *Electro-Gel* for *Electro-Caps* from *Bio-Medical Instruments* with a blunt tip syringe. The most appropriate technique is to perform circular motions while dispensing the gel until the electrode cup is full.

4. Repeat step 3 for each of the electrodes to be used.

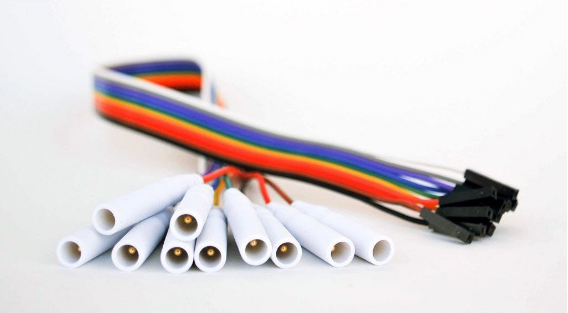


Figure 5.3: Cyton - EEG electrode adapter [85]

Similar setups are used in [49, 91].

Although it takes more time to set up the system then the *Ultracortex Headset*, approximately 20 minutes, the electrode impedance are much better, average 4 KΩ, translating into a better signal quality. A disadvantage of this system is that the electrode locations are limited to its position on the cap. Also, if the size of the wearable is not right, it will result in a too tight or too loose fit, resulting in poor electrode contact.

5.1.3 Standard Gold Cup Electrodes

Using the same adapter shown in the last section, Figure 5.3, each electrode was connected to the acquisition board. The setup protocol is different from the EEG Cap and goes as follows:

1. Use a marker such as a *Sharpie* pen or a crayon to mark the electrode location. This is a delicate step, and these marks should be made with the assistance of a measuring tape.
2. Use a Skin Prep Gel, such as the one from *Nuprep*, to strip away the top layer of skin and moisten the underlying layer.
3. Fill the cup of the electrode with an electroconductive paste substance containing *Collodion*. This paste will be responsible not only for the scalp to electrode conductivity but also for glueing it to the skin.
4. Place the electrode on the spot marked on Step One applying some pressure.
5. Place a piece of gauze on top of the electrode.
6. Repeat step 3, 4 and 5 for each of the remaining electrodes.

Protocols identical to the described were used in the following works: [52, 32].

This is the system that takes the most time to set up. It needs two persons to follow the protocol, and if both are inexperienced, it might take longer then an hour to finish. However, it is the more versatile one as the electrodes can be placed on any chosen location along the scalp. The average electrode impedance is 4KΩ resulting in a good quality signal.

Evaluating the pros and cons shown in this section, the *Ultracortex "Mark IV" EEG Headset* was immediately discarded due to being uncomfortable to the user and the resulting signal not having the best quality. Although the *Standard Gold Cup Electrodes* yield an excellent signal quality with low impedance, it needs too much effort and time to properly set it up, therefore for this particular application the *EEG cap + the cyton board* system was chosen as the optimal solution.

5.2 EEG Sub module Software Architecture

In this section, it is explained how the data is read from the device, how it is pre-processed, how the features are extracted, and how the classification process is handled, Figure 5.4.

All software in this module was developed in Python3.

5.2.1 Receiving and Conversion Module

The receiving module receives the data from the *cyton board* by *Bluetooth* using used the *pybluez library* to handle the communication and return a stream of 33 bytes arrays. The structure of this message, according to the *openBCI* official website [2] is as follows:

[2] https://docs.openbci.com/Hardware/03-Cyton_Data_Format

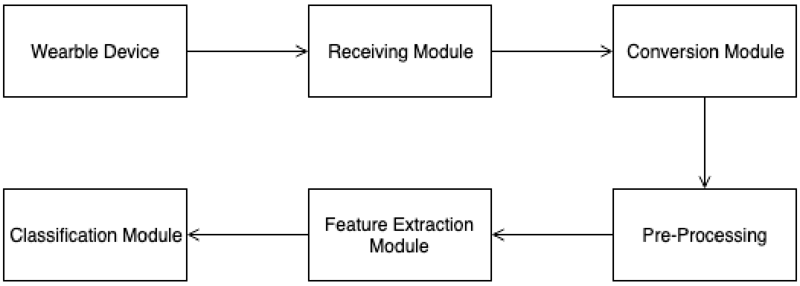


Figure 5.4: Data flow between the different software modules for a general wearable processing pipeline

- **Header**

  Byte 1 0xA0

  Byte 2 Sample Number

- **EEG Data**

  Byte 3-5 Data value for EEG channel 1

  Byte 6-8 Data value for EEG channel 2

  Byte 9-11 Data value for EEG channel 3

  Byte 12-14 Data value for EEG channel 4

  Byte 15-17 Data value for EEG channel 5

  Byte 18-20 Data value for EEG channel 6

  Byte 21-23 Data value for EEG channel 7

  Byte 24-26 Data value for EEG channel 8

- **Aux Data**

  Bytes 27-32 This bytes code data from the accelerometer and board internals that was not used in this work.

- **Footer**

  Byte 33 0xCX where X is 0-F in hex

The conversion module transforms this 'meaningless' stream of bytes into EEG values that can be used by the next pipeline elements. It first converts the 24 bits to a decimal integer and then to measure EEG value using the scaling factor formula given by the datasheet:

$$ScaleFactor(Volts/count) = 4.5Volts/gain/(2^{23} - 1); \tag{5.1}$$

In our application, the gain used was 24x, so our scaling factor was 0.02235 μVolts per count.

5.2.2 Pre-Processing, Feature Extraction and Classification Modules

The Pre-processing module is responsible for all signal modification needed for the computation of relevant features. In our work, it had two goals: Filter the data in order to eliminate external artefacts or components with no interest and divide the signal into epochs for the Feature Extraction Module to process.

A review of possible features to be used in motor movement/imagery BCI was already presented in Chapter 3. Based on literature the simplest method with best results was the *Common Space Filters*, being chosen and implemented using the *MEG+EEG Analysis and Visualization Toolbox (MNE).*

For the Classification Module, the chosen classifier was the Linear Discriminant Analysis, according to literature performs well when used together with the CSF, and implemented using the Scikit-learn Python Library.

5.3 Results in Public Datasets

In order to validate the algorithm before using acquired data, the BCI2000 dataset [92] available on PhysioNet [69] was used. Three patients were chosen randomly and the proposed BCI pipeline was tested in each one.

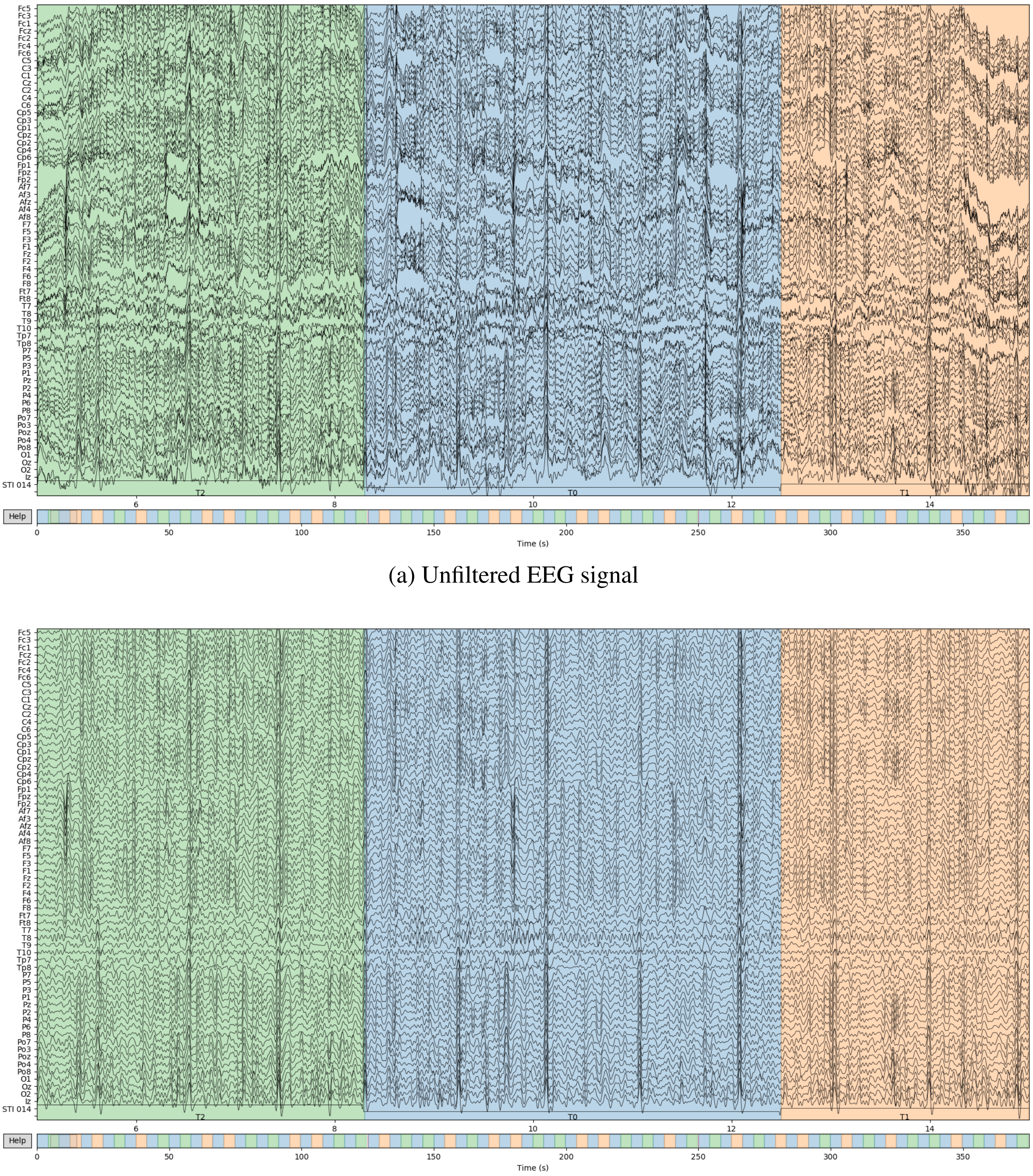
5.3.1 Pre-processing Results

In this section, an FIR (Finite Impulse Response) bandpass filter was designed using the pysignal firwin method. This function computes the coefficients of a finite impulse response filter using the window method (Hamming Window) yielding the results shown in Table 5.2

Table 5.2: FIR Filter Characteristics

Parameter	Value
Window	Hamming window with 0.0194 passband ripple and 53 dB stopband attenuation
Lower passband edge	7.00 Hz
Lower transition bandwidth	2.00 Hz (-6 dB cutoff frequency: 6.00 Hz)
Upper passband edge	30.00 Hz
Upper transition bandwidth	7.50 Hz (-6 dB cutoff frequency: 33.75 Hz)
Filter length	265 samples (1.656 sec)

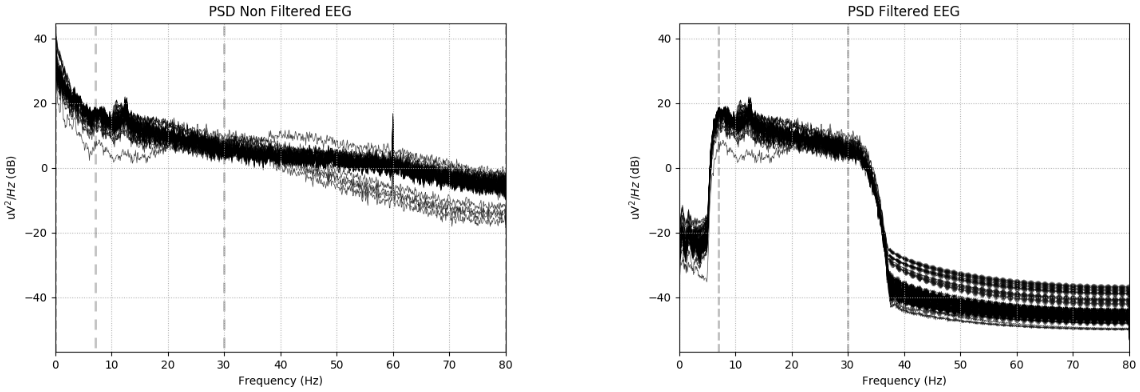
The filter output is shown in Figure 5.6 and can be concluded that this procedure correctly filters some non-physiological artefacts like baseline wandering and the 60 Hz power line interference while also limiting the information to the frequency band of interest for this application. On Figure 5.6, the Power Spectral Densities of the original and filtered signal are calculated. It can be seen that for frequencies between 7 Hz to 30 Hz the signal maintains its power while in lower and higher frequencies it falls for -20 and -40 db respectively.



(a) Unfiltered EEG signal

(b) Unfiltered EEG

Figure 5.5: Filtered vs Unfiltered EEG Signal. Each colors corresponds to an event. T0 (Green) - Rest; T1 (Orange) - Imagined Right Hand Movement; T2 (Blue) - Imagined Left Hand Movement;



(a) Power Spectral density estimation for the filter and unfiltered signal

(b) Power Spectral density estimation for the filter and filtered signal

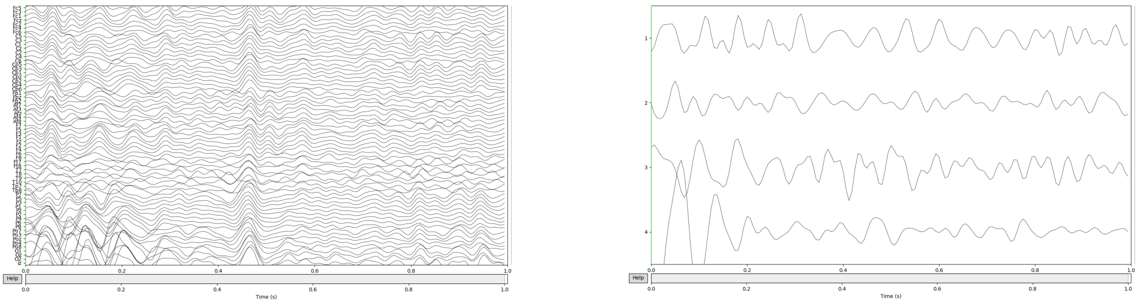
Figure 5.6: Power Spectral Density comparison; Raw vs Filtered

Before advancing to Feature Extraction, the data also needs some preparation. The signal was divided into each event (T0, T1 and T3) and then divided into 1-second epochs with a 0.1 seconds step size.

That data was then split into training and testing set, in a ratio of 0.8 and 0.2 respectively, ten times randomly, therefore, creating ten folds for future cross-validation. This division is made on the overall event blocks and not in the global pool of epochs. This means that if a specific event, for example, the first time the user executes the ’close hand’ task, is chosen as test data, no epoch from that event will go to the train data pool. However, data from the second time the user execute the ’close hand’ task can go to the training data.

5.3.2 Feature Extraction and Classification Results

For each fold in cross-validation, a set of common space filters were generated using training data. This translates into a transformation matrix that transforms the pre-processed data into a more discriminate space: In a 4 filter transformation, the 64 channel × 160 samples data, Figure 5.7a is transformed into a 4 channel × 160 samples matrix, Figure 5.7b.



(a) Pre-Processed EEG epoch

(b) EEG epoch in CSP space

Figure 5.7: Transformation to CSP Space

In order to predict the correct class on unseen data an LDA, using Singular Value Decomposition, was trained on each training set of the fold and tested in the respective test set. The features fed into the classifier corresponding to the average power of the output of each filter.

The results for each person are displayed in table 5.3 in terms of accuracy and Information Transfer Rate.

Table 5.3: Classification Results using BCI2000 Dataseet

Subject	Accuracy	Information Transfer Rate (Bits per Trial)
S1	0.75+-0.10	0.19
S2	0.93+-0.07	0.63
S3	0.71+-0.08	0.13

From this results, it can be noted that this setup performed well for one subject, subject 2, well beyond State of Art results, but the accuracy falls to 0.72 for S1 and 0.71 for S3. This decrease in accuracy significantly affects the Information Transfer Rate of the BCI and can induce errors in real time BCIs.

5.4 Acquired Data

For data acquired using the openBCI board, only one user, the author of the document, was considered. To train the pipeline described and used in Section 5.3, a protocol was developed to generate a dataset in order to have data together with the synchronized labels available. Two different types of sessions were performed in this step: one with motor imagery and the other with motor movement. Each session type was completed three times.

5.4.1 Dataset Generation

5.4.1.1 Aquisition Protocol

The acquisition Protocol developed based on literature is summarized in Figure 5.8. Each session starts with a 1 min Eyes Open Block and a 1 min Eyes Closed Block in order to capture a signal baseline and to check the presence of Alpha waves.

After two blocks of 1 minute, intending to capture possible noise that can contaminate the EEG, Jaw Clenching and Eye Movement, are performed.

The protocol then enters a loop that runs five times consisting in a Run and a 1-minute short break. Each run is composed of 10 trials of a random class. The random algorithm is implemented in such a way as that having two classes each class is performed five times, preventing class imbalance.

Each trial is composed of the following sequence:

1. *Cross*: A cross appears on the screen for two seconds.

2. *Cue*: A text box prompting the user to perform a due task is displayed for 1 second.

3. *Task*: The user performs the given task for 5 seconds.

4. *Rest*: The trial ends with a 2 second resting moment.

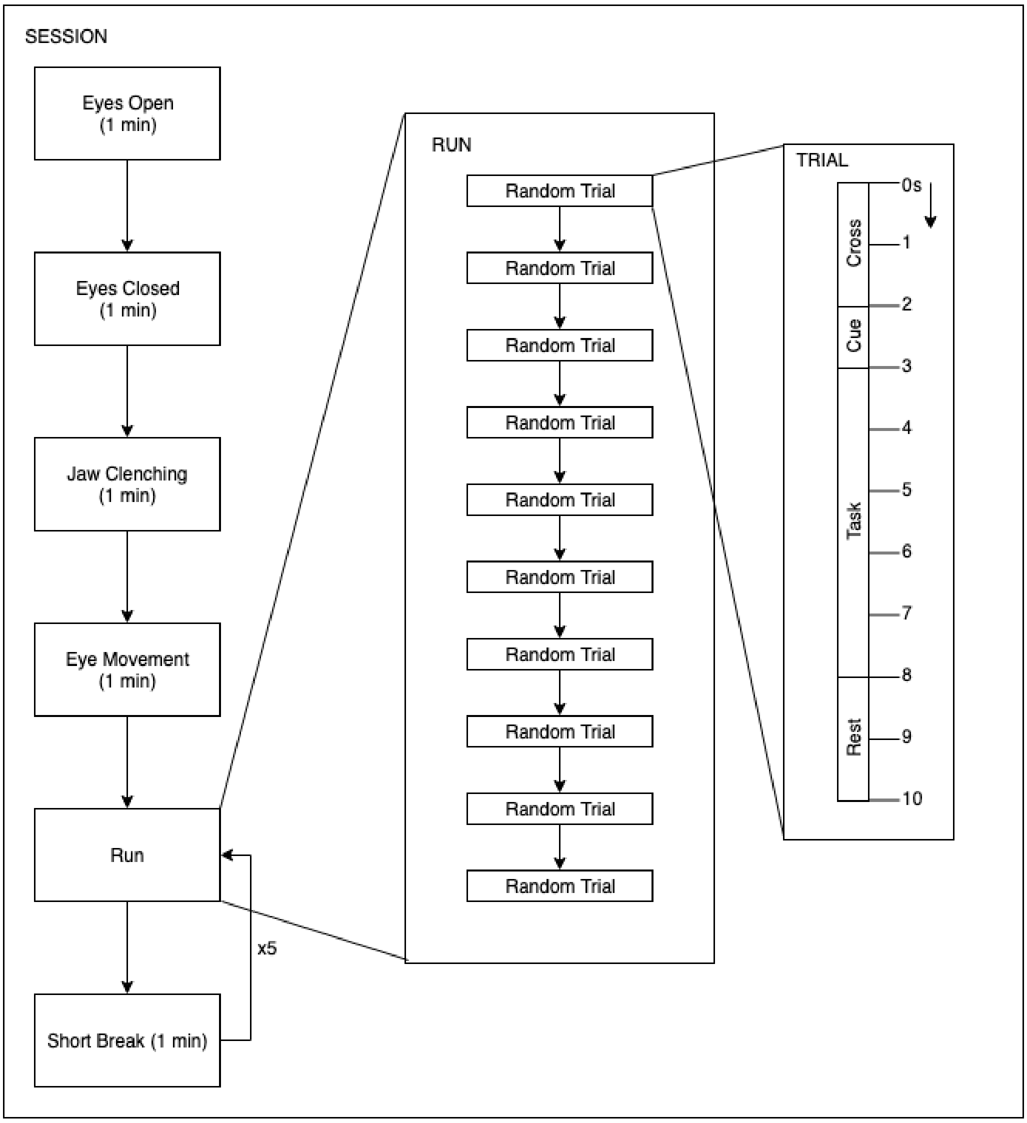


Figure 5.8: Acquisition Protocol

The classes chosen for both sessions was left hand and right-hand motor imagery/movement. The channels used were FP1, FP2, C3, Cz, C4, Pz, O1 and O2. O1 and O2 were chosen to check for alpha waves, C3, Cz and C4 are on top of the pre-motor cortex, being the most relevant in motor imagery. At last Fp1 and Fp2 goal is to capture eye blinks.

5.4.1.2 Data synchronization

The visual part of the protocol was implemented using the Psycopy Library. Each time new block was started in the protocol an event was recorded with the respective timestamp, being then possible to synchronize with the data acquired with the module described in Section 5.2.1 using both timestamps.

A network streaming module was also developed to be used in case of the protocol visualization tool and EEG acquisition software can't run on the same machine. Using the lab streaming layer (lsl) protocol, developed by Swartz Center for Computational Neuroscience, one can have several time series data streaming from different devices and as long as they are on the same network, this data can be sent to the lsl, synchronized using the network clock and can be pulled to any other devices being the contemporary data paired with sub-millisecond precision. Using this both the events and EEG data are pushed to the lsl and as long as both machines running the modules are on the same network, both or any other device in the same network can pull the already synchronized time series data.

5.4.2 Results in Acquired Data

The results for both sessions, motor imagery vs motor movement, are shown in Table 5.4 using the same methodology as the one used in BCI 2000 dataset.

Table 5.4: Aquired Data Results

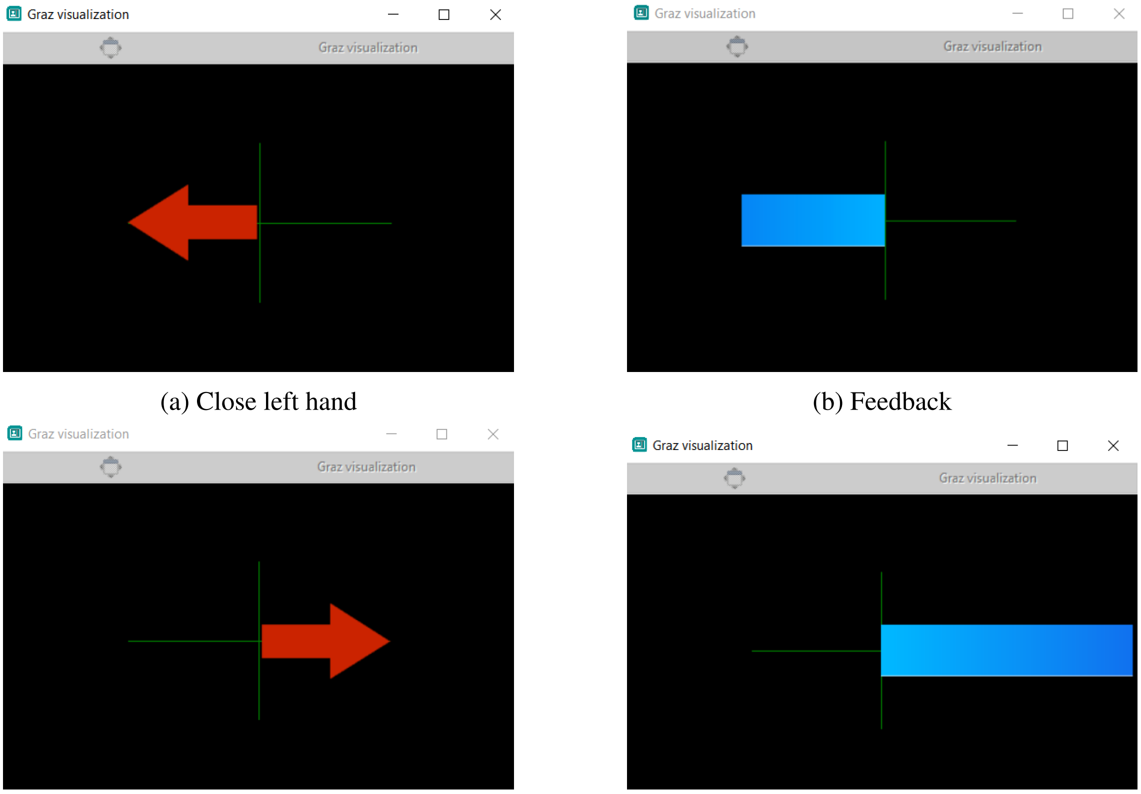
Session	Type	Accuracy	Information Transfer Rate (Bits per Trial)
Session 1	Motor Imagery	0.52+-0.20	0.00
Session 2	Motor Imagery	0.63+-0.15	0.04
Session 3	Motor Imagery	0.60+-0.14	0.02
Session 4	Motor Movement	0.70+-0.10	0.11
Session 5	Motor Movement	0.73+-0.11	0.15
Session 6	Motor Movement	0.75+-0.11	0.18

From the accuracy metric, it is trivial to observe that the pipeline is performing poorly. Session 1 with .52 accuracy is just little over chance level classification(0.5). The result of this first session is quite reasonable as the subject is still new to the protocol and trying for the first time to 'perform' motor imagery. In Session 2, the accuracy did increase, 52 to 59%, but with an ITR of 0.04, this trained classifier will not be reliable enough to be able to develop a stable BCI. Session 3 yield similar results to session 2.

The performance was expected to decrease, due to factors like the reduced number of electrodes and the signal being captured by a wearable device, but the result was worse than expected, and it proved impossible to develop a BCI with motor imagery.

According to a study developed by Pfurtscheller [39], in a population of 324 subjects, during a field trip, participated in BCI studies where a similar BCI setup was used. Only about 12% of the population, the two brain states proposed were distinguishable with an accuracy of 80% or higher. 78 % achieved an accuracy between 60%-80% and the last 10% no discrimination could be achieved. This shows that this type of interfaces are highly variable from subject to subject and in some cases, might be even impossible to perform. Other literature examples [39, 93, 94] show that motor imagery BCI performance does improve with months of training using neurofeedback technologies.

In the sessions with movement during the protocol, the method did perform better, being the lowest accuracy 70% and the highest 75%. This improvement was expected as the signal that originates actual motor movement is more prominently expressed in the EEG signal, and the behaviour of the subject during the protocol is not as subjective or dubious as imagining a movement. However, it still performs worse than the best case using a public dataset.



(a) Close left hand

(b) Feedback

(c) Close right hand

(d) Feedback

Figure 5.9: Neurofeedback Application

In order to try to tackle the problem and improve the accuracy of the BCI, a neurofeedback application was developed with the goal of the human adapting to the classifier and not the opposite. This application, Figure 5.9, runs in real time, and the user is prompted with a message (arrow) asking to perform a movement (close left or close right). The live signal is then inputted into the same pre-processing and feature extraction pipeline described in past sections, and the resulting features are seen to the classifier trained on Session 6. The results of the classification are shown to the user with a bar, Figure 5.9, whose size is directly related to how confident the classifier is of the resulting class. After five neurofeedback sessions the classification results raised from 75 to 80% showing that this kind of systems can improve the performance of an offline trained BCI classifier.

Trying even further to improve the accuracy metric, a post-processing module was also implemented. This module takes five classifications from the system and returns the class that appears the most (max). Even though it adds some latency to the BCI, it improved the accuracy from 80 to 82%.

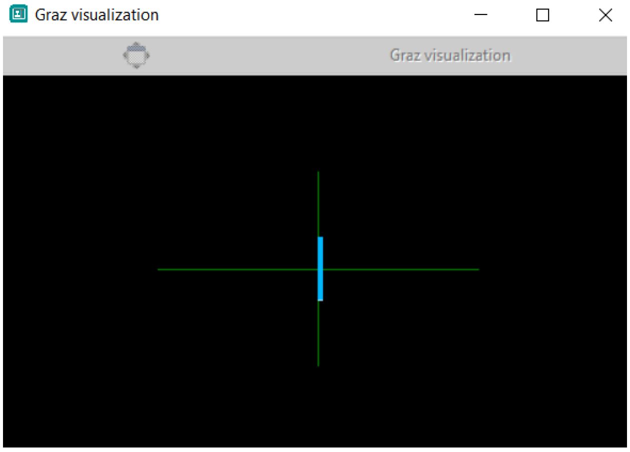


Figure 5.10: Feed back when the user holds the same state through time

With the help of this neurofeedback tool, an intriguing event was also observed. Each time the user switched states, that is whenever the subject was prompted to perform some action, the BCI had an excellent performance with the bar being almost always at his maxed size. However, as the user holds the same state through time, by performing the same movement, the BCI starts to output random results. A possible explanation for this event might be that the brain starts to automate the movement resulting in its execution without 'thinking it', being other structures recruited to be responsible for this movement.

Chapter 6

ECG signal in Human-Drone Symbiosis

In the developed system, ECG was used to develop an implicit communication between the UAV and human, more precisely a bio-identification mechanism. This module recovers developed work by the group reported in [15] and with a registered patent [67]. In the following sections, all work developed in order to implement this module is described.

6.1 Wearable ECG system

As stated in the last chapter, a requirement for our system is that the biosignals should be collected using wearable devices. The ECG system used in this work was the *VitalSticker*, a patch developed in *BRAIN LAB/INESC-TEC* Figure 6.1 . The major characteristics for this board are presented in Table 6.1.

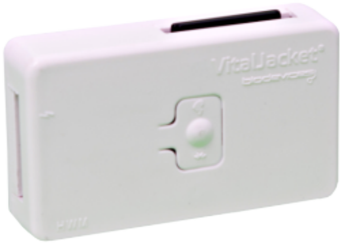


Figure 6.1: VitalSticker

Table 6.1: VitalSticker Specifications

Specifications/Systems	**VitalSticker**
Sampling Frequency	500 Hz
Resolution	8 bits
Connectivity	Bluetooth 2.0
Autonomy	3 Days of use

6.2 ECG Sub module Software Architecture

This section modules follows an architecture similar to the on presented in Section 5.2, Figure 5.4.

All software was developed in Python3.

6.2.1 Receiving and Conversion Module

Unlike the EEG processing module, the *Vital Sticker* already provides a receiving and conversion library for their data. The new part developed was a bridge between the biodevices software and the python3 module using named pipes.

6.2.2 Pre-processing, Feature Extraction and Classification Modules

The Pre-processing module is responsible for all signal modifications needed for the computation of relevant features, mainly noise filtering.

For fiducial points computation, the algorithm used for R point computing is the one reported by Pan & Thompkins [95].

Q, R, S and T points are identified using the first derived of the signal and must be located inside a defined window of physiologically possible size.

Q points were identified by finding the last temporal mark (and the closest one relative to each R complex) at which the derivative signal crossed zero in a time windows of 0.100 seconds before the identified R point.

A similar method was used for the computation of the S point. The first derivative changing signals in a window of 0.050 seconds after the calculated R point is pointed as S.

T wave was computed by finding the last derivative 0 crossings from positive to negative between 0.050 and 0.400 seconds after the computed R index.

All this physiological windows are reported on Paiva *et al.* [15] and taken from [96].

After these steps, the distance between fiducials was calculated, and ’bad heartbeats’ were rejected if failed the next two rules:

$$QR \leq 0.075s \tag{6.1}$$

$$0.200s < \frac{QT}{\sqrt{RR}} < 0.360s \tag{6.2}$$

The feature vector to be passed to the classifier is the distance between S and T, ST, the distance between R and T, RT and the distance between Q and T, QT, normalized by the average RR distance across the training pool of users.

A 5 fold cross-validation split was then performed in order to train a classifier. The classifier used was a multiclass SVM, Radial Basis Function kernel, together with a One vs One

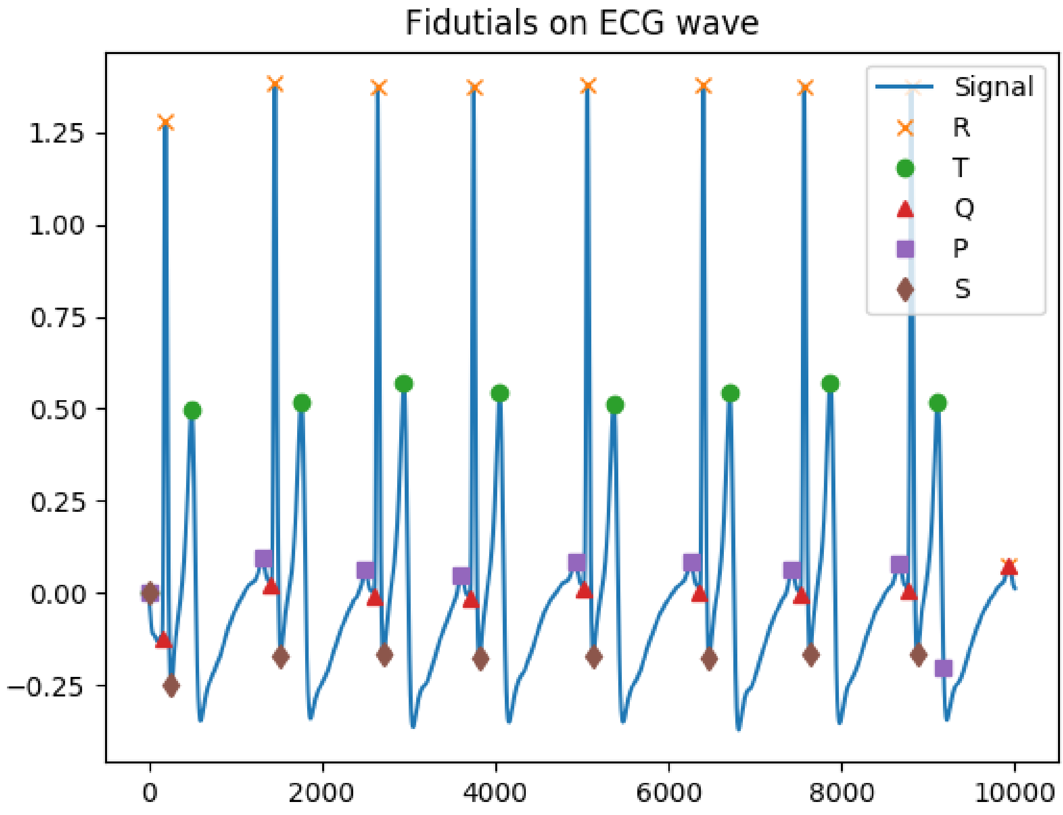


Figure 6.2: Fiducials in Dataset ECG

strategy. The other parameters were interated during the train phase on the following values: $C = 0.01, 0.1, 1, 10, 50, 100, 150, 200$ and $\theta = 0.01, 0.1, 1, 10, 50, 100, 150, 200$

6.3 Results in Public Datasets

To validate the feature extraction algorithm, a test on the PTDB dataset in a pool of 10 healthy subjects was first performed. These subjects were the same as the ones evaluated by Paiva *et al.* [15] in order to be possible to compare both implementations.

6.3.1 Pre-processing Results

The filter used in this step was simple bandpass FIR filter with cut frequencies of 3 and 45 Hz.

6.3.2 Feature Extraction and Classification Results

For each subject, the pool, a 120-second signal was extracted and the fiducials calculated. Figure 6.2 shows the result of the computation of fiducial points.

A comparison between a list of fiducials computed by Paiva *et al.* [15] and the points obtained by the developed algorithm was made and results are presented at table 6.2.

Table 6.2: Feature extraction results in comparison with Paiva *et al. [15]*

Subject N	Se%	+P%	DER%	TB	FN	FP
1	100.0	100.0	0.0	1141	0	0
2	100.0	100.0	0.0	1136	0	0
3	100.0	100.0	0.0	1143	0	0
4	100.0	100.0	0.0	1139	0	0
5	100.0	100.0	0.0	1141	0	0
6	100.0	100.0	0.0	1143	0	0
7	100.0	100.0	0.0	1136	0	0
8	100.0	100.0	0.0	1131	0	0
9	100.0	100.0	0.0	1142	0	0
10	100.0	100.0	0.0	1132	0	0

These metrics were found by comparing the points calculated by each algorithm. A match is counted if the absolute value of the time difference between the detected R point and the reference is less than 150 ms.

TP is the number of True Positive beats (correctly detected), FN is the number of False Negative beats (erroneously missed), and FP is the number of False Positive beats (erroneously detected).

Comparing the results, it can be concluded that all R points identified by Paiva *et al.* [15] are also identified by the implemented algorithm as expected.

The same process was repeated to the other fiducials yielding the same result.

As for classification results, for this ten subjects, the results from the five cross-validation process yielded an accuracy of 0.97±0.04. These are similar results to the ones shown by Paiva *et al.* [15]. Therefore it can be concluded that the process was replicated successfully.

6.4 Results on Acquired Data

After validating the algorithm on the public dataset, a real-time demonstration, presented publicly, was developed using acquired data on five volunteers (4 Male and 1 Female), all members of the host laboratory.

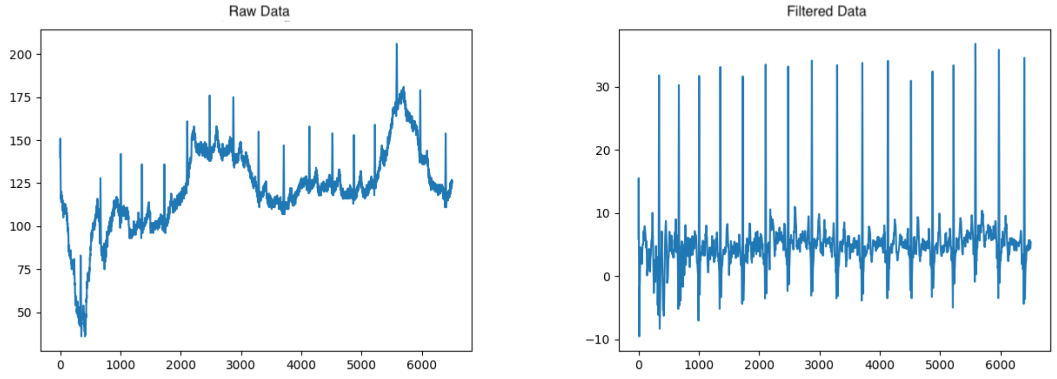
Figure 6.8 shows the difference between the Raw and the Filtered signal.

At first glance, the signal greatly improves. The baseline is corrected and power-line artefacts and high-frequency noise are removed.

On figure 6.4, the result of the fiducial extraction algorithm is also shown.

It can be seen that points of interest are assessed correctly and are ready for computing the feature distances to input into the classifier.

The feature pool was then split into the train, and test data and the SVM classifier was trained using a five-fold cross-validation process.



(a) Raw Data

(b) Filtered Data.

Figure 6.3: Raw vs Filtered Data

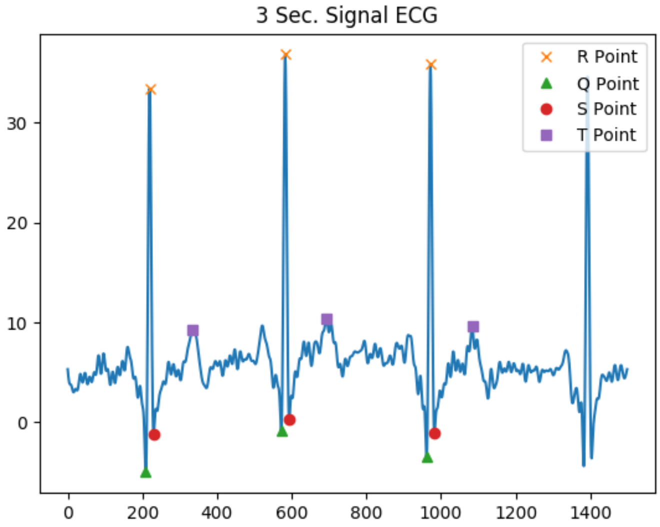


Figure 6.4: ECG Signal from Sticker (6 seconds)

The classification for the pool of 5 people is presented in 6.5, having a 0.83 (± 0.08).

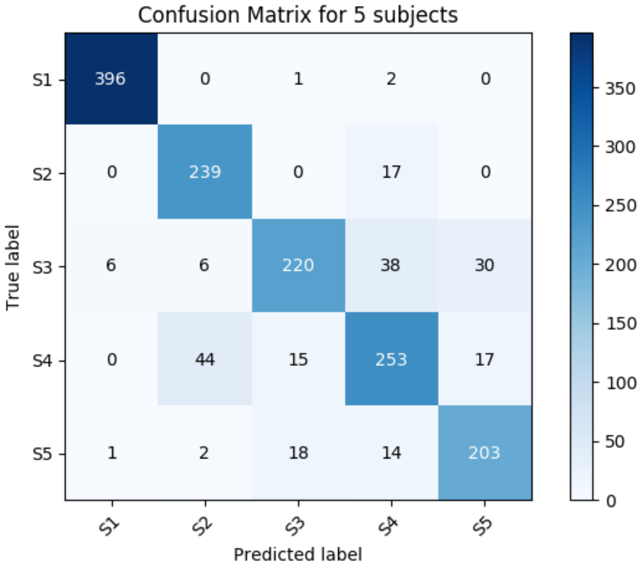


Figure 6.5: Confusion matrix for the best fold on a 5 person pool

Comparing with the data set results, this had a significant drop on accuracy, however, it is vital to notice that this signal was taken by a wearable device and by placing the fingers on the pads while the data acquired from the last section was taken by a clinical ECG acquisition system.

In order to assess the relation between users on the pool and classification accuracy, two people were removed from the pool and the resulting confusion matrix can be seen in: 6.6

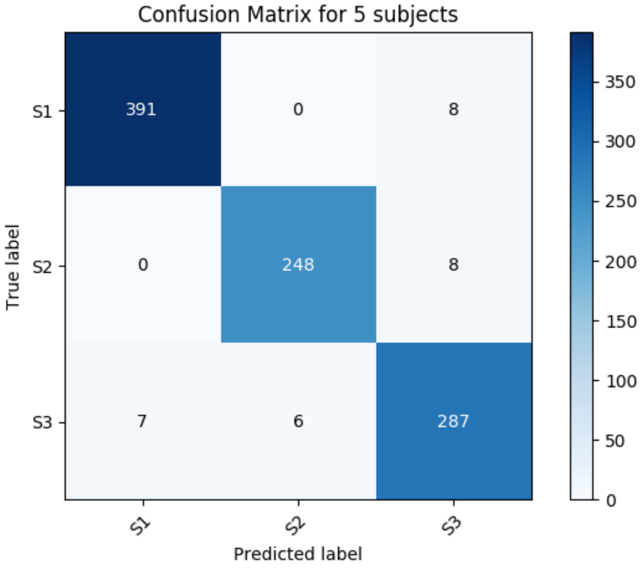


Figure 6.6: Confusion matrix for the best fold on a 3 person pool

This results had an accuracy of 0.95+-0.04. From this, it can be concluded that the classification improved its performance when lowering the size of the pool. However, these results can be misleading because it might be one the person of the pool destabilizing the classification. Further testing involving each combination possible of persons should be done to assess this possibility.

6.4.1 Live-Demo

Although the algorithm needs only one heartbeat to compute the fiducials and feature vector, 5 seconds blocks of the signal were used for each classification to ensure that there are at least 4 QRS complexes classifications. Each classification is associated with a class if the certainty of that classification is greater then .80. The final criteria for outputting the correct subject is the max of the classification array, as seen in Figure 6.7.

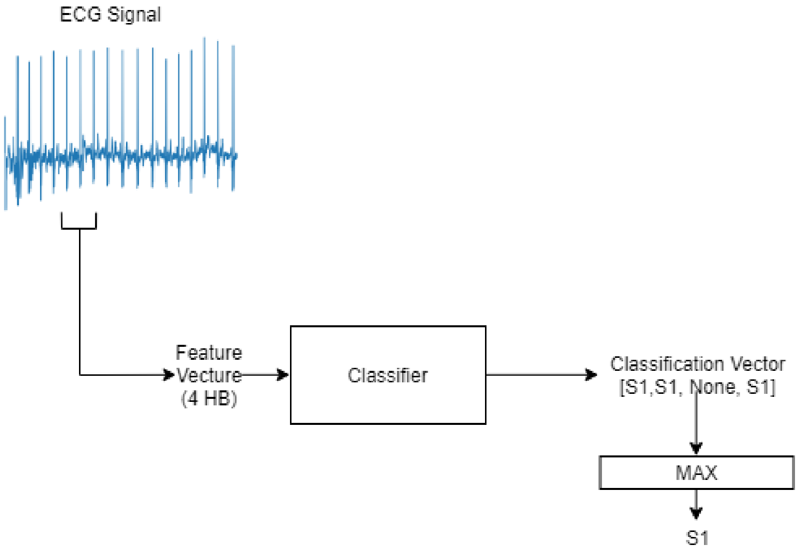
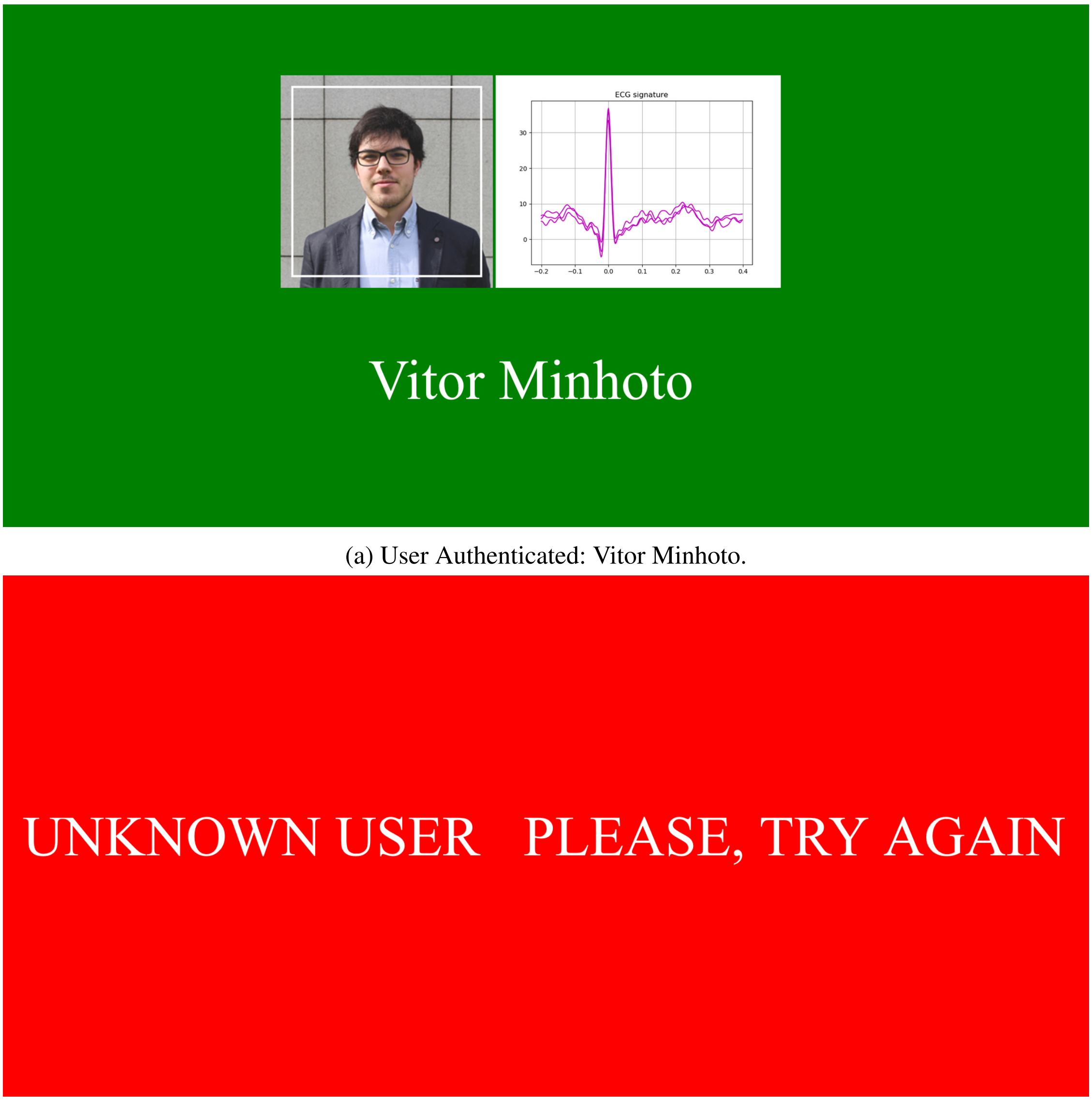


Figure 6.7: Live-Demo Classification Process

Despite this technique allowing the system to be more precise in identifying a user, it also introduces a delay, corresponding to the window length, in this case, 5 seconds.

A simple web app was also developed in order to show the results in a Demonstration to be used by the BRAIN group. Each time the system was sure about the identity of the subject, a page, Figure 6.8a, corresponding to that user would pop up, and when not sure a no user page would appear Figure 6.8b. On the user authenticated the heartbeats used for the identification are also show.



Vitor Minhoto

(a) User Authenticated: Vitor Minhoto.

UNKNOWN USER PLEASE, TRY AGAIN

(b) User Non Recognized.

Figure 6.8: Demo Web Interface

Chapter 7

System integration: The JIFX field trial

One of the goals for this work was to test the symbiosis at a field trial event that happened from the 29th of April to the 3rd of May on Camp Roberts, California. This event, *Join Interagency Field Experimentation*, happens several times a year and is a great opportunity to test project related with UAVs because it is hosted in an isolated military environment where participants have clearance to fly experimental devices.

Figure 7.1: Author on Mcmillan Airfield, Camp Roberts, entrance

The goal for these three and a half day event was to implement the explicit communication module described in Chapter 5 and the implicit communication module discussed in Chapter 7 on an intelligent software module that would control a drone (3DR SOLO), figure 7.2, supplied by the CMU-Silicon Valley Partners.

Even though the event was planned as a multi-day endeavour, the test phase ended up having the duration of only one morning. The next sections present the reasons of this change of plans and how they were solved

Figure 7.2: 3rd Solo drone from CMU partners

7.1 Information Flow

As stated, all modules were written in python3. However the communication library, that sends commands to the drone, *DroneKit*, runs on python2. It took two full days to solve this problem and the fix is shown in Figure 7.3

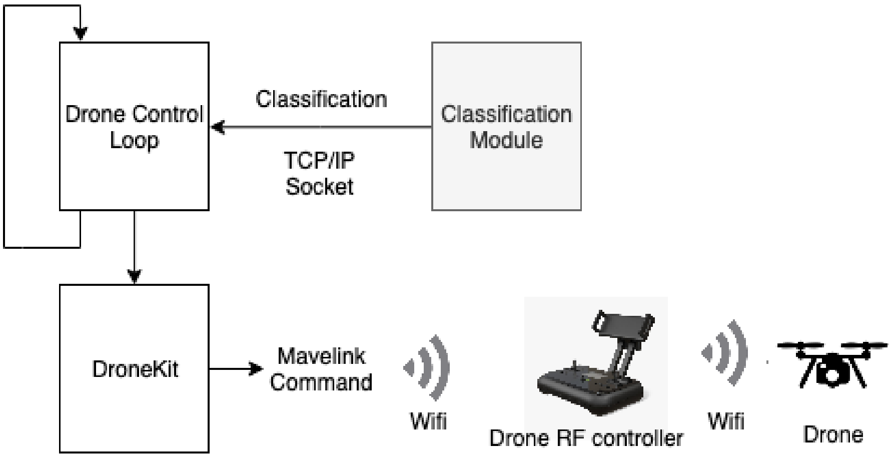


Figure 7.3: Information flow Classifier-drone

The shaded element is the module already described in Chapter 5 or Chapter 6 as it can be abstracted to any classification output object. A blocking TCP/IP socket was then implemented to transfer information from a python3 kernel to the Drone Control Loop running on python2. The Drone Control Loop is the module that handles the intelligence of the drone. It is the module that changes the drone mode or issues a command accordingly to information not only from the classifier but from the drone itself. If the control loop receives a classification, it then sends a command to the drone using the *dronekit Mavelink interface*. Mavelink is a communication protocol between software/hardware and the drone where commands can be sent or information can be received. It works as a broadcast mechanism, where the drone broadcasts information and the controller's broadcast commands. The command used in this experience was "move to a certain location" where the location was defined with GPS. Each class corresponded to a location so the drone would go from point A if the Classification Module yielded class A, or to point B if the outputs class B, in a protocol denominated by First Base Second Base. This mavelink command is broadcast to the RC remote controller private wi-fi connection, and if the drone successfully receives it, it will go to point A or point B.

7.2 Signal Quality

Being in the middle of a communication center serving the UAVs in the area affected the signal quality. The fix for this was to go to an isolated room where the signal captured by the openBCI board + eeg cap, Figure 7.4 had no interference. However, being isolated had the problem that the user could not be near the drone, therefore making it a difficult task to drive it without seeing where it was and where to go.

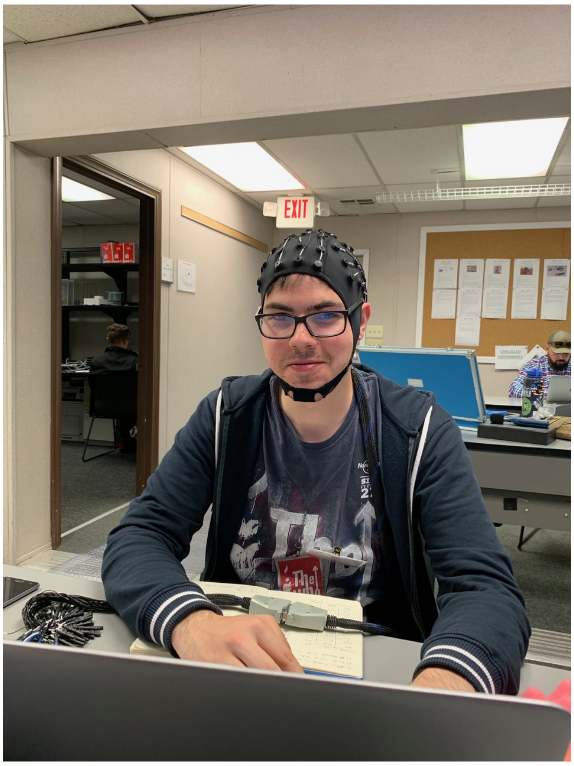


Figure 7.4: Author wearing the eeg cap in the control room in Camp Roberts.

7.3 A new Human-Computer Interface

Due to the problem presented in Section 7.1, the system could only be tested during a 1-hour window on the last day morning. It had to be decided that a new explicit communication module that was easier to set up, straightforward and easy to use should be implemented overnight to test on the following morning. The wearable used was MuseEEG [1], figure 7.5

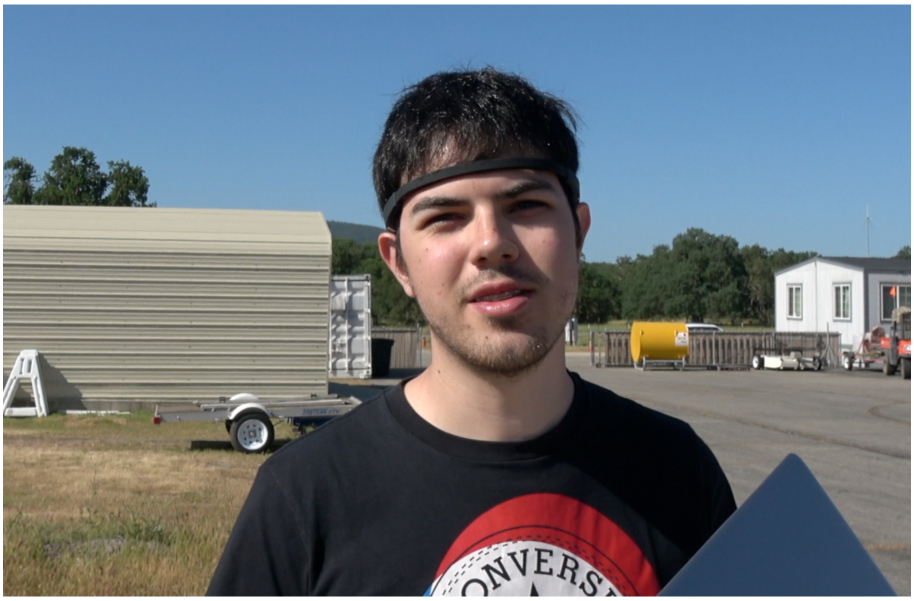


Figure 7.5: Muse Band (and author) in Camp Roberts

The muse band is a four channel (+1 auxiliar channel) EEG wearable device. It uses dry contact electrodes and it is mains characteristics presented on the following table.

Table 7.1: Muse Band Characteristics [97]

Specifications/Systems	Muse 2016
EEG sensors	4 channels TP9, AF7, AF8, TP10 (dry)
Connectivity	Bluetooth Low Energy
Sampling rate	256Hz Samples per Second
Resolution	12 bits / sample
Power Source	Rechargeable Li-Ion battery (Maximum 10 hours)

This wearable is more comfortable than the ones referred previously but has fewer channels. This band was used to develop an interface with the drone using eye blinks.

The eyeball acts as a dipole having a positive pole (cornea) and a negative pole (retina). When it rotates about its axis, it generates an electrical signal with a large amplitude that can be detected by the frontal electrodes (Fp1 and Fp2).

[1] https://choosemuse.com/

When the eye rotates upwards, the positive pole is closer to the frontal electrodes producing a positive spike on the EEG signal. When it rotates downwards the opposite happens, the negative pole rotates away from Fp electrodes and a negative spike is produced, Figure 7.6.

This process is very similar to what happens in eye blinks. When the eye closes the cornea become closer to the frontal electrodes and a positive deflection is produced. When the eye closes, the cornea rotates away from the FPs, resulting in a negative deflection.

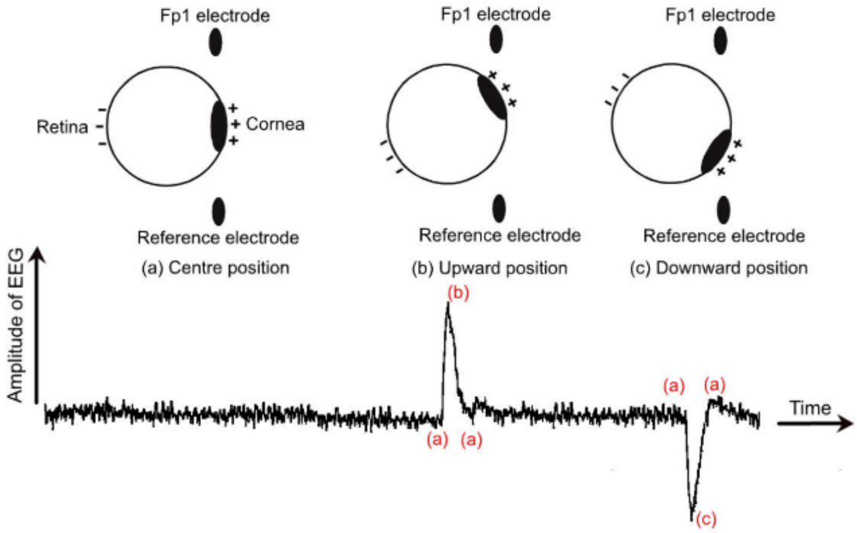


Figure 7.6: Effects of eye blinks in EEG channel[27]

This process is very similar to what happens in eye blinks. When the eye closes the cornea become closer to the frontal electrodes and a positive deflection is produced. When the eye closes the cornea rotates away from the FPs resulting in a negative deflection, Figure 7.7

The algorithm developed in order to identify blink or double blink is described below.

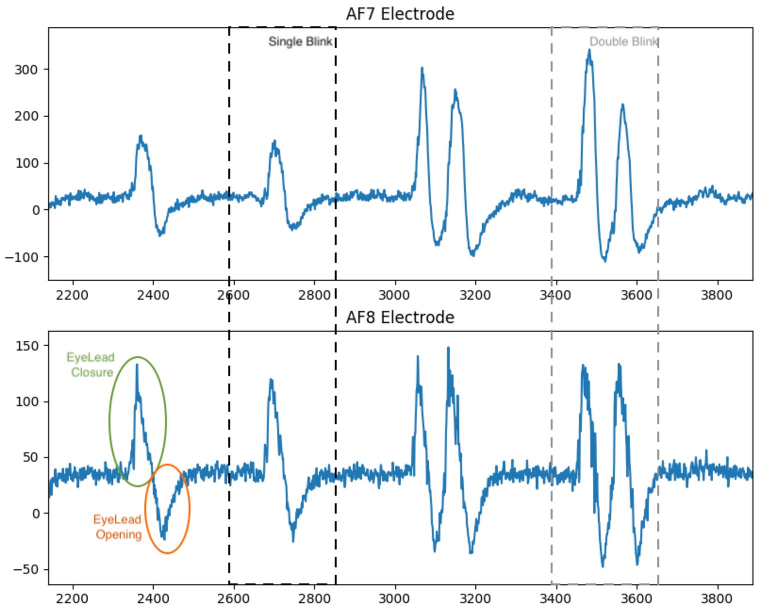


Figure 7.7: Raw signal: Blink and Double Blink

Algorithm 1: Blink Detector Algorithm

```
Result: Assess if user is blinking or double blinking.
while Signal is Streaming do
    $Signal = Signal^2$( Figure 7.8);
    if signal > th then
        Wait for 1 second signal;
        if Signal passes th 2 times then
            Return double blink;
        else
            Return blink;
        end
    else
        Do nothing;
    end
end
```

Although this is a very simple method to assess blinks in the EEG, it served the requirements of being independent of the user and easy to set up on the next morning flight. Despite performing well and identifying all the blinks while the user is still when there are brute head movements, this are also identified as blinks or double blinks.

The classification result from this module was then connected to the *First Base Second Base*

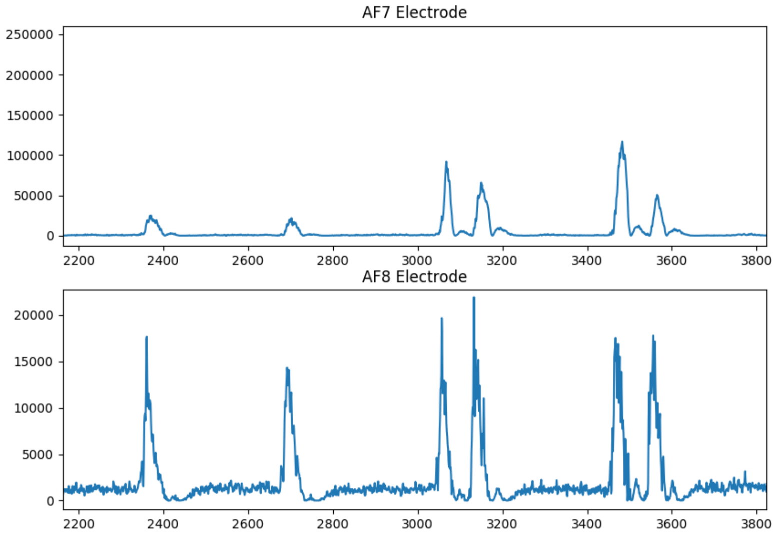


Figure 7.8: Squared signal: Blink and Double Blink

Protocol in a way that if the subject did blink once the drone would go to point A and if the user would blink twice, the drone would go to point B.

7.4 JIFX Results Discussion

The main goal of this participation in JIFX event was to integrate the modules explained on Chapter 5 and Chapter 6 and test a symbiosis between a UAV and a human.

The event did not go as expected, and several difficulties made it impossible to achieve the planned symbiotic interaction such as bad signal quality and computer-drone communication problems.

Despite all these setbacks, small objectives were still achieved. The communication pipeline, following the architecture proposed in 2, between wearable-computer and computer-drone was developed and documented for future use.

A new human-computer interface, with eye blinks, using more straightforward features and easier to detect features was created and easily integrated into the system, proving its flexibility and modularity.

Joining these two achievements together, a real-world test was performed on the last morning of the event where the user did fly a drone with the "First base, second base protocol", Figure 7.9. This experience consisted in the user blinking once for the drone to go to point A and blinking twice for the drone to go to point B, Figure. Although this was not the planned two-way interaction set as the goal, it still was an overall positive event alloying to understand how this symbiosis can truly be built.

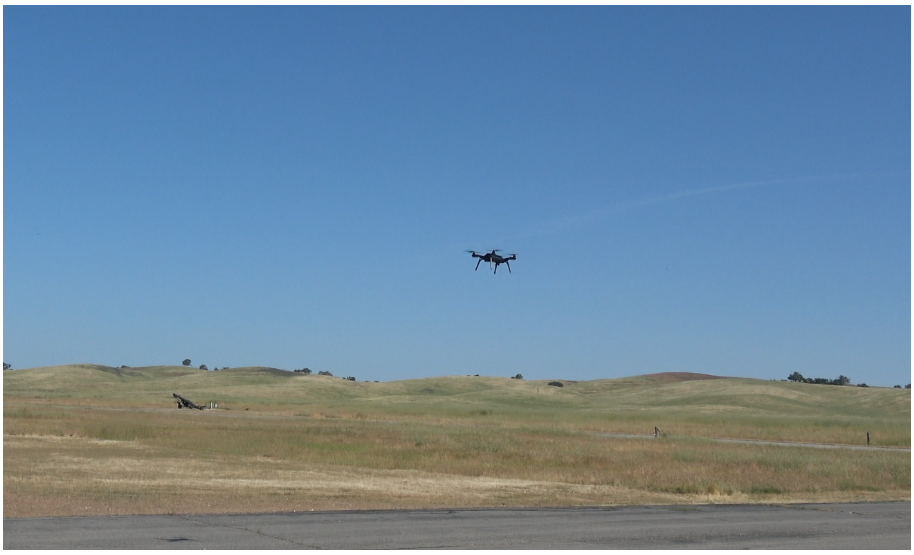


Figure 7.9: Drone following user commands.

The experience did raise interest on the other agencies participating in the event and the project was rewarded with a *Develop Now Coin*. This event also improved the relationship between the host group and the CMU-Silicon Valley team, both teams are willing to continue to further develop the project.

Chapter 8

Simulated environment

During this work, there was a need to test and *see* how does the UAV behave. In this chapter, the integration of our system within the *AirSim* Environment is described.

8.1 System Architecture

AirSim, as stated in Chapter 3, is a *openSource* vehicle simulation platform running on top of *Unreal Engine*. It adds support for drones, cars and environment sensors while maintaining the base photo-realistic environment and base classes of the *Unreal Engine* itself.

AirSim was developed as a platform for deep learning, computer vision and reinforcement learning algorithms for autonomous vehicles, explaining the need to have a photo-realistic environment, but as it is *openSource* and well documented it is possible to expand behind its core and bend it to fit our system.

There are three system components that are essential to simulate:

1. Environment
2. UAV
3. Human

Each of these simulations are detailed further over the next sections.

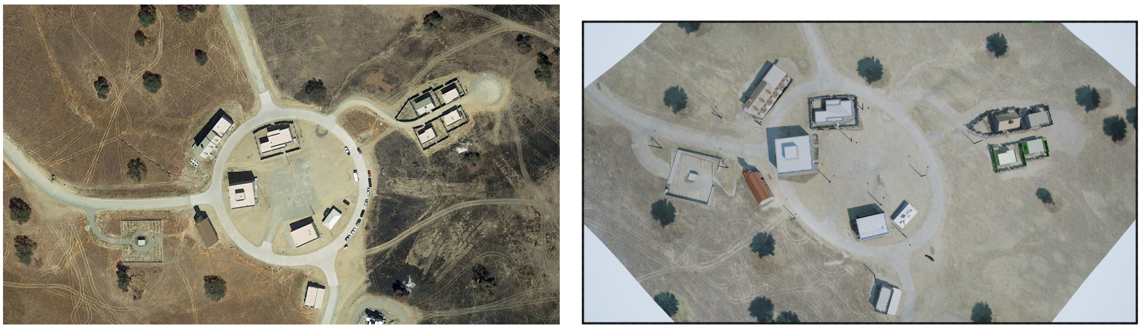
8.1.1 Environment

In order for the simulation to be as accurate as possible, the environment where the actors are included should be as close as possible to the real environment where they will be performing its goal action. On Chapter 3, it was already discussed the impressive power of photo-realistic scenarios. However, the ability to rebuild a real-world location is a must have in this scenario.

To test the abilities of this engine, a scene replication of the environment at Camp Roberts where the JIFX event took place was developed by the CMU partners.

Figure 8.1 shows a comparison between a real world photo, Figure 8.1a, and a top view of the simulation level, Figure 8.1b.

It can be observed that the simulated level is identical to the real world environment.



(a) Google Earth Photo of Camp Roberts CACTF (b) Simulator Screenshot of Camp Roberts CACTF

Figure 8.1: Comparison between real world environment and simulation.

8.1.2 UAV

The UAV simulation is handled by the *AirSim plugin*. It changes the *gameMode* of the base *Unreal* transforming the base Pawn class into a FlyingPawn and exposes an API to control the UAV. This API can then be accessed using a software module, like our Intelligence Module, supporting Python 3, C++ and Java or though Hardware like an RC or X-box Controller.

The simulated drone can be seen in Figure 8.2.

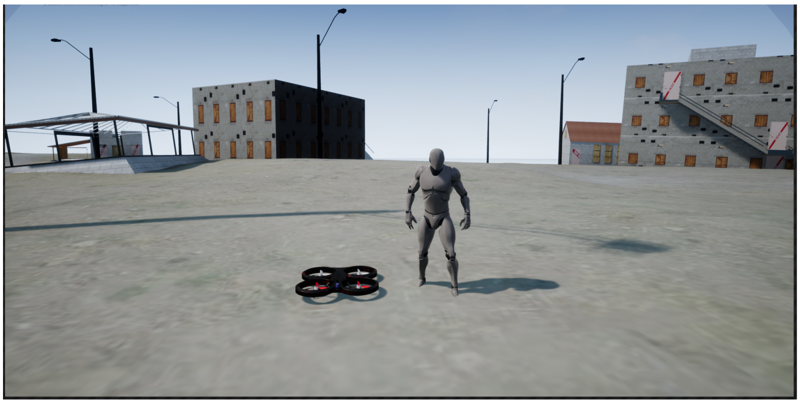


Figure 8.2: Human and Drone in Simulation

8.1.3 Human

This was the hardest User to implement into the simulation because AirSim game mode kept trying to transform the Human Pawn (Base Unreal Engine human class) into a FlyingPawn.

In order to solve this a new object was created, called humanAI, and an exception was added to the airSim source code in order for the gameMode to not populate this class.

This humanAI class was also created with the ability to have any biosignal streaming from it and an API exposing it to both classes inside *Unreal* or devices exterior to the simulation. For the asset a general puppet mesh, available in the standard *Unreal* content and seen on Figure 8.2, was used, but can be easily replaced by a real human-like skin developed in a software like *Maya*.

Three behaviours were also designed for this class:

1. Roam: The character walks randomly on a given area of the map.
2. Chase: The character chases another pawn that can be another humanAI or FlyingPawn.
3. Patrol: Given an array of points the humanAI will follow those points in a given speed.

With these three components, environment, UAV and Human simulated it is possible to test our architecture in simulation saving time and resources that are spent in field trials.

In order to assess the reliability of this simulated environment 2 test cases were developed and are presented over the next two sections.

8.2 Recreation example: JFIX event

In this test the goal was to recreate the results presented in Chapter 7. For this simulation the following objects were needed:

1. JIFX environment
2. Human simulation streaming EEG data
3. UAV simulation

For the environment the level described in Section 8.1.1 is used.

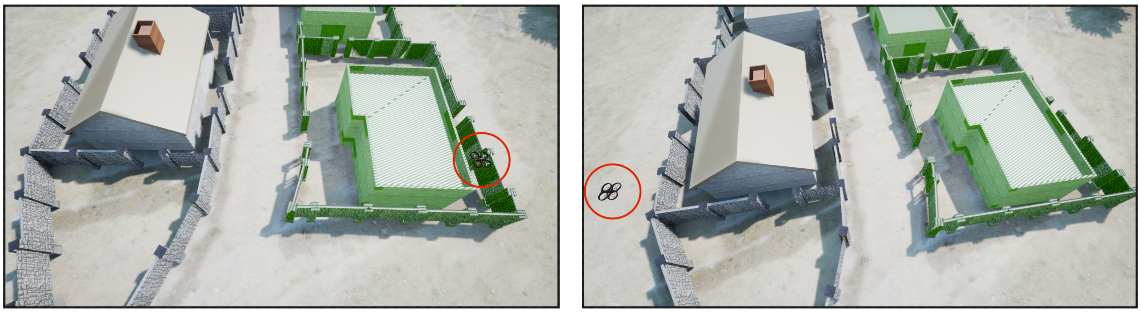
In this level the human did not have to perform any action, its only function was to stream EEG data previously recorded with the MuseEEG band.

For the drone the airsim FlyingPawn was used.

It can be observed that the initial architecture is still valid and easily adaptable to work with the simulator.

The intelligence module, in this case, receives the recorded EEG signal from the simulated human, identifies blinks and commands the drone accordingly following the first base, second base protocol, Figure 8.3.

The recreation was successfully built proving that this environment can be used to test and train the human-UAV interaction. In this case, as pre-recorded data was used, mainly the interaction on the drone part can be developed, but the system is easily adaptable to have live data streaming from a subject to the simulated human. This functionality was not tested during the duration of this work due to time constraints and the fact that the computer used to run this simulation did not have a BLE module to receive data from the wearable.



(a) Drone on Point A

(b) Drone on point B

Figure 8.3: First Base, Second Base protocol

8.3 Recreation Example: JIFX November 2016

Another recreation example was developed with that collected by Dr. João Paulo Cunha in a JIFX event that he took part in November 2016.

This experiment had several officers, also on CACTF, with two collecting systems and a smartphone collecting the data listed in Table 8.1.

Although all of this data was available, in this recreation only two data streams were used: GPS and ECG. In the recreation scenario the human simulation should follow the path that the officer did and stream his ECG data for the drone to take actions accordingly.

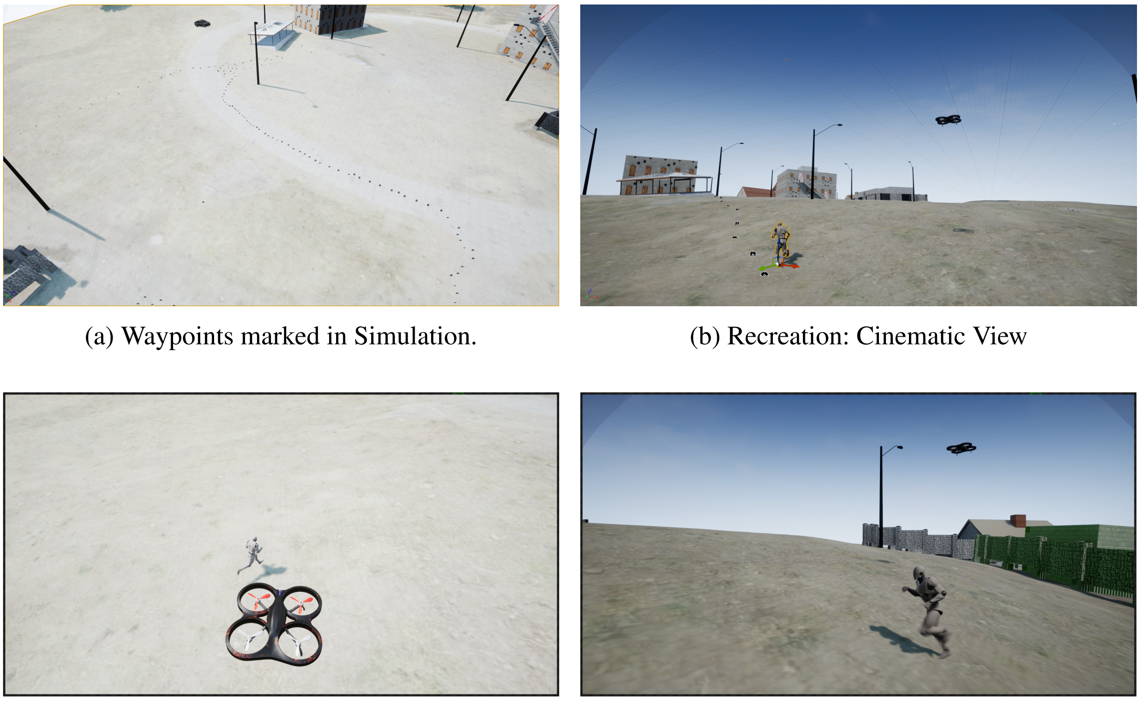
The GPS data was successfully loaded to the simulation and marked as waypoints, as seen in Figure 8.4a.

In the experiment the officer would go down into a group of houses, check for a 'strange object', then had to run to the starting point. Figure 8.4 shows several snapshots of the 'Return to base' action from different views. Image 8.4b shows the path the 'human' is taking, Figure 8.4c shows an air view and figure 8.4d shows a ground view. The simulated human works exactly as intended following the given path.

As for the drone, the programmed behaviour was to follow the human.

Table 8.1: Data Collected in JIFX 2016

Device	**Data**
Vital signals Monitor (VSM)	Medical-grade ECG Body temperature Body actigraphy
Ambiance Sensing Device (ASD)	Ambiance temperature Humidity Luminosity Athmospheric pressure Two hazardous gases: CO and NO2 Videos/photos
Smartphone	Time-tags GPS coordinates



(a) Waypoints marked in Simulation.

(b) Recreation: Cinematic View

(c) Recreation: Top View

(d) Recreation: Bottom View

Figure 8.4: Recreation Scenario.

Similar to the case shown in Section 8.2, this recreation can only be used for drone behaviour training. However it could also be coupled with a VR headset a VR omnidirectional treadmill and a real user with the same wearable systems attached. This would allow the simulator to receive human data, move the simulated human accordingly, and stream the received sensing data from the wearable all in an immersive world. In this case the human could adapt to the drone behaviour testes previously. With enough iterations, every time the Human or the UAV took action or change behaviour, the other part would immediately perceive it, achieving a real symbiosis.

Chapter 9

Conclusion

This work covers the vast field of Human-Machine Interactions by proposing a new concept: Human-Machine Symbiosis. A conceptual abstract architecture of this symbiotic system was designed and a specific case using EEG as explicit interaction and ECG as implicit interaction was proposed to be implemented.

For Explicit communication, a Brain-Computer Interface using motor imagery was tested yielding results similar to the ones presented in State of the Art on public datasets. However, when using acquired data from the wearable system, this Interface was impossible to achieve due due to differences between the data itself (64 clinical vs 8 wearable data channels) and the user not beeing familiar to the concept of motor imagery. A new BCI using motor movement was then developed, having good enough results to implement a real-time Interface. During this process was also concluded that neurofeedback does improve the BCI performance as the human adapts to the classifier. Another important observation was that the trained model drops accuracy as the movement is maintained and it was postulated that it might be because after the preparation and initial execution of movement, other brain structures are then recruited to continue it. This way the structures that we are evaluating are no longer responsible for the action the user is performing

For the Implicit Communication past work from the host laboratory on ECG bio-identification was re-implemented and implemented thinking in this particular scenario. The results yielded were similar to the ones obtained previously by a group member and a demonstration was successfully developed showing a wonrking real-time version of the algorithm.

During the JIFX event, the implementation of the two already described models were tried, but problems with communications with drone did not allow to perform this integration. However, the development of a new Human-UAV interaction using blinks and the quick integration of this system with the drone did prove that the architecture proposed is abstract enough to support different types of streaming data and algorithms to perform any kind of Human-drone communication.

Finally the work on simulation leads to the conclusion that it is possible to recreate (or newly create) near infinite scenarios where both humans and drones can quickly adapt and learn from each other. The augmented intelligence module can be quickly trained saving time and resources wasted in a normal field trial.

9.1 Future Work

Due to the magnitude of the Human-UAV symbiosis concept, there is a vast number of details that can be further explored. Over the following sections, the main objectives that could be improved on the current system, as well as new development ideas, are presented.

9.1.1 Integration of the development modules

Although the Explicit Communication module using motor movement and the Implicit Communication module using ECG are ready, it was not possible to implement them on the attended JIFX. The next interaction of this event could prove an excellent opportunity for another group member to do so. These could also be integrated into the simulation.

9.1.2 Simulation Test Cases

Two recreations were implemented during the span of this dissertation, but there is a lot more data available within the group to develop more scenarios. Unreal Engine is a really powerful tool that will is essential to the train and prepares both the human User and the UAV augmented intelligence module for the symbiosis.

9.1.3 Augmented Intelligence System module

Due to unexpected results, the communication modules took most of the time of this work and it was not possible to focus on the development of the augmented Intelligence System module. This block is an essential part of the symbiosis as it is what allows the UAV to evolve from being a tool to being a right partner and a 'Humans best friend'.

9.1.4 Transfer learning

Another interesting point to further explore is if all learning developed in simulation is transferable to the real world. The next iteration of the JIFX event can help to validate this idea.

9.1.5 CMU-Portugal Project

As stated in the Introduction, the group prepared and submitted a proposal to a CMU-Portugal call.

This proposal focuses on this same Human-Symbiosis concept and aims to develop further all work done within this dissertation. If this project is funded, this dissertation could serve as a starting point and most software developed can be used or adapted to build this symbiotic system.

References

[1] Licklider Joseph Carl Robnett. Man-Computer Symbiosis. *IRE transactions on human factors in electronics*, pages 4–11, 1960.

[2] John F. Keane and Stephen S. Carr. A brief history of early unmanned aircraft. *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, 32(3):558–571, 2013.

[3] Drone Industry Insights. The European Drone Industry Drone Industry Insights | Whitepaper |. (June), 2018.

[4] Jonathan Dandois, Marc Olano, Erle Ellis, Jonathan P. Dandois, Marc Olano, and Erle C. Ellis. Optimal Altitude, Overlap, and Weather Conditions for Computer Vision UAV Estimates of Forest Structure. *Remote Sensing*, 7(10):13895–13920, 10 2015. URL: http://www.mdpi.com/2072-4292/7/10/13895, doi:10.3390/rs71013895.

[5] Luis Mejias, Juan F. Correa, Ivan Mondragon, and Pascual Campoy. COLIBRI: A vision-Guided UAV for Surveillance and Visual Inspection. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2760–2761. IEEE, 4 2007. URL: http://ieeexplore.ieee.org/document/4209501/, doi:10.1109/ROBOT.2007.363883.

[6] Christoforos Kanellakis and George Nikolakopoulos. Survey on Computer Vision for UAVs: Current Developments and Trends. *Journal of Intelligent & Robotic Systems*, 87(1):141–168, 7 2017. URL: http://link.springer.com/10.1007/s10846-017-0483-z, doi:10.1007/s10846-017-0483-z.

[7] Ang Liu, Weiyu Mo, and Yonggen Ling. Multi-sensor environmental mapping, 3 2016.

[8] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner. Networked UAVs as aerial sensor network for disaster management applications. *e & i Elektrotechnik und Informationstechnik*, 127(3):56–63, 3 2010. URL: http://link.springer.com/10.1007/s00502-010-0717-2, doi:10.1007/s00502-010-0717-2.

[9] Gary B. Lamont, James N. Slear, and Kenneth Melendez. UAV Swarm Mission Planning and Routing using Multi-Objective Evolutionary Algorithms. In *2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, pages 10–20. IEEE, 4 2007. URL: http://ieeexplore.ieee.org/document/4222976/, doi:10.1109/MCDM.2007.369410.

[10] Axel Bürkle, Florian Segor, and Matthias Kollmann. Towards Autonomous Micro UAV Swarms. *Journal of Intelligent & Robotic Systems*, 61(1-4):339–353, 1 2011. URL: http://link.springer.com/10.1007/s10846-010-9492-x, doi:10.1007/s10846-010-9492-x.

[11] Sonia Waharte and Niki Trigoni. Supporting Search and Rescue Operations with UAVs. In *2010 International Conference on Emerging Security Technologies*, pages 142–147. IEEE, 9 2010. URL: http://ieeexplore.ieee.org/document/5600072/, doi:10.1109/EST.2010.31.

[12] WHO - World Health Organization. *Emergency Response Framework (ERF)*. 2013. URL: https://www.afro.who.int/publications/emergency-response-framework-erf.

[13] Patrick Doherty and Piotr Rudol. A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization. In *AI 2007: Advances in Artificial Intelligence*, pages 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. URL: http://link.springer.com/10.1007/978-3-540-76928-6_1, doi:10.1007/978-3-540-76928-6{_}1.

[14] A. Ryan and J.K. Hedrick. A mode-switching path planner for UAV-assisted search and rescue. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1471–1476. IEEE. URL: http://ieeexplore.ieee.org/document/1582366/, doi:10.1109/CDC.2005.1582366.

[15] Joana S. Paiva, Duarte Dias, and João P.S. Cunha. *Beat-ID: Towards a computationally low-cost single heartbeat biometric identity check system based on electrocardiogram wave morphology*, volume 12. 2017. doi:10.1371/journal.pone.0180942.

[16] Martin Fowler and Jim Highsmith. The Agile Manifesto (Article). *Software Development Magazine: The Lifecycle Starts Here*, (August):1–7, 2001. URL: www.martinfowler.com/articles/newMethodology.html.

[17] Mike Cohn. *User Stories Applied for agile software development*. Addison-Weslwy Professional, 2004.

[18] Ali Shoeb. Application of machine learning to epileptic seizure onset detection and treatment. pages 157–162, 2009. URL: http://dspace.mit.edu/handle/1721.1/54669.

[19] John G. Webster and John W. (John William) Clark. *Medical instrumentation : application and design*. John Wiley & Sons, 2010. URL: https://books.google.pt/books?id=muRmDwAAQBAJ&pg=PA10&lpg=PA10&dq=table+1.1+medical+and+physiological+parameters&source=bl&ots=xVGouRWjGV&sig=ACfU3U0c04aIQXaWoQp3AX4VcI40VBrtjQ&hl=en&sa=X&ved=2ahUKEwi7rvbexf7iAhUmAWMBHcl_DDAQ6AEwC3oECAkQAQ#v=onepage&q=tab.

[20] Thalía Harmony. The functional significance of delta oscillations in cognitive processing. *Frontiers in Integrative Neuroscience*, 7(December):1–10, 2013. URL: http://journal.frontiersin.org/article/10.3389/fnint.2013.00083/abstract, doi:10.3389/fnint.2013.00083.

[21] Wolfgang Klimesch. Eeg alpha and theta oscillations reflect.pdf. 1999. doi:10.1016/S0165-0173(98)00056-3.

[22] Ashley Craig, Yvonne Tran, Nirupama Wijesuriya, and Hung Nguyen. Regional brain wave activity changes associated with fatigue. *Psychophysiology*, 49(4):574–582, 2012. doi:10.1111/j.1469-8986.2011.01329.x.

[23] Wolfgang Klimesch. Alpha-band oscillations, attention, and controlled access to stored information. *Trends in Cognitive Sciences*, 16(12):606–617, 2012. URL: http://dx.doi.org/10.1016/j.tics.2012.10.007, doi:10.1016/j.tics.2012.10.007.

[24] Kazutaka Takahashi, Maryam Saleh, Richard D. Penn, and Nicholas G. Hatsopoulos. Propagating Waves in Human Motor Cortex. *Frontiers Human Neuroscience*, 5(April):1–8, 2011. URL: http://journal.frontiersin.org/article/10.3389/fnhum.2011.00040/abstract, doi:10.3389/fnhum.2011.00040.

[25] Xiaoxuan Jia and Adam Kohn. Gamma rhythms in the brain. *PLoS Biology*, 9(4):1–5, 2011. doi:10.1371/journal.pbio.1001045.

[26] Shlomit Yuval-Greenberg, Orr Tomer, Alon S. Keren, Israel Nelken, and Leon Y. Deouell. Transient Induced Gamma-Band Response in EEG as a Manifestation of Miniature Saccades. *Neuron*, 58(3):429–441, 2008. doi:10.1016/j.neuron.2008.03.027.

[27] M. Abo-Zahhad, Sabah M. Ahmed, and Sherif N. Abbas. A New EEG Acquisition Protocol for Biometric Identification Using Eye Blinking Signals. *International Journal of Intelligent Systems and Applications*, 7(6):48–54, 5 2015. URL: http://www.mecs-press.org/ijisa/ijisa-v7-n6/v7n6-5.html, doi:10.5815/ijisa.2015.06.05.

[28] George H Klem, Hans Otto Lüders, H H Jasper, and C Elger. The ten-twenty electrode system of the International Federation. *Electroencephalography and Clinical Neurophysiology*, 10(2):371–375, 1999. doi:10.1016/0013-4694(58)90053-1.

[29] Gonzalo M. Rojas, Carolina Alvarez, Carlos E. Montoya, María de la Iglesia-Vayá, Jaime E. Cisternas, and Marcelo Gálvez. Study of resting-state functional connectivity networks using EEG electrodes position as seed. *Frontiers in Neuroscience*, 12(APR):1–12, 2018. doi:10.3389/fnins.2018.00235.

[30] Valer Jurcak, Daisuke Tsuzuki, and Ippeita Dan. 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems. *NeuroImage*, 34(4):1600–1611, 2007. doi:10.1016/j.neuroimage.2006.09.024.

[31] Gian Emilio Chatrian, Ettore Lettich, and Paula L. Nelson. Modified nomenclature for the “10%” electrode system, 1988. doi:10.1097/00004691-198804000-00005.

[32] D F Marks and A R Isaac. Topographical distribution of EEG activity accompanying visual and motor imagery in vivid and non-vivid imagers. *British journal of psychology (London, England : 1953)*, 86 (Pt 2):271–82, 5 1995. URL: http://www.ncbi.nlm.nih.gov/pubmed/7795945.

[33] Mariana Bendersky. *Book Review: Illustrated Manual of Clinical Evoked Potentials*, volume 82. 2018. URL: https://academic.oup.com/neurosurgery/article/82/5/E121/4782237, doi:10.1093/neuros/nyx620.

[34] G. Pfurtscheller. Functional brain imaging based on ERD/ERS. *Vision Research*, 41(10-11):1257–1260, 2001. doi:10.1016/S0042-6989(00)00235-2.

[35] R Cantello, M Gianelli, C Civardi, and R Mutani. Magnetic brain stimulation: the silent period after the motor evoked potential. *Neurology*, 42(10):1951–9, 10 1992. URL: http://www.ncbi.nlm.nih.gov/pubmed/1407578, doi:10.1212/wnl.42.10.1951.

[36] Ali Haider and Reza Fazel-Rezai. Application of P300 Event-Related Potential in Brain-Computer Interface. In *Event-Related Potentials and Evoked Potentials*. InTech, 11 2017. URL: http://www.intechopen.com/books/event-related-potentials-and-evoked-potentials/application-of-p300-event-related-potential-in-brain-computer-interface, doi:10.5772/intechopen.69309.

[37] Emanuel Donchin and Michael G. H. Coles. Is the P300 component a manifestation of context updating? *Behavioral and Brain Sciences*, 11(03):357, 9 1988. URL: http://www.journals.cambridge.org/abstract_S0140525X00058027, doi:10.1017/S0140525X00058027.

[38] M D Comerchero and J Polich. P3a and P3b from typical auditory and visual stimuli. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 110(1):24–30, 1 1999. URL: http://www.ncbi.nlm.nih.gov/pubmed/10348317.

[39] G. Pfurtscheller and C. Neuper. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, 89(7):1123–1134, 7 2001. URL: http://ieeexplore.ieee.org/document/939829/, doi:10.1109/5.939829.

[40] Elaine N Marieb and Katja Hoehn. *Human anatomy & physiology*. 2007. doi:10.1007/BF00845519.

[41] B. Blankertz, K.-R. Muller, G. Curio, T.M. Vaughan, G. Schalk, J.R. Wolpaw, A. Schlogl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schroder, and N. Birbaumer. The BCI Competition 2003: Progress and Perspectives in Detection and Discrimination of EEG Single Trials. *IEEE Transactions on Biomedical Engineering*, 51(6):1044–1051, 6 2004. URL: http://ieeexplore.ieee.org/document/1300800/, doi:10.1109/TBME.2004.826692.

[42] B. Blankertz, K.R. Muller, D.J. Krusienski, G. Schalk, J.R. Wolpaw, A. Schlogl, G. Pfurtscheller, J.D.R. Millan, M. Schroder, and N. Birbaumer. The BCI Competition III: Validating Alternative Approaches to Actual BCI Problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):153–159, 6 2006. URL: http://ieeexplore.ieee.org/document/1642757/, doi:10.1109/TNSRE.2006.875642.

[43] P. Sajda, A. Gerson, K.-R. Muller, B. Blankertz, and L. Parra. A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):184–185, 6 2003. URL: http://ieeexplore.ieee.org/document/1214716/, doi:10.1109/TNSRE.2003.814453.

[44] Dinesh (Dinesh K.) Kumar, Sridhar Poosapadi Arjunan, and Behzad Aliahmad. *Fractals : applications in biological signalling and image processing*. URL: https://www.crcpress.com/

Fractals-Applications-in-Biological-Signalling-and-Image-Processing/Kumar-Arjunan-Aliahmad/p/book/9781498744218.

[45] T. Higuchi. Approach to an irregular time series on the basis of the fractal theory. *Physica D: Nonlinear Phenomena*, 31(2):277–283, 6 1988. URL: https://www.sciencedirect.com/science/article/pii/0167278988900814, doi:10.1016/0167-2789(88)90081-4.

[46] P.C. Petrantonakis and L.J. Hadjileontiadis. Emotion Recognition From EEG Using Higher Order Crossings. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):186–197, 3 2010. URL: http://www.ncbi.nlm.nih.gov/pubmed/19858033http://ieeexplore.ieee.org/document/5291724/, doi:10.1109/TITB.2009.2034649.

[47] Nurhan Gursel Ozmen, Levent Gumusel, and Yuan Yang. A Biologically Inspired Approach to Frequency Domain Feature Extraction for EEG Classification. *Computational and mathematical methods in medicine*, 2018:9890132, 2018. URL: http://www.ncbi.nlm.nih.gov/pubmed/29796060http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5896285, doi:10.1155/2018/9890132.

[48] M. Akin. Comparison of Wavelet Transform and FFT Methods in the Analysis of EEG Signals. *Journal of Medical Systems*, 26(3):241–247, 2002. URL: http://link.springer.com/10.1023/A:1015075101937, doi:10.1023/A:1015075101937.

[49] Christos E. Vasios, Leonardo M. Angelone, Patrick L. Purdon, Jyrki Ahveninen, John W. Belliveau, and Giorgio Bonmassar. EEG/(f)MRI measurements at 7 Tesla using a new EEG cap ("InkCap"). *NeuroImage*, 33(4):1082–1092, 12 2006. URL: http://www.ncbi.nlm.nih.gov/pubmed/17035045https://linkinghub.elsevier.com/retrieve/pii/S1053811906007968, doi:10.1016/j.neuroimage.2006.07.038.

[50] Firdaus Mohamed, Syed Faiz Ahmed, Zunaidi Ibrahim, and Sazali Yaacob. Comparison of Features Based on Spectral Estimation for the Analysis of EEG Signals in Driver Behavior. In *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, pages 1–7. IEEE, 8 2018. URL: https://ieeexplore.ieee.org/document/8477633/, doi:10.1109/ICASSDA.2018.8477633.

[51] Qibin Zhao, Liqing Zhang, Andrzej Cichocki, and Jie Li. Incremental Common Spatial Pattern algorithm for BCI. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2656–2659. IEEE, 6 2008. URL: http://ieeexplore.ieee.org/document/4634170/, doi:10.1109/IJCNN.2008.4634170.

[52] R Cunnington, R Iansek, J L Bradshaw, and J G Phillips. Movement-related potentials associated with movement preparation and motor imagery. *Experimental brain research*, 111(3):429–36, 10 1996. URL: http://www.ncbi.nlm.nih.gov/pubmed/8911937.

[53] Kai Keng Ang, Zheng Yang Chin, Haihong Zhang, and Cuntai Guan. Filter Bank Common Spatial Pattern (FBCSP) in brain-computer interface. *Proceedings of the International Joint Conference on Neural Networks*, pages 2390–2397, 2008. doi:10.1109/IJCNN.2008.4634130.

[54] Wei Yen Hsu. EEG-based motor imagery classification using enhanced active segment selection and adaptive classifier. *Computers in Biology and Medicine*, 41(8):633–639, 2011. URL: http://dx.doi.org/10.1016/j.compbiomed.2011.05.014, doi:10.1016/j.compbiomed.2011.05.014.

[55] C. Vidaurre, A. Schlogl, R. Cabeza, R. Scherer, and G. Pfurtscheller. Study of On-Line Adaptive Discriminant Analysis for EEG-Based Brain Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 54(3):550–556, 3 2007. URL: http://ieeexplore.ieee.org/document/4100850/, doi:10.1109/TBME.2006.888836.

[56] Mehrnaz Kh. Hazrati and Abbas Erfanian. An online EEG-based brain–computer interface for controlling hand grasp using an adaptive probabilistic neural network. *Medical Engineering & Physics*, 32(7):730–739, 9 2010. URL: http://www.ncbi.nlm.nih.gov/pubmed/20510641https://linkinghub.elsevier.com/retrieve/pii/S1350453310000901, doi:10.1016/j.medengphy.2010.04.016.

[57] Xiaomu Song, Suk-Chung Yoon, and Viraga Perera. Adaptive Common Spatial Pattern for single-trial EEG classification in multisubject BCI. In *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 411–414. IEEE, 11 2013. URL: http://ieeexplore.ieee.org/document/6695959/, doi:10.1109/NER.2013.6695959.

[58] A. Llera, V. Gómez, and H. J. Kappen. Adaptive Classification on Brain-Computer Interfaces Using Reinforcement Signals. *Neural Computation*, 24(11):2900–2923, 11 2012. URL: http://www.ncbi.nlm.nih.gov/pubmed/22845827http://www.mitpressjournals.org/doi/10.1162/NECO_a_00348, doi:10.1162/NECO{_}a{_}00348.

[59] John Cacioppo, Louis Tassinary, and Gary Berntson. *Handbook of Psychophysiology*. 2016. doi:https://doi.org/10.1017/9781107415782.

[60] Hugo Silva, André Lourenço, and Ana Fred. Device and method for continuous biometric recognition based on electrocardiographic signals, 2012.

[61] Foteini Agrafioti, Francis Minhthang Bui, and Dimitrios Hatzinakos. System and method for enabling continuous or instantaneous identity recognition based on physiological biometric signals, 2012.

[62] Johan Hjelm and Joakim Soderberg. Communication apparatus using biometrics, 2012.

[63] Daniel Lange H. Electro-Biometric Authentication, 2012.

[64] Moon-Seog Jun. System, terminal, and method for digital electrocardiogram authentication, 2015.

[65] Yummi Jin. Method for biometric human identification on electrocardiogram and ptt smartwatch system using the same method. 2016.

[66] Scott Baskerville and Sara Englad. System and method for biometric ECG verification of patient identity, 2016.

[67] João Paulo TRIGUEIROS DA SILVA CUNHA and Joana Isabel SANTOS PAIVA. BIOMETRIC METHOD AND DEVICE FOR IDENTIFYING A PERSON THROUGH

AN ELECTROCARDIOGRAM (ECG) WAVEFORM. 11 2017. URL: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2017187422.

[68] R. Bousseljot, D. Kreiseler, and A. Schnabel. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. *Biomedizinische Technik/Biomedical Engineering*, 40(s1):317–318, 7 2009. URL: https://degruyter.com/view/j/bmte.1995.40.issue-s1/bmte.1995.40.s1.317/bmte.1995.40.s1.317.xml, doi:10.1515/bmte.1995.40.s1.317.

[69] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation*, 101(23), 6 2000. URL: https://www.ahajournals.org/doi/10.1161/01.CIR.101.23.e215, doi:10.1161/01.CIR.101.23.e215.

[70] AutoQuad. URL: https://uav.berkeley.edu/2018/05/01/autoquad.html.

[71] UAVs-at-Berkeley/UnityDroneSim: Simulating a drone in Unity. URL: https://github.com/UAVs-at-Berkeley/UnityDroneSim.

[72] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In *Field and Service Robotics*, 2017.

[73] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Ijcai*, 14(2):1137–1145, 1995.

[74] Karl Rosaen. 2016-06-20 | scikit-learn Pipeline gotchas, k-fold cross-validation, hyperparameter tuning and improving my score on Kaggle's Forest Cover Type Competition | ML Learning Log. URL: http://karlrosaen.com/ml/learning-log/2016-06-20/.

[75] Qing-Song Xu and Yi-Zeng Liang. Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1):1–11, 4 2001. URL: https://www.sciencedirect.com/science/article/abs/pii/S0169743900001222, doi:10.1016/S0169-7439(00)00122-2.

[76] Vineet Maheshwari. KNN ALGORITHM AND IMPLEMENTATION FROM SCRATCH - Data Driven Investor - Medium. URL: https://medium.com/datadriveninvestor/knn-algorithm-and-implementation-from-scratch-b9f9b739c28f.

[77] J.A.K. Suykens and J. Vandewalle. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, 9(3):293–300, 1999. URL: http://link.springer.com/10.1023/A:1018628609742, doi:10.1023/A:1018628609742.

[78] Lapi. Reconhecimento de Padrões::Support Vector Machines. URL: http://www.lapix.ufsc.br/ensino/reconhecimento-de-padroes/reconhecimento-de-padroessupport-vector-machines.

[79] Irina Rish. An empirical study of the naive Bayes classifier. *IJCAI*, 2(22):41–46, 2001.

[80] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop*, pages 41–48, 1999.

[81] Rossella Spataro, Antonio Chella, Brendan Allison, and Vincenzo La Bella. Reaching and Grasping a Glass of Water by Locked-In ALS Patients through a BCI-Controlled Human Robot. *Frontiers in Human Neuroscience*.

[82] Donald L. Schomer and Fernando H. Lopes da Silva, editors. *Niedermeyer's Electroencephalography*, volume 1. Oxford University Press, 11 2017. URL: http://www.oxfordmedicine.com/view/10.1093/med/9780190228484.001.0001/med-9780190228484, doi:10.1093/med/9780190228484.001.0001.

[83] J.R. Wolpaw, H. Ramoser, D.J. McFarland, and G. Pfurtscheller. EEG-based communication: improved accuracy by response verification. *IEEE Transactions on Rehabilitation Engineering*, 6(3):326–333, 1998. URL: http://ieeexplore.ieee.org/document/712231/, doi:10.1109/86.712231.

[84] OpenBCI Cyton | OpenBCI Documentation. URL: https://docs.openbci.com/Hardware/02-Cyton.

[85] OpenBCI - Open Source Biosensing Tools (EEG, EMG, EKG, and more). URL: https://openbci.com/.

[86] Home - BESDATA. URL: http://www.besdatatech.com/index.php/Index/index_e.html.

[87] Witman Alvarado-Diaz, Peru Lima, Brian Meneses-Claudio, and Avid Roman-Gonzalez. Implementation of a brain-machine interface for controlling a wheelchair. In *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, pages 1–6. IEEE, 10 2017. URL: http://ieeexplore.ieee.org/document/8229668/, doi:10.1109/CHILECON.2017.8229668.

[88] Shivanthan A.C. Yohanandan, Isabell Kiral-Kornek, Jianbin Tang, Benjamin S. Mshford, Umar Asif, and Stefan Harrer. A Robust Low-Cost EEG Motor Imagery-Based Brain-Computer Interface. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, volume 2018, pages 5089–5092. IEEE, 7 2018. URL: http://www.ncbi.nlm.nih.gov/pubmed/30441485https://ieeexplore.ieee.org/document/8513429/, doi:10.1109/EMBC.2018.8513429.

[89] Matthew D Turner, Darryl H Burnet, and Jessica A Turner. Expanding EEG research into the clinic and classroom with consumer EEG systems. (February 2018), 2017. doi:10.13140/RG.2.2.10917.81126.

[90] Nicholas Polosky, Jithin Jagannath, Daniel O'Connor, Hanne Saarinen, and Svetlana Foulke. Artificial neural network with electroencephalogram sensors for brainwave interpretation: brain-observer-indicator development challenges. In *2017 13th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT)*, pages 1–6. IEEE, 11 2017. URL: http://ieeexplore.ieee.org/document/8263139/, doi:10.1109/CEWIT.2017.8263139.

[91] Catharina Zich, Maarten De Vos, Cornelia Kranczioch, and Stefan Debener. Wireless EEG with individualized channel layout enables efficient motor imagery training. *Clinical Neurophysiology*, 126(4):698–710, 4 2015. URL: http://www.ncbi.nlm.nih.gov/pubmed/25091344https://linkinghub.elsevier.com/retrieve/pii/S1388245714003708, doi:10.1016/j.clinph.2014.07.007.

[92] Gerwin Schalk, Dennis J Mcfarland, Thilo Hinterberger, Niels Birbaumer, Jonathan R Wolpaw, and A Brain-computer Interface B C I Technology. BCI2000 : A General-Purpose Brain-Computer Interface (BCI) System. *IEEE Trans Biomed Eng*, 51(6):1034–1043, 2004.

[93] Corina Schuster, Roger Hilfiker, Oliver Amft, Anne Scheidhauer, Brian Andrews, Jenny Butler, Udo Kischka, and Thierry Ettlin. Best practice for motor imagery: a systematic literature review on motor imagery training elements in five different disciplines. *BMC Medicine*, 9(1):75, 12 2011. URL: http://bmcmedicine.biomedcentral.com/articles/10.1186/1741-7015-9-75, doi:10.1186/1741-7015-9-75.

[94] H C Dijkerman, M Ietswaart, M Johnston, and R S MacWalter. Does motor imagery training improve hand function in chronic stroke patients? A pilot study. *Clinical Rehabilitation*, 18(5):538–549, 8 2004. URL: http://journals.sagepub.com/doi/10.1191/0269215504cr769oa, doi:10.1191/0269215504cr769oa.

[95] Jiapu Pan and Willis J. Tompkins. A Real-Time QRS Detection Algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 2007. doi:10.1109/tbme.1985.325532.

[96] Gari D. Clifford, Francisco. Azuaje, and Patrick. McSharry. *Advanced methods and tools for ECG data analysis*. Artech House, 2006. URL: https://dl.acm.org/citation.cfm?id=1213221.

[97] Hardware Specifications - Muse Developers. URL: http://developer.choosemuse.com/hardware-firmware/hardware-specifications.