

2016

Body swarm interface (BOSI) : controlling robotic swarms using human bio-signals

<https://hdl.handle.net/2144/17083>

Boston University

BOSTON UNIVERSITY
COLLEGE OF ENGINEERING

Thesis

**BODY SWARM INTERFACE (BOSI) : CONTROLLING ROBOTIC
SWARMS USING HUMAN BIO-SIGNALS**

by

AAMODH SURESH

B.Tech., National Institute of Technology Karnataka, Surathkal, 2014

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science

2016

© 2016 by
AAMODH SURESH
All rights reserved

Approved by

First Reader

Mac Schwager, PhD
Assistant Professor of Aeronautics and Astronautics
Stanford University

Second Reader

John Baillieul, PhD
Distinguished Professor of Mechanical Engineering
Distinguished Professor of Systems Engineering
Distinguished Professor of Electrical and Computer Engineering

Third Reader

Calin Belta, PhD
Professor of Mechanical Engineering
Professor of Systems Engineering
Professor of Bioinformatics

*Yesterday is History;
Tomorrow is a Mystery;
But, Today is a Gift
That is why its called the “Present”.*

Master Oogway (Kung Fu Panda)

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Prof. Mac Schwager. From the day I went up to him with my idea of "Controlling robots with my mind", I have received immense teaching, support, and encouragement from him, without which this thesis would not have even started. Even after he moved to Stanford midway, he has continued to show the same level of interest and support in my project, for which I am really grateful.

Then I would like to thank all my lab mates from the BU Robotics Lab and the Multi-Robot Systems Lab who have always made me feel welcome, and made me feel at home. From board game nights to lunch outings, my time at the lab would have been a lot less fun without you roboticists. I would also like to thank Prof. Calin Belta for giving me a desk at the BU Robotics Lab, which enabled me to connect with such wonderful and talented people. I would like to extend special thanks to Eric Cristofalo, who has been my lab partner even though we worked on completely different projects.

I am deeply indebted to the BU UAV team as well, their resources and technical expertise has greatly contributed towards this thesis. Special thanks to Gaurav Hirlekar for his help in writing codes in Nodejs to control the aerial drones. I am also grateful to Allen Huang for his insightful ideas and help in various aspects of my projects. I would also like to thank Joe Squeri for his help in developing Matlab codes for the Myo armband.

Lastly I would like to thank my family and friends, who have provided me continual love, encouragement, and energy throughout my studies in Boston. Special thanks to my parents, whose unconditional love and support has helped me complete this thesis.

Aamodh Suresh

Master of Science candidate

Mechanical Engineering Department

BODY SWARM INTERFACE (BOSI) : CONTROLLING ROBOTIC SWARMS USING HUMAN BIO-SIGNALS

AAMODH SURESH

ABSTRACT

Traditionally robots are controlled using devices like joysticks, keyboards, mice and other similar human computer interface (HCI) devices. Although this approach is effective and practical for some cases, it is restrictive only to healthy individuals without disabilities, and it also requires the user to master the device before its usage. It becomes complicated and non-intuitive when multiple robots need to be controlled simultaneously with these traditional devices, as in the case of Human Swarm Interfaces (HSI).

This work presents a novel concept of using human bio-signals to control swarms of robots. With this concept there are two major advantages: Firstly, it gives amputees and people with certain disabilities the ability to control robotic swarms, which has previously not been possible. Secondly, it also gives the user a more intuitive interface to control swarms of robots by using gestures, thoughts, and eye movement.

We measure different bio-signals from the human body including Electroencephalography (EEG), Electromyography (EMG), Electrooculography (EOG), using off the shelf products. After minimal signal processing, we then decode the intended control action using machine learning techniques like Hidden Markov Models (HMM) and K-Nearest Neighbors (K-NN). We employ formation controllers based on distance and displacement to control the shape and motion of the robotic swarm. Comparison for ground truth for thoughts and gesture classifications are done, and the resulting pipelines are evaluated with both simulations and hardware experiments with swarms of ground robots and aerial vehicles.

Contents

1	Introduction	1
1.1	Historical overview of control of robots	1
1.2	Motivation	2
1.3	Body Swarm Interface	3
1.3.1	Brain-Swarm Interface (BSI)	4
1.3.2	Arm-Swarm Interface (ASI)	5
1.4	Bio-signals and their Acquisition	6
1.4.1	Brain Signal Acquisition	7
1.4.2	Muscle Signal Acquisition	9
1.5	Neural Mechanisms Involved	10
1.6	Related Work	12
1.7	Thesis Outline	14
2	System Description	15
2.1	Problem Formulation	15
2.2	The Brain Swarm Interface Problem	16
2.2.1	Proposed solution framework	16
2.3	The Arm Swarm Interface Problem	17
2.3.1	Proposed solution framework	17
3	Decoding Swarm Control Parameters	20
3.1	Choosing the right algorithm	20
3.2	HMM: An Overview	21

3.2.1	The Training Phase with Baum-Welch Algorithm	23
3.2.2	Online Estimation with the Forward Algorithm	25
3.3	Thought Recognition	25
3.4	Gesture Recognition	27
3.5	Formation Recognition	29
3.6	Tracking Eye Movements	33
3.6.1	EOG : An Overview	33
4	Swarm Formation Control	40
4.1	Swarm control techniques overview	40
4.2	Potential Field based controller	42
4.2.1	System Description	43
4.2.2	A few key Notes	43
4.2.3	Controlling motion	45
4.3	Displacement based controller	46
5	Hardware involved and Signal Acquisition	47
5.1	EEG Signal Acquisition	47
5.1.1	Headset setup and signal acquisition	47
5.2	Signal Acquisition from MYO armband	49
5.3	Ground Robotic Platform	51
5.4	Aerial Drone Platform	52
6	Results	53
6.1	Brain Swarm Interface	53
6.1.1	Simulations	53
6.1.2	Hardware Experiments	54
6.2	Arm Swarm Interface	59
6.2.1	Accuracy Tests	59

6.2.2 Experiments	61
7 Summary and Conclusions	66
7.1 Summary of the thesis	66
7.2 Conclusion	66
7.3 Future Work	67
A Lyapunov stability analysis of potential function	68
References	70
Curriculum Vitae	77

List of Tables

6.1 Formation suite for variable number of drones used 60

List of Figures

1.1	The various research areas involved in implementing a BoSI is shown. . . .	4
1.2	Key components the system and their interaction. i): The user generates brain and eye movement signals. ii): The Emotiv Epoc headset records these signals from the scalp and wirelessly transmits them to the computer. iii): The computer receives the pose of the robots. iv): The computer decodes the eye movements and thoughts, and transmits control signals to the robots. v): The Optitrack captures pose of robots. vi): The user receives visual feedback.	5
1.3	Key components the system and their interaction. i). The user's arm movement is captured by the IMU in armband at 50 Hz. ii). EMG signals containing arm gestures are recorded at 200 Hz. iii). The computer receives IMU and EMG signals using Bluetooth .iv). The computer receives the pose of the quadrotors. v). The computer decodes the control parameters, and transmits control signals to the quadrotors. vi). The Optitrack sytem captures pose of robots. vii). The user receives visual feedback. viii). The user performs required arm movement and gestures.	6
1.4	Different measurement techniques for brain activity	8
1.5	Monopolar and Bipolar configurations for recording surface EMG. Source : http://www.nrsign.com/	9
1.6	The labelled cortex of the brain showing different functional areas.	11
2.1	Process Flow from input signals in the left to control of swarm on the right.	17

2.2	Process Flow from input signals in the left to control of swarm on the right.	18
3.1	A typical HMM model topology is shown. The components within the box indicate a Markov Process with just the states $Q(t)$	22
3.2	(a) Shows the 3 metrics from the EEG signal (red for 'Meditation', green for 'Excitement' and blue for 'Engagement') during the training period of 60 seconds. (b) Shows the estimated state of the HMM during the training period by plotting the color map of γ . The lower state corresponds to a "disperse" thought by the user, and the upper to an "aggregate" thought by the user.	26
3.3	(a) Shows the raw data captured during the training period of 120 seconds from the armband. (b) Shows the standard deviation of raw data over a 0.5s window with 0.05s overlap, used as input to Baum Welch algorithm. (c) Shows the state of the HMM during the training period by plotting the color map of γ .	28
3.4	(a) Shows the raw data captured during the trial period of 36 seconds from the armband. (b) Shows the standard deviation of raw data over a 0.5s window with 0.05s overlap, used as input to Baum Welch algorithm. (c) Shows the state of the HMM during the trial period by plotting the color map of α .	30
3.5	Gestures employed during drone and formation selection phase	31
3.6	(a)-(e) Steps to specify a triangle formation by using fist gesture for drone position and normal gesture during movement	32
3.7	EEG sensor placement on the human scalp using the 10-20 system. The locations highlighted in orange depict the locations used by the Emotiv Epoc headset (http://www.emotiv.com). We specifically read the red circled locations to get EOG signals for eye movement.	34

3·8	Horizontal Eye Movement Tracking. The main steps in Algorithm 1 are shown graphically from top to bottom. The bottom plots show the final extracted left and right eye movements.	36
3·9	Vertical Eye Movement Tracking. The main steps in Algorithm 2 are shown graphically from top to bottom. The bottom plots show the final extracted up and down eye movements.	37
5·1	The Emotiv epoc headset along with its technical specifications is shown. .	48
5·2	A screenshot of Emotiv application. Top-Left corner shown the contact quality information. Raw signals from 14 electrodes is shown in the right half of the picture.	49
5·3	Myo armband	50
5·4	M3pi robot platform with Mbed micro-controller and xbee radios.	51
5·5	Assorted parrot mini drones used for our experiments.	52
6·1	Simulation Results. a) Shows the color map of the variable α during the forward procedure. b) Shows the horizontal eye movement. c) Shows the vertical eye movement.	55
6·2	Path of the swarm (blue) along with its centroid (red) traveling in a simulated environment with 128 robots.	56
6·3	M3pi Robots in the Optitrack Arena.	57
6·4	Experimental area and setup.	57
6·5	Experiment Results. a) Shows the color map of the variable α during the forward procedure. b) Shows the horizontal eye movement. c) Shows the vertical eye movement. d) Shows the distance from centroid of the three robots in centimeter.	58
6·6	Position of the swarm of 3 robots along with their centroid during the experiment.	59

6.7	Accuracy test for five different HMM and K-NN implementations.	60
6.8	Gestures employed during drone and formation selection phase	63
6.9	(a)-(e) Steps to specify a triangle formation by using fist gesture for drone position and normal gesture during movement, (f) Experimental validation of triangle formation with three mini Parrot drones.	64
6.10	Specifying vertical line formation by using fist gesture for drone position and normal gesture during movement	64
6.11	Specifying rectangle formation by using fist gesture for drone position and normal gesture during movement	65

List of Abbreviations

ASI	Arm Swarm Interface
BCI	Brain Computer Interface
BSI	Brain Swarm Interface
EEG	Electroencephalography
EMG	Electromyography
EOG	Electrooculography
hBCI	Hybrid Brain Computer Interface
HCI	Human Computer Interface
HSI	Human Swarm Interface
IMU	Inertial Measurement Unit

Chapter 1

Introduction

1.1 Historical overview of control of robots

Control of Robots has been a popular aspect in both academic literature (Brooks, 1986) as well as science fictions (Asimov, 1950) over the ages, way before the existence of physical and functional robots like the ones we see today. Earliest form of remote control of devices can be traced back to the 19th century, where torpedos were controlled remotely using wires and mechanisms (Gray, 2004). With the development of wireless communication technology (Tse and Viswanath, 2005), devices could be controlled remotely without the use of wires, as long as they are within communication range. Arguably the first demonstration of wireless technology was given by Nikola Tesla in 1898, where he performed a show and moved a boat using tuned radio waves, for which he was granted a patent later on (Goldsmith, 2005). Technology has progressed significantly, now robots can be controlled on other planets as well (Rusch, 2012). Around the early 21th century, the idea of swarm robotics started becoming popular (Şahin, 2004). Instead of having a single, complex, bulky, and expensive robot, which was prone to failure, researchers were interested to solve the problem using multiple simple robots instead. This concept enhanced the reliability, cost effectiveness and robustness of the system. Most of the research is concentrated on autonomous control of these swarms, Human Swarm Interaction (HSI) (McLurkin et al., 2006) has relatively received less importance.

Recently, due to increasing technological development of computers and sensors, there has been a push towards the development of autonomous robots (Winner, 1977). Even with

all the efforts in making robots autonomous, there are still certain aspects and scenarios, for example disaster management (Murphy, 2004), covert ops (Marzouqi and Jarvis, 2004), and other areas where human control becomes necessary. So Human Robot Interaction (HRI) is a key area for research. However, most of the current interfaces rely on a joystick, keypad and other Human Computer Interaction (HCI) devices. The use of intuitive gestures as control inputs have been explored to control single (Waldherr et al., 2000) as well as swarms of robots (Alonso-Mora et al., 2015). Typically gestures are detected by using external cameras (Shimada et al., 1998), and these systems require additional setup and may not be practical for all scenarios.

Our work focuses on the use of bio-signals to control robotic swarms. This idea is novel and revolutionary since controlling swarms of robots by simply thinking or performing intuitive gestures is very appealing and intuitive. The ubiquity of bio-signals, and the vast improvements in sensor technology to acquire these signals makes the idea tangible and concrete. This idea has been explored in science fiction and in the movies (DeCandido, 2015), which has brought tremendous fascination and excitement amongst the public. We take a small step in bringing this futuristic fictional concept into reality.

1.2 Motivation

The pinnacle of evolution of Human-Robot Interaction lies in the nascent field of Neuro-Robotics where robots and prostheses are controlled by the bio-signals from the human body (Arbib et al., 2008). The recent advent of non-invasive techniques to capture these signals without any clinical procedures and restrictions makes this field more accessible and practical. Previously Human Swarm Interaction has been performed exclusively through HCI devices like joysticks, keyboards and cameras. Since a huge number of potential users are disabled, the current approach has been limiting. By integrating various aspects of Neuroscience, signal processing, machine learning, control theory, and robotics, we have

taken field of Robotics further by developing an intuitive and accessible interface to control swarms of robots.

Whereas the motivation for using bio-signals to operate prosthetics or wheelchair is evident, the applications for a body swarm interface may be less obvious. Firstly, people who are mobility-impaired may use a swarm of robots to manipulate their environment using a body-swarm interface. Indeed, a swarm of robots may offer a greater range of possibilities for manipulation than what is afforded by a single mobile robot or manipulator. For example a swarm can reconfigure to suit different sizes or shapes of objects (Murata and Kurokawa, 2007) , or to split up and deal with multiple manipulation tasks at once (Michael et al., 2008). Another motivation for our work is that using a body interface may unlock a new, more flexible way for people to interact with swarms. Currently human swarm interfaces are largely restricted to gaming joysticks with limited degrees of freedom. However, swarms typically have many degrees of freedom, and the brain and muscle activity together have an enormous potential to influence those degrees of freedom beyond the confines of a traditional joystick. We envision that this interface can eventually craft shapes and sizes for swarm, split a swarm into sub swarms, aggregate or disperse the swarm, and perhaps much more. In this work, we take a small step toward this vision.

1.3 Body Swarm Interface

In this thesis we present a new interface for a human user to control a swarm of robots, which we call a Body-Swarm Interface (BoSI). The BoSI uses various concepts from different research areas in science and engineering as illustrated in Fig. 1-1. The BoSI has two components: 1) The Brain-Swarm Interface (BSI), which uses brain signals and eye movement signals to control robotic swarm. 2) The Arm Swarm interface (ASI), which uses arm muscle signals and arm movement to control robotic swarm.

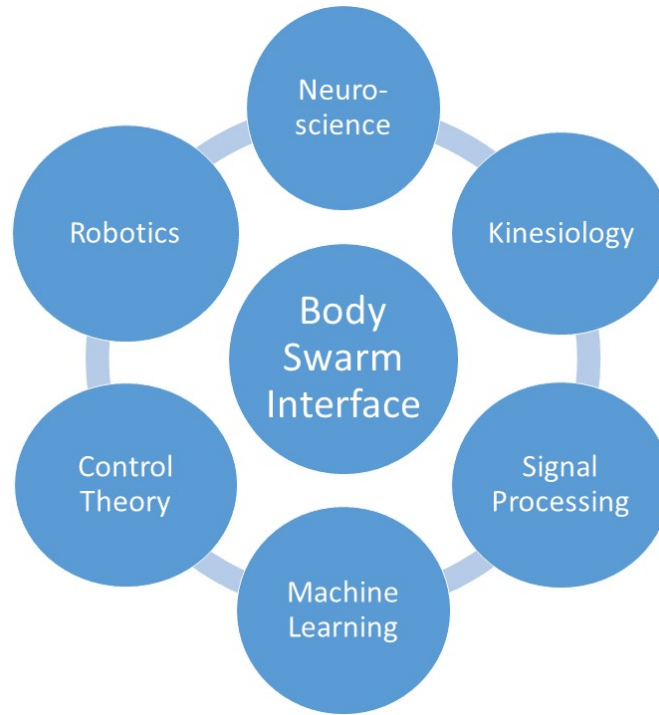


Figure 1-1: The various research areas involved in implementing a BoSI is shown.

1.3.1 Brain-Swarm Interface (BSI)

The BSI uses an off-the-shelf Electroencephalogram (EEG) headset to record brain and muscle activity from the user's scalp. We use both signals from brain activity, as well as signals from contracting muscles due to eye movement, to generate control signals for the robot swarm. We allow the user to control the dispersion/aggregation of the swarm, as well as the direction of motion of the swarm. The dispersion/aggregation is determined from the user's brain signals, and is decoded using an Hidden Markov Model (HMM) based method. The direction of motion is decoded from the user's eye movements with a multi-step signal processing algorithm. The robots maintain a cohesive swarm using a potential-field based swarm controller. Fig. 1-2 shows the overview of our BSI system used graphically. The system will be explained in detail in the following chapters.

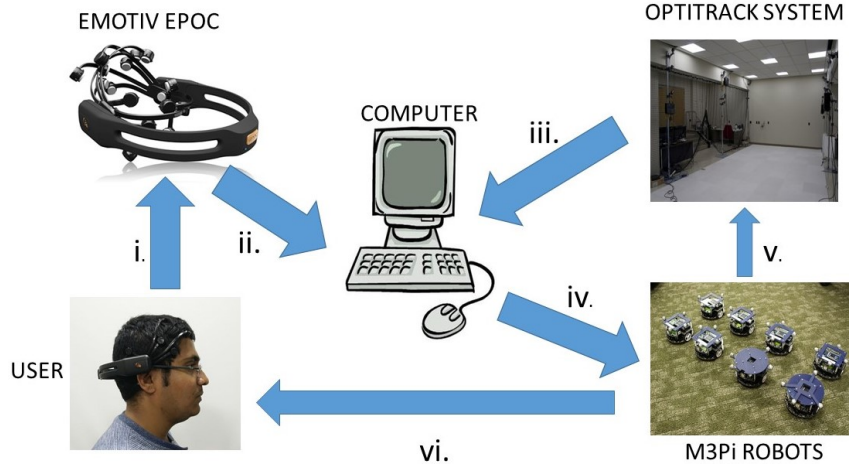


Figure 1-2: Key components the system and their interaction. i): The user generates brain and eye movement signals. ii): The Emotiv Epoc headset records these signals from the scalp and wirelessly transmits them to the computer. iii): The computer receives the pose of the robots. iv): The computer decodes the eye movements and thoughts, and transmits control signals to the robots. v): The Optitrack captures pose of robots. vi): The user receives visual feedback.

1.3.2 Arm-Swarm Interface (ASI)

The ASI uses a myo armband, which is an off-the-shelf Electromyographic (EMG) armband to record muscle activity, and inertial measurements from the user's arm. We use both signals from the muscle activity (EMG) in the arm, and the orientation of the arm measured by the magnetometer in the Inertial Measurement Unit (IMU), to generate control signals for the Quadrotor swarms. Hidden Markov Model (HMM) based classifier is running in the background on the EMG data to decode gesture, arm orientation is measured during the gesture from IMU data, and fed into a K-Nearest Neighbour (K-NN) classifier to decode the desired formation. We then use displacement based formation control techniques to control the motion and formation of the robotic swarm. The ASI system is briefly described graphically in Fig. 1-3. The following chapters will explain the components and processes in detail.

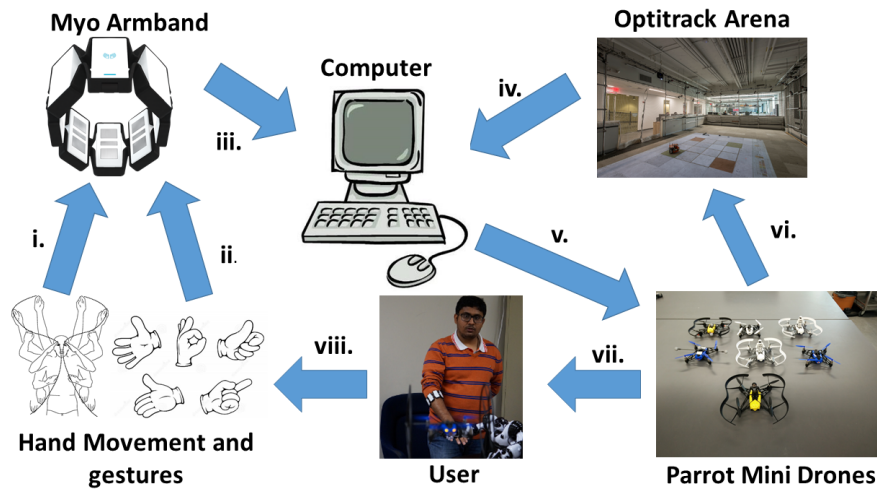


Figure 1.3: Key components the system and their interaction. *i).* The user's arm movement is captured by the IMU in armband at 50 Hz. *ii).* EMG signals containing arm gestures are recorded at 200 Hz. *iii).* The computer receives IMU and EMG signals using Bluetooth. *iv).* The computer receives the pose of the quadrotors. *v).* The computer decodes the control parameters, and transmits control signals to the quadrotors. *vi).* The Optitrack system captures pose of robots. *vii).* The user receives visual feedback. *viii).* The user performs required arm movement and gestures.

1.4 Bio-signals and their Acquisition

The human body has been an inspiration for many technologies, from the invention of electricity to development of complex learning algorithms, human body has been a source of inspiration and innovation (Benyus, 1997). Bio-signals have been observed and measured starting from 20 centuries ago (Liang et al., 2012). There are various bio-signals which can be measured and analyzed from the human body. Our work specifically deals with measuring brain signals and muscle activity. These signals can be measured by inserting electrodes inside the human body (Invasive techniques) or employ other imaging techniques from the surface of the body.

1.4.1 Brain Signal Acquisition

The human brain is the most complex organ known to mankind (Dayan and Abbott, 2005). Some of the techniques which are used to record brain activity are described as follows:

- **Electroencephalography (EEG)** : It records electrical activity of the brain along the scalp. EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain
- **Magnetoencephalography (MEG)**: is a functional neuro-imaging technique for mapping brain activity by recording magnetic fields produced by electrical currents occurring naturally in the brain, using very sensitive magnetometers. These are typically large equipment.
- **Functional Magnetic Resonance Imaging or functional MRI (fMRI)**: is a functional neuroimaging procedure using MRI technology that measures brain activity by detecting associated changes in blood flow. This technique relies on the fact that cerebral blood flow and neuronal activation are coupled. When an area of the brain is in use, blood flow to that region also increases.
- **Functional Near-Infrared Spectroscopy (fNIRS)** : is the use of NIRS (near-infrared spectroscopy) for the purpose of functional neuroimaging. Using fNIR, brain activity is measured through hemodynamic responses associated with neuron behavior. fNIR is a non-invasive imaging method involving the quantification of chromophore concentration resolved from the measurement of near infrared (NIR) light attenuation, temporal or phasic changes.

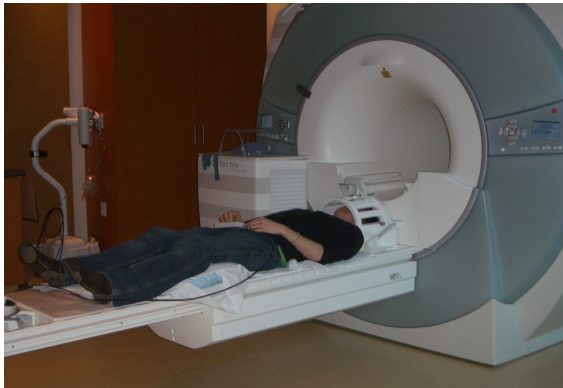
Fig. 1-4 shows some of the technology currently used to measure brain activity (Wolpaw and Wolpaw, 2012). MEG (Fig. 1-4(b)) and fMRI (Fig. 1-4(c)) clearly require expensive and complex equipment. Since fNIRS (Fig. 1-4(d)) is not widely available, we use



(a) A typical EEG cap used to measure electrical activity from the scalp



(b) A typical MEG setup to measure magnetic activity of brain

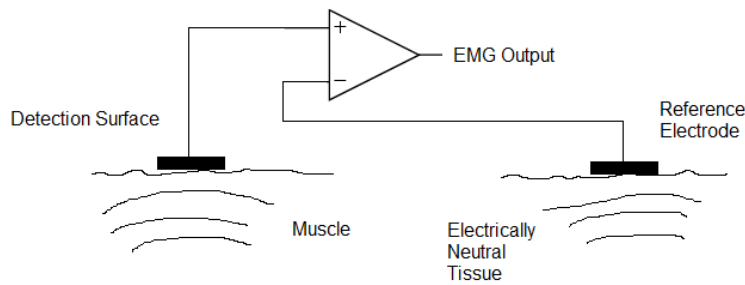


(c) A typical fMRI setup

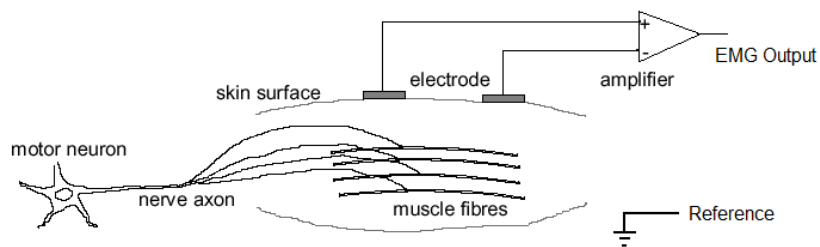


(d) An fNIRS hat used to measure brain activity

Figure 1.4: Different measurement techniques for brain activity



(a) Monopolar EMG recording configuration



(b) Bipolar EMG recording configuration

Figure 1-5: Monopolar and Bipolar configurations for recording surface EMG.Source : <http://www.nrsign.com/>

a non-invasive EEG technique (Fig. 1-4(a)) to measure brain activity, since it is the most accessible and practical, and there are off the shelf devices to measure EEG signals.

1.4.2 Muscle Signal Acquisition

Traditionally the electrical signals due to muscle flexion and contraction were measured by inserting needle electrodes into the skin, and were primarily reserved for medical purposes to treat and diagnose muscle disorders (Lee and Saridis, 1984). The signals measured were called Electromyography (EMG), since they were essentially measuring electrical signals from contracting muscles. But with the improvement in sensor technology and electronics, invasive EMG is rarely used and the sensors and electrodes are placed on the surface of the skin to record electric potentials. There are two major configurations in which this can be done: *Monopolar* and *bipolar*.

In case of monopolar configuration, one target electrode is placed on the skin over the target muscle, and the other electrode, which serves as the reference electrode is placed somewhere farther away. The signal between these two electrodes is amplified and recorded to observe muscle activity as shown in Fig. 1.5(a).

In case of bipolar configuration two target electrodes are placed on the skin over the target muscle within 1-2 cm from each other, and one electrode, which serves as the reference electrode is placed somewhere farther away. The signal between the two target electrodes is amplified differentially respect to the reference electrode and recorded to observe muscle activity as shown in Fig. 1.5(b). The advantage in this configuration is the common noise between the two electrodes is eliminated and hence we have a cleaner EMG signal with higher signal to noise ratio. For further details and information can be found in (Merletti and Parker, 2004).

The MYO armband which we use is a bipolar surface EMG acquisition device. More about the device and its signal acquisition will be explained in detail in Chapter 5

1.5 Neural Mechanisms Involved

In this section we would like to like to briefly discuss the neural mechanisms involved in a typical human brain, which would give the reader a concise background on EEG related Neuroscience and a better insight about our work.

The Brain is the most complex organ known to mankind. Though there are several aspects of the brain, we will limit our focus to the cortex, which is a folded structure forming the outer layer of the brain as shown in Fig. 1.6. Each of the areas in the cortex have different functions and there will be different signals emanating from them.

The cortex is believed to be folded uniquely among every individual, similar to fingerprints. However, There are many neural mechanisms which are common among all humans and are widely used in Brain-Computer Interface (BCI) applications. Since we are using

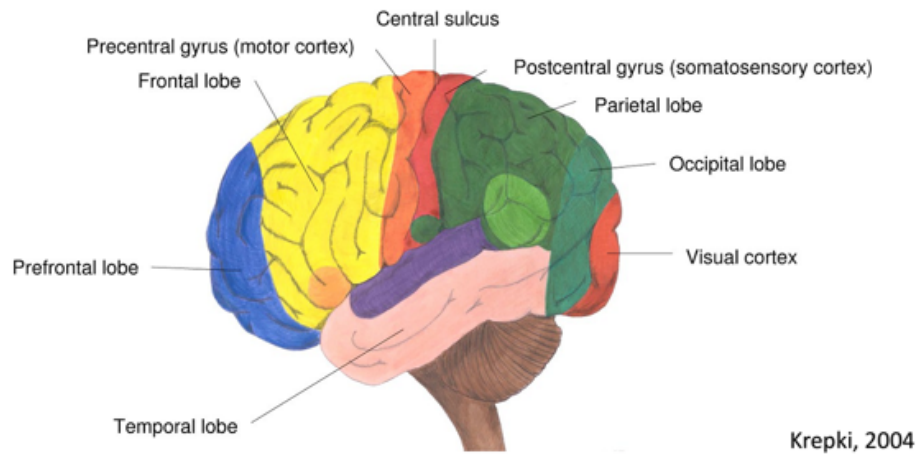


Figure 1-6: The labelled cortex of the brain showing different functional areas.

the performance metrics calculated by the Emotiv software, we will briefly describe the underlying neural mechanism specific to these metrics.

Since, EEG measurement from the surface of the scalp measures only large scale neural activation, there needs to be millions of neurons firing synchronously to detect the small signal coming from the brain. Hans Berger in the early 20th century observed a synchronous firing of neural activity at about 10 Hz, when the subject is at rest. This was the first time brain signals were measured, and he was credited for the discovery of Alpha Waves (Palva and Palva, 2007) and subsequently the EEG technique. Such oscillations are exhibited by all people and vary in different regions of the brain, depending on the activity performed by the subjects. To observe these oscillations the user needs to observe different frequency spectra to detect changes in level of activity.

Neuroscience research (Rao, 2013) has given us important frequency spectra to analyze :

1. **Delta 0.1-3 Hz**- Subjective feeling states: deep, dreamless sleep, non-REM sleep, trance, unconscious, Associated tasks and behaviors: lethargic, not moving, not attentive
2. **Theta 4-7 Hz** - Subjective feeling states: intuitive, creative, fantasy, imagery, cre-

ative, dreamlike, Associated tasks and behaviors: creative, intuitive, distracted, unfocused

3. **Alpha 8-12 Hz** - Subjective feeling states: relaxed, not agitated, tranquil, conscious
Associated tasks and behaviors: meditation, no action
4. **Beta 13-30 Hz** - Subjective feeling states: thinking, aware of self and surroundings, alertness
Associated tasks and behaviors: mental activity, planning.
5. **Gamma 30-50 Hz** - Subjective feeling states: thinking; integrated thought
Associated tasks and behaviors: high-level information processing.

So Emotiv exploits these signals to generate their performance metrics. For example for calculating the level of meditation, we can do a Fast Fourier Transform (FFT) to get the power distribution in each frequency band to determine the power level in the alpha band which correlates to the level of meditation. Other metrics can be found out similarly. Since Emotiv has already taken the trouble to calculate these levels and they work reasonably well, we will use their performance metrics for our application.

1.6 Related Work

The primary applications of using bio-signals in robotics has been in controlling assistive devices and rehabilitation engineering like wheelchairs and prosthetics. Leeb et al. (Leeb et al., 2007) have used a BCI paradigm to enable a tetraplegic to control a wheelchair. There have been few other works which have used various BCI paradigms to control wheelchairs (Galán et al., 2008; Li et al., 2013; Turnip et al., 2015). Control of prosthetic arms and legs are another important application of BCI technology (Santhanam et al., 2006; Hochberg et al., 2006). BCI technology has also been used for developing spellers (Cecotti, 2010) and other communication platforms (Wolpaw et al., 2002). The primary research applications

for EMG technology has been in developing prosthetics and exoskeletons. Kazerooni et al. (Kazerooni, 1989) used the concept of EMG signals to build robotic body support structures to enhance the performance capabilities of normal humans. Several other works have focused on improving control of EMG prosthetics (Zecca et al., 2002; Nishikawa et al., 1999) and also enable individual finger movements (Khushaba et al., 2012; Tenore et al., 2007) to make the interface more intuitive.

Comparatively very few works have concentrated on applying EMG gesture recognition in controlling robots other than assistive devices. Gestures can be a very intuitive way to communicate with, and control robots. Most of the previous research in this domain is concentrated on using signals from cameras and/or IMUs to detect gestures (Faudzi et al., 2012; Tsui et al., 2007). Wolf et al. (Wolf et al., 2013; Stoica et al., 2012) have demonstrated the use of custom EMG sensors to control a single robot by performing a variety of gestures. Human Swarm Interfaces have primarily used external hardware like joysticks and cameras. (Zhou and Schwager, 2016; Saget et al., 2008) to control swarms of aerial vehicles. Alonso et al. (Alonso-Mora et al., 2012) have used interactive multimedia and communication to perform Human Swarm Interaction (HSI). There have been a few HSIs using vision based gesture decoding (Nagi et al., 2011; Podevijn et al., 2013; Monajjemi et al., 2013) . Our method is based on intuitive gestures, for which the user requires much lesser training and minimal setup time. EMG signals can also be harnessed from people with certain disabilities, which makes our platform more accessible as compared with joysticks and other methods.

The idea of controlling a single robot using a Brain Computer Interface (BCI) has been explored in a few works: Bell et al. (Bell et al., 2008) have controlled a humanoid robot, Akce et al. (Akce et al., 2010) have used a BCI to steer a fixed wing unmanned aerial vehicle (UAV), and Lafleur et al. (LaFleur et al., 2013) have flown a single quadcopter using BCI. Vourvopoulos et al. (Vourvopoulos and Liarokapis, 2012) have used off-the-

shelf headsets like ours and navigated a single robot in both real and virtual environment.

The notion of using human bio-signals to control swarms of robots is relatively a new concept. Karavas et al. (Karavas and Artemiadis, 2015) used the concept of Brain Swarm Interface to study the human perception of swarm motion, by recording brain activity while the subject sees a swarm changing parameters over time. However, we are actively controlling a swarm with brain signals. Stoica et al. (Stoica et al., 2014) have introduced the idea of controlling quadrotor swarms with EMG signals, but our work differs in both the theoretical framework and practical application.

1.7 Thesis Outline

The thesis is organized as follows:

We first present a theoretical and practical overview of the various concepts involved in constructing our interface in Chapter 1. We then mathematically define our problem statement and suggest solving it in two different ways, which we refer to as the Arm-Swarm Interface and the Brain-Swarm Interface in Chapter 2. We describe the signal processing and machine learning pipeline in detail in Chapter 3, which we use to decode thoughts, gestures, formations, and eye movement. We then move onto formation control in Chapter 4, we illustrate and apply two different techniques based on relative displacements to control the swarms for both the ASI and the BSI application. The hardware used for signal acquisition, and the robotic platforms used for experiments are described in detail in Chapter 5. We discuss our simulations and experimental results which validate our framework in Chapter 6. Chapter 7 concludes the thesis with summary, conclusion, and future work.

Chapter 2

System Description

This chapter defines the system used for the Body Swarm Interface. We give the problem formulation and description to give the user an intuitive and mathematical insight to our problem.

2.1 Problem Formulation

The goal of this interface is to allow the user to control a swarm of robots according to the user's intention. Let the position of the swarm be described by the vector $x(t) = [x_1^T(t), \dots, x_M^T(t)]^T \in \mathbb{R}^{nM}$ where M denotes the number of agents in an n dimensional Euclidean space. The position of the i^{th} individual at time t is described by $x_i(t) \in \mathbb{R}^n$. Suppose the user has an intended trajectory for the swarm, which we denote as $x^*(t)$. We acquire some bio-signals from the user's body, which can be denoted by $U(t) \in \mathbb{R}^N$, where N is the number of sensors on the acquisition device. Then our problem is to design a framework which receives the input signals $U(t)$ from the acquisition device, and using some set control parameters say $\Theta \in \mathbb{R}$ changes the current swarm position $x(t)$ to the user intended swarm position $x^*(t)$. We state our problem as follows.

Problem 1 (Body-Swarm Interface) *Design a pipeline to determine the swarm control parameters $\Theta(t)$ from the bio-signals $U(t)$, so that the swarm trajectory approaches the user's intended swarm trajectory, $x(t) \rightarrow x^*(t)$.*

Now we solve this problem using two different interfaces which are described as follows.

2.2 The Brain Swarm Interface Problem

Using the notations defined in the previous section we will describe the problem specific to Brain Swarm Interface and the framework involved to solve Problem 1. When thinking about the user intended trajectory $x^*(t)$, the user's EEG headset records signals at time t , denoted by $U(t) \in \mathbb{R}^N$, where N is the number of sensors on the EEG headset. Let Θ be a vector of control parameters of the system given by $\Theta(t) = (a(t), b(t), v(t))$ where $a(t)$, $b(t) \in \mathbb{R}$ denote the attraction and repulsion gains, respectively, which are used to control the size of the swarm, and $v(t) \in \mathbb{R}^n$ denotes the displacement vector for controlling the movement of the swarm.

2.2.1 Proposed solution framework

To control the swarm, we adapt a potential field based swarm controller of a type that is common in controlling swarms of ground robots (Gazi and Passino, 2004; Olfati-Saber and Murray, 2002; Howard et al., 2002). With this type of controller, the system dynamics are given by

$$\dot{x}_i = \sum_{j=1, j \neq i}^M f_{a,b}(x_i, x_j) + v, \quad (2.1)$$

where $i = 1, 2, \dots, M$ and $f_{ij} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a function which depends on pairwise interactions between agents i and j , and has parameters a , b , and v .

In our proposed solution, we first train an HMM, and use it to determine the “thought state” of the user $Q(t) \in \{1, 2\}$, where for example, $Q = 2$ indicates “aggregation” and $Q = 1$ indicates “dispersion.” Then we map this thought state Q_t to values for $\{a(t), b(t)\}$ to control the aggregation/dispersion of the swarm. Secondly, we design an eye movement classifier which takes EEG signals $u(t) \subset U(t)$, where $u(t) \in \mathbb{R}^k$ with $k < N$, as input to determine the user's intended motion of the swarm, producing a control parameter $v(t)$ for the swarm. This pipeline is shown graphically in Fig. 2-1, and its components are described in more detail in the following chapters.

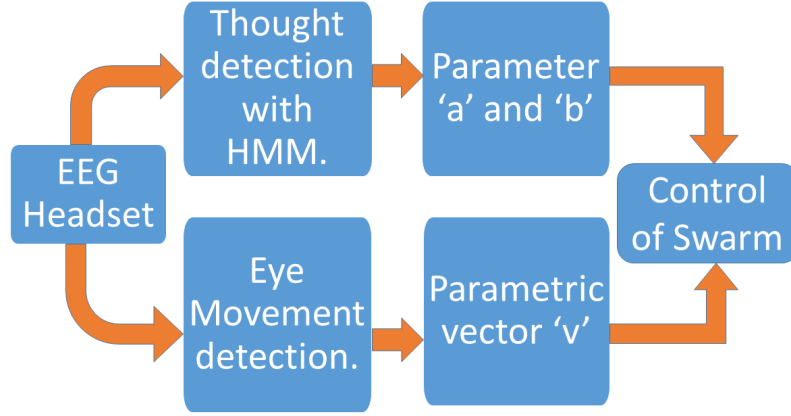


Figure 2.1: Process Flow from input signals in the left to control of swarm on the right.

2.3 The Arm Swarm Interface Problem

The user's armband produces IMU and EMG signals at time t . EMG signals are denoted by $U(t) \in \mathbb{R}^N$, where N is the number of sensors on the EMG armband. The signals from the magnetometer in the IMU is denoted by $V(t) \in \mathbb{R}^4$, which is a vector of quaternions. Let Θ be a vector of control parameters of the system given by $\Theta(t) = (M, \star p, v(t))$, where M denotes the number of agents, $\star p \in \mathbb{R}^{n \times M}$ denotes the formation matrix specifying the desired formation, and $v(t) \in \mathbb{R}^n$ denotes the displacement vector for controlling the movement of the swarm.

2.3.1 Proposed solution framework

To control the swarm, we adapt a displacement based formation control that is common in controlling swarms of aerial vehicles (Oh et al., 2015). Using single integrator model the system dynamics are given by

$$\dot{x}_i = u_i, \quad (2.2)$$

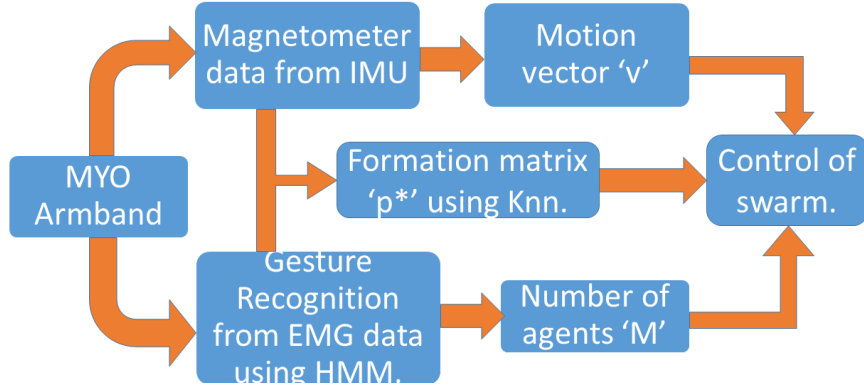


Figure 2.2: Process Flow from input signals in the left to control of swarm on the right.

where $i = 1, 2, \dots, M$. The desired formation can be specified as ,

$$E_p := \{p : x_j - x_i = \star p_j - \star p_i, i, j \in \{1, \dots, M\}\} \quad (2.3)$$

Our proposed solution is divided into three parts :

1. We first determine the number of drones M the user wants to control by training an HMM to recognize the number of fingers the user is showing. Arm gesture of the user $G(t) \in \{1, \dots, m\}$, where $m = 4$ in this case, corresponding to the maximum number of drones the user can control. For example, if the user is sticking out three fingers, then the HMM should assign $G(t) = '3'$, and the user controls three drones.
2. Next we enter the formation selection phase. We use both EMG and IMU data to determine formation of drones. During this phase $m = 2$ is used. $G = 1$ indicates “normal” gesture, and when $G = 2$ which indicates “Fist” gesture, the magnetometer readings are recorded and stored. These recordings are fed to a Knn trained classifier, from which we get our desired formation $\star p$.
3. Lastly, during flight the user controls the movement of the swarm $v(t)$. Magnetometer data $v(t)$ is recorded and the motion vector $v(t)$ is decoded from it.

This pipeline is shown graphically in Fig. 2-2, and its components are described in more detail in the following chapters.

Chapter 3

Decoding Swarm Control Parameters

This chapter describes the machine learning pipeline used to decode gestures and thoughts from the bio-signals acquired, and also outlines our custom signal processing pipeline implemented to extract eye movement from raw EEG signals. There are a plethora of machine learning and classification algorithms used widely in the literature, and it is very important to choose the right algorithm to ensure successful decoding.

3.1 Choosing the right algorithm

Machine Learning algorithms can be divided into supervised learning algorithms, semi supervised learning algorithms, and unsupervised learning algorithms. Based on the stochastic nature of our signals it would be a arduous task to use supervised learning algorithms, where we would have to generate large amounts of training data sets to train and implement the algorithm. So we were looking at techniques which did not require supervised learning.

However, in the case of classifying eye movement signals, we rely on simplicity and intuition to design a signal processing pipeline instead of a machine learning algorithm. We believe that using simple heuristics are sufficient in this case, and machine learning would prove wasteful in terms of computation power and time. Our method is outlined later in this Chapter.

We were interested in transforming EMG signals and EEG performance metrics to gestures and thoughts respectively. After a literature review,. We adopted Hidden Markov Models based on a survey by F. Lotte et al. (Lotte et al.,) to train and classify the per-

formance metrics signals and EMG signals. Previously, researchers have used HMMs for different BCI applications. Pfurtscheller et al. (Obermaier et al., 2001) used an HMM to classify EEG data, and HMMs have been used in conjunction with other techniques (Lee and Choi, 2003) as well. Aziz et al. (Aziz et al., 2014) have used HMM to classify EEG and EOG signals to generate control inputs for a wheelchair. Wissel et al. (Wissel et al., 2013) have used HMM in conjunction with SVM for classification of ECoG signals. In the following section we describe the theory behind HMMs, then in the later sections we illustrate its application to both our Arm-Swarm Interface and Brain-Swarm Interface pipelines. Using standard terminology and notation for HMMs, (Rabiner, 1989), (Bilmes, 1997), here we will describe our system in detail.

3.2 HMM: An Overview

A Hidden Markov Model is a joint probabilistic model of a collection of discrete random variables $O = O_1, \dots, O_T$ and $Q = Q_1, \dots, Q_T$ described by:

$$P(O, Q) = P(Q_0) \prod_{t=1}^T P(Q_t | Q_{t-1}) \prod_{t=0}^T P(O_t | Q_t) \quad (3.1)$$

Where Q_t denotes the state of the system at time t , and O_t denotes the observation of these hidden states at time t . It is an extension of Markov Models, in which the states of the system are directly observable and known. In the case of HMM these states are hidden and only noisy observations of these states are available to us. This is graphically illustrated in Figure 3-1.

Many algorithms exist for both learning the transition probabilities and observation probabilities of such a system from data, and determining a likely sequence of states hidden $\{Q_t\}$ from data. The observations of these states $O_t \in \mathbb{R}^l$ can be discrete or continuous.

The main components of the model from Equation 3.1 are:

- $P(Q_0)$ which is the initial state probability distribution and is represented by $\pi_i =$

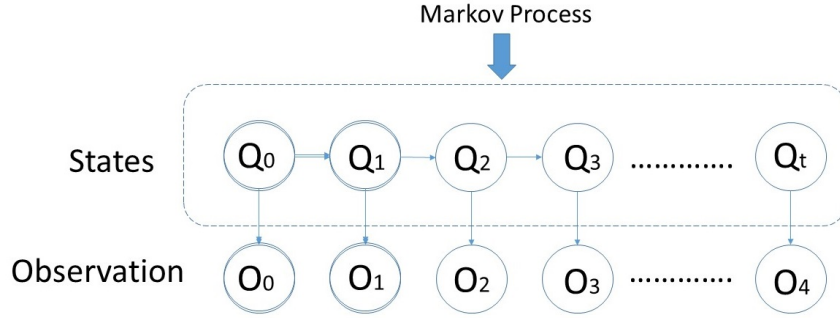


Figure 3-1: A typical HMM model topology is shown. The components within the box indicate a Markov Process with just the states $Q(t)$

$$P(Q_1 = i) \text{ where } i \in \{1, \dots, m\},$$

- $P(Q_t | Q_{t-1})$ is the state transition probability represented by the matrix $X = \{x_{ij}\} = P(Q_t = j | Q_{t-1} = i)$ where $i, j \in \{1, \dots, m\}$
- $P(O_t | Q_t)$ is the observation probability represented by the gaussian distribution $B_i(o_t) = P(O_t = o_t | Q_t = i) = \frac{\exp\{-1/2(o_t - \mu_i)^T R_i^{-1}(o_t - \mu_i)\}}{\det \sqrt{2\pi R_i}}$ with means $\mu_i \in \mathbb{R}^l$ and covariance matrix $R = \{r_{i,j}\} \in \mathbb{R}^{l \times l}$.

These are the model parameters of an HMM. We can see that the model can be completely described by parameters

$$\theta = \{\pi_i, X, \mu_i, R\}$$

The initial phase involves learning these parameters θ to generate the model using training data consisting of observations of the expected state space. We employ the Baum-Welch algorithm, a version of Expectation Maximization (EM)) to train the model parameters. After the training phase we use the Forward Algorithm to estimate the state of the model online using the current observations.

3.2.1 The Training Phase with Baum-Welch Algorithm

Training data consisting of the three metrics mentioned before is recorded in a single trial. During the training period the user repeats two thoughts through a pre-defined switching sequence. We enforced this thought sequence using a timed slide presentation which the user observed during training. This training data is fed into the Baum-Welch algorithm which estimates the parameters θ for the system.

The EM Algorithm is a two step iterative process (Expectation followed by Maximization), which can be expressed in the single expression

$$\theta_{c+1} = \underset{\theta}{\operatorname{argmax}} E_{P(Q,O|\theta_c)}[\mathcal{L}(Q,\theta)], \quad (3.2)$$

where $\mathcal{L}(Q,\theta) = \ln P(Q,O|\theta)$ is the log likelihood function, Q is the unobserved state, θ is the unknown parameters of the model, and O is the observed variable. The expectation step can be summarized by calculating the following quantities:

$$\alpha_i(t) = P(O_1 = o_1, \dots, O_t = o_t, Q_t = i | \theta), \quad (3.3)$$

where $\alpha_i(t)$ is known as the forward variable, and is the probability of ending in state i and seeing the partial observations $\{o_1, \dots, o_t\}$ given the model parameters θ , and

$$\beta_i(t) = P(O_{t+1} = o_{t+1}, \dots, O_T = o_T | Q_t = i, \theta), \quad (3.4)$$

where $\beta_i(t)$ is known as the backward variable, and is the probability of observing partial sequences $\{o_{t+1}, \dots, o_T\}$ given the model parameters and state at time t .

With α and β we can compute the forward-backward variable γ

$$\gamma_i(t) = P(Q_t = i | O, \theta) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^m \alpha_j(t)\beta_j(t)}, \quad (3.5)$$

where $\gamma_i(t)$ is the probability of being in state i given the model parameters and observa-

tions. Now we can compute

$$\begin{aligned}\zeta_{ij}(t) &= \frac{P(Q_t = i, Q_{t+1} = j, O | \theta)}{P(O | \theta)} \\ &= \frac{\alpha_i(t) x_{ij} B_j(o_{t+1}) \beta_j(t+1)}{\sum_{i=1}^m \sum_{j=1}^m \alpha_i(t) x_{ij} B_j(o_{t+1}) \beta_j(t+1)},\end{aligned}\tag{3.6}$$

where $\zeta_{ij}(t)$ is the probability of being in state i at time t and state j at time $t+1$.

Using the above defined quantities in the expectation step we can estimate the parameters θ in the maximization step of our system as follows :

$$\mu_i^{p+1} = \frac{\sum_{t=0}^T \gamma_i^p(t) o_t}{\sum_{t=0}^T \gamma_i^p(t)}\tag{3.7}$$

$$R_i^{p+1} = \frac{\sum_{t=0}^T \gamma_i^p(t) (o_t - \mu_i^{p+1}) (o_t - \mu_i^{p+1})^T}{\sum_{t=0}^T \gamma_i^p(t)}\tag{3.8}$$

$$X_T^{p+1} = \frac{\sum_{t=1}^T \zeta_{ij}^p(t-1)}{\sum_{t=1}^T \gamma_j^p(t-1)}\tag{3.9}$$

$$\pi_i^{p+1} = \gamma_i^p(0).\tag{3.10}$$

Naturally, for this procedure to start we need θ_0 which is the set of initial model parameters from which recursion begins according to Eqn. (3.2). We adopt a K-means clustering approach to initialize the model parameters, specifically the mean matrix μ . We used a g class K-means approach with the observations O as inputs, which give us the centroids of the g classes that we used to initialize μ . The number of classes g varies with the application, with $g = 2$ for thought recognition in BSI, and upto $g = 5$ in case of implementation of gesture recognition in ASI. The other parameters of θ are initialized randomly. We stop the iterative process when we observe only minute changes (order of 10^{-4}) in the estimated parameters from their previous estimations.

3.2.2 Online Estimation with the Forward Algorithm

After the model parameters θ have been estimated we can employ the HMM for online estimation of the state. However, we cannot immediately proceed to online estimation. First, we have to relate the HMM states back to the original thoughts. For example in case of thought recognition specifically, we do not know whether $Q = 1$ means an “aggregate” thought and $Q = 0$ means a “disperse” thought, or visa versa. We assign the abstract states to meaningful thoughts by examining the order of thoughts visited by the user during training to the $\gamma(t)$ value calculated during the training. Which ever assignment makes the γ sequence best match the thought sequence is the chosen assignment.

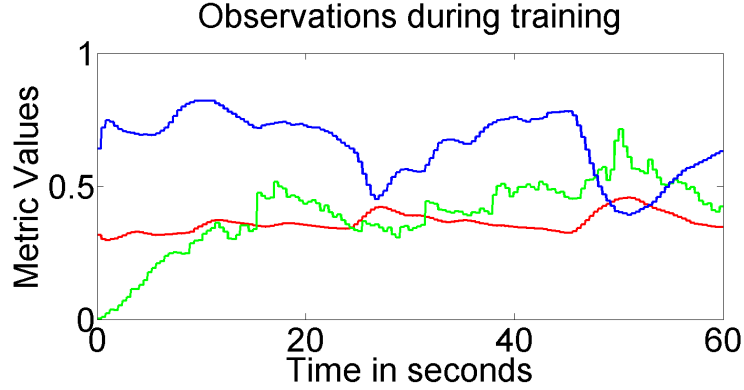
Online estimation of state is now a straightforward application of the Forward algorithm for HMMs using the learn parameters from the training phase. This allows us to find the most likely sequence of the desired states on line as a streaming signal arrives from the EEG. All we need for this phase is to calculate the value of $\alpha_i(t)$ from Eqn.3.3 for the current time t for all the states i and determine the most probable state at that time. So we can describe the control output from the HMM at time t given by $h(t) \in \{1, \dots, m\}$ as

$$h(t) = \underset{i}{\operatorname{argmax}} \alpha_i(t). \quad (3.11)$$

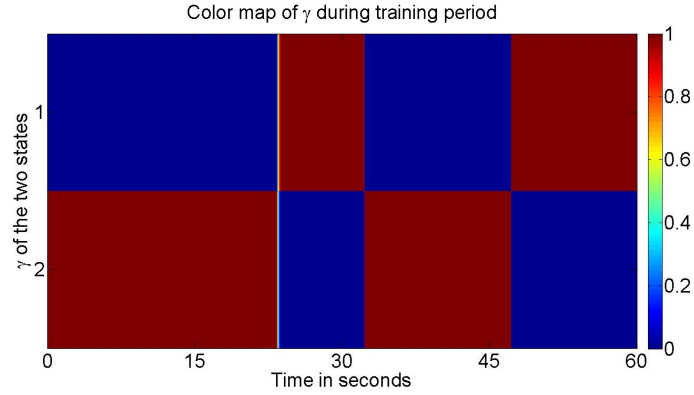
3.3 Thought Recognition

Our novel HMM implementation uses performance metrics generated by the Emotiv software suite as observations and maps them to discrete thoughts. We observe that each thought will corresponds to different signatures of these metrics. However, these signatures vary greatly across different trials and experimental conditions, so a heuristic approach to classify them is not effective. Instead we implement a training phase to train the HMM to detect the users though signatures and transition probabilities.

We are use a two state HMM which represents two distinct thoughts of the user, so



(a) Observation data over training period



(b) State estimation of the training data

Figure 3-2: (a) Shows the 3 metrics from the EEG signal (red for 'Meditation', green for 'Excitement' and blue for 'Engagement') during the training period of 60 seconds. (b) Shows the estimated state of the HMM during the training period by plotting the color map of γ . The lower state corresponds to a “disperse” thought by the user, and the upper to an “aggregate” thought by the user.

$m = 2$ in our case. The observation space consists of the EEG output $U(t)$. In this case, the signal $U(t)$ is derived from performance metrics provided by the manufacturer of the EEG headset, Emotiv. Emotiv provides six metrics: “Engagement”, “Meditation”, “Excitement”, “Frustration”, “Valence” and “Long-Term Excitement” out of which we use the first three metrics, which makes $l = 3$ and also $O_t \in [0, 1]$ for all metrics.

A typical training signal from our experiments is shown in Fig. 3-2(a) and the resulting state sequence after training is shown in Fig. 3-2(b). The user visited the two states

(thoughts) twice each during the training period. From Fig. 3.2(b) one can see that the Baum-Welch algorithm detects the switching sequence between the thoughts, since at each all times the state is found to be decisively in either one or the other state with high probability. That is, at all times one state is red (meaning the probability that the user is in that thought state is nearly one), while the other is blue (meaning probability that the user is in that thought state is almost zero).

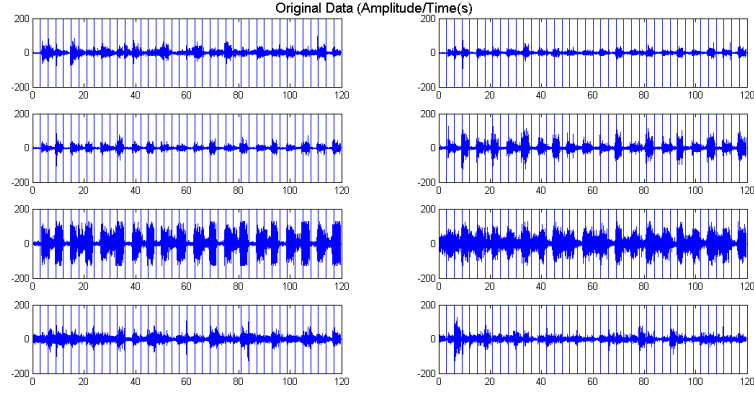
The output h is used to determine the the control parameters $(a(t), b(t))$ for aggregation and dispersion of the swarm.

3.4 Gesture Recognition

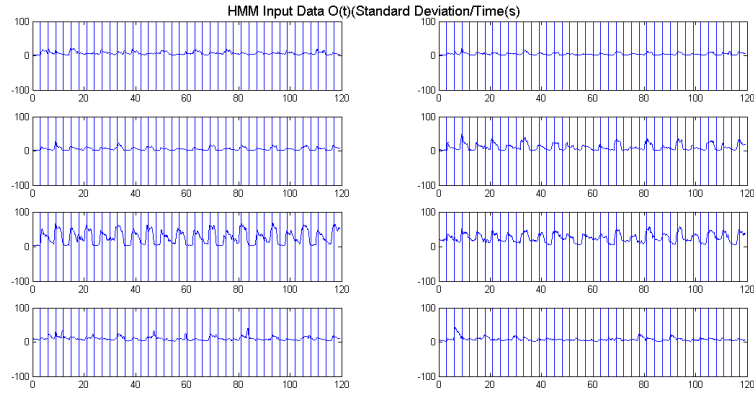
We have the 8-dimensional raw EMG signal as our input. Our HMM implementation uses standard deviation of these EMG signals of 0.5s window, with 0.05s overlap as input features. This makes $l = 8$, and also $O_t \in [0, 128]$ for all the signals since the raw signals are pre-processed and digitized to 8 bit signals. However, these signatures vary greatly across different trials and experimental conditions, so a heuristic approach to classify them is not effective. Instead we implement a training phase to train the HMM to detect the users gestures and transition probabilities.

Training data consisting of a TXl matrix, where T is the number of samples, which is recorded in a single trial. During the training period the user repeats the m gestures in a particular sequence. We enforced this gesture sequence using a timed audio sequence which the user listens during training. This training data is fed into the Baum-Welch algorithm which estimates the parameters θ for the system.

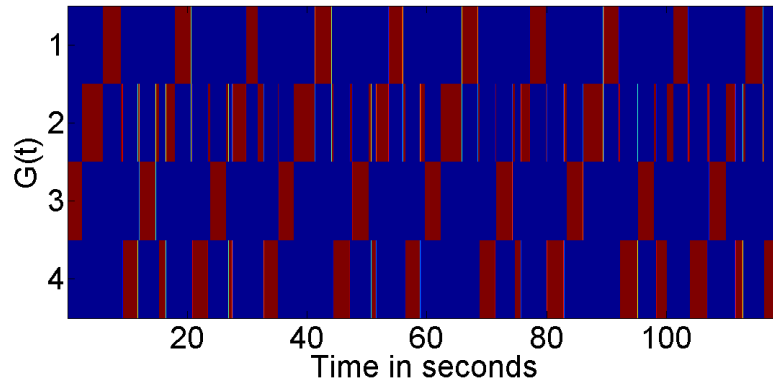
A typical raw training signal from our experiments is shown in Fig. 3.3(a). After performing windowing and standard deviation and the resulting state sequence after training is shown in Fig. 3.3(c). The user repeated the 4 states (gestures) 10 times each during the training period of 120 seconds. The gesture was performed for 3 seconds each, as shown



(a) Raw data from armband over training period



(b) Observation data as input

 γ during Training period

(c) State estimation of the training data

Figure 3.3: (a) Shows the raw data captured during the training period of 120 seconds from the armband. (b) Shows the standard deviation of raw data over a 0.5s window with 0.05s overlap, used as input to Baum Welch algorithm. (c) Shows the state of the HMM during the training period by plotting the color map of γ .

by the windows on the figure. From Fig. 3-3(c) one can see that the Baum-Welch algorithm detects the switching sequence between the gestures, since at each all times the state is found to be decisively in either one or the other state with high probability, as indicated by the color map. That is, at all times one state is red (meaning the probability that the user is in that gesture state is nearly one), while the other is blue (meaning probability that the user is in that gesture state is almost zero).

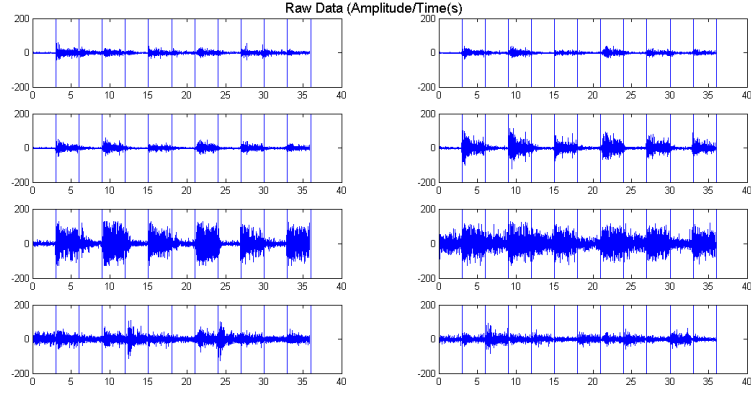
The gestures to specify the number of drones are used are shown in Figures 6-8(a), 6-8(b), 6-8(c), and 6-8(d). The gestures used during formation selection phase is shown in Fig. 6-8(e) and Fig. 6-8(f).

A typical implementation of the forward algorithm from our experiments is shown in Fig. 3-4 and the resulting state sequence after training is shown in Fig. 3-4(c). The user repeated the 4 states (gestures), 3 times each during the training period of 36 seconds, with 3 seconds window for each gesture. From Fig. 3-4(c) one can see that the Baum-Welch algorithm detects the switching sequence between the gestures, since at all times the state is found to be decisively in either one or the other state with high probability, as indicated by the color map.

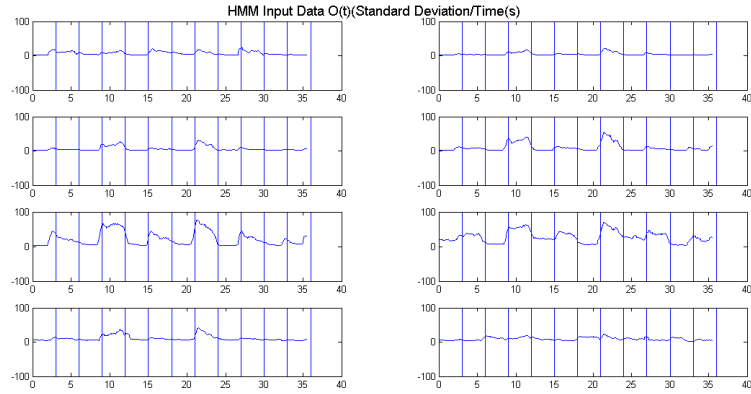
3.5 Formation Recognition

We need to recognize the user intended formation $\star p$ after choosing the number of drones M . To recognize formations we propose to draw the formation in the air and have a pipeline to recognize it. We use the magnetometer readings from the Inertial Measurement Unit (IMU) present in the MYO armband as raw signals.

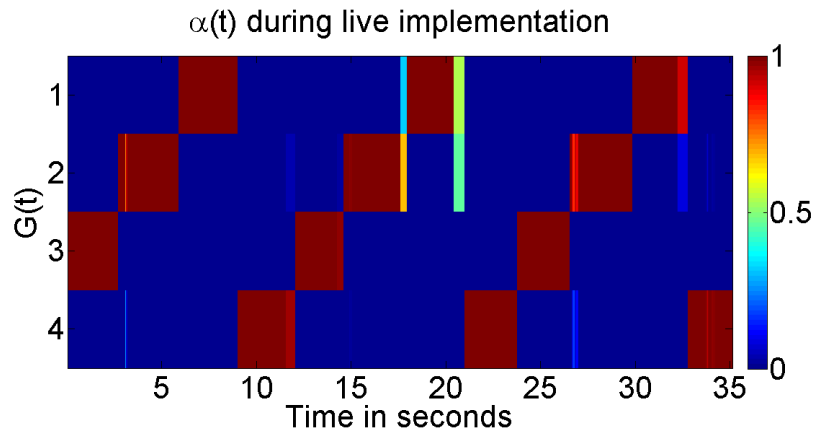
The problem involved here is twofold: Firstly, the gestures here are dynamic ,and there is no single duration in which it can be performed, hence the input data is varying in dimensions. A regular machine learning algorithm would need fixed dimensions for input feature vectors. Secondly with a series of magnetometer readings as input, it is hard to specify the



(a) Raw data from armband over live trial period



(b) Observation data as input during trial period



(c) State estimation of the training data

Figure 3.4: (a) Shows the raw data captured during the trial period of 36 seconds from the armband. (b) Shows the standard deviation of raw data over a 0.5s window with 0.05s overlap, used as input to Baum Welch algorithm. (c) Shows the state of the HMM during the trial period by plotting the color map of α .

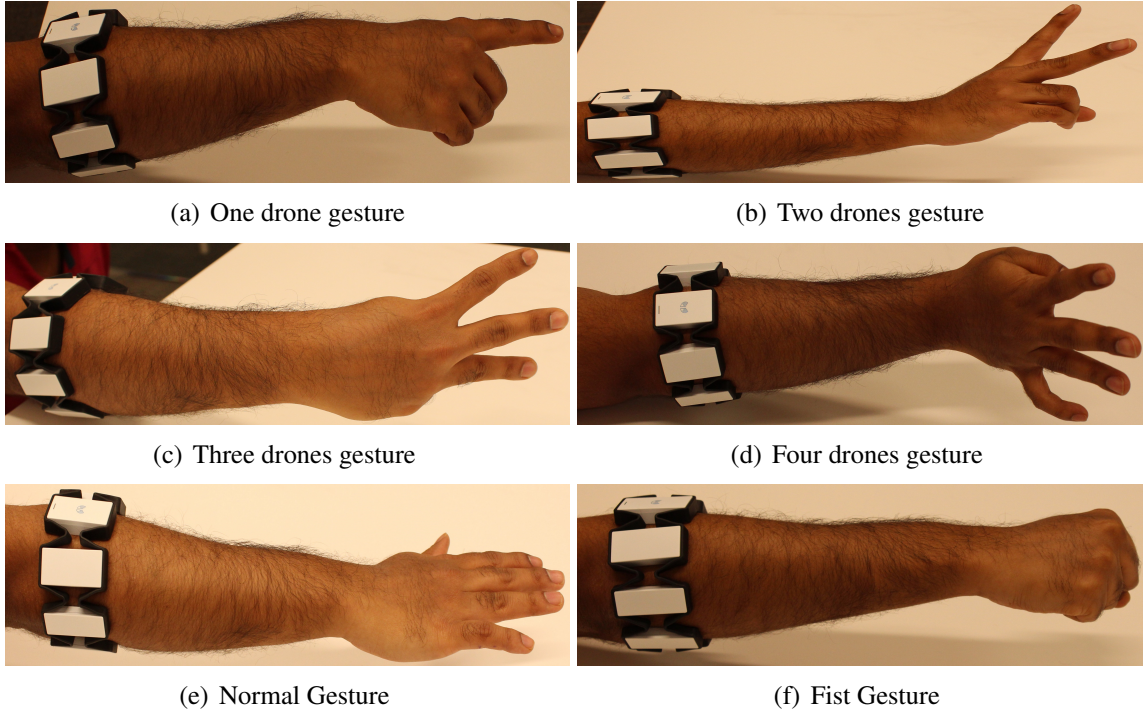


Figure 3-5: Gestures employed during drone and formation selection phase

drone positions, as there is not enough information in the IMU readings alone to determine both the formation and the position of each drones in this formation.

So we have come up with a novel hybrid approach to solve this problem by implementing a two stage machine learning pipeline to recognize both the formation and the position of drones in the formation. We first use a two class HMM with a 'normal' gesture and 'fist' gesture as shown in Fig. 6-8(e) and Fig. 6-8(f). We use both EMG and IMU data to determine formation of drones. The user draws the formation in the air and makes a fist to place a node (drone) in the formation, the magnetometer readings are recorded when the fist gesture is detected and stored. So the magnetometer readings are converted to euler angles, for example if the formation has three drones then there will be three magnetometer readings and correspondingly 9 euler angles.

These recordings are fed to a K-Nearest Neighbour (K-NN) trained classifier, which determines our desired formation $\star p$. A two class HMM is used to classify fist and normal

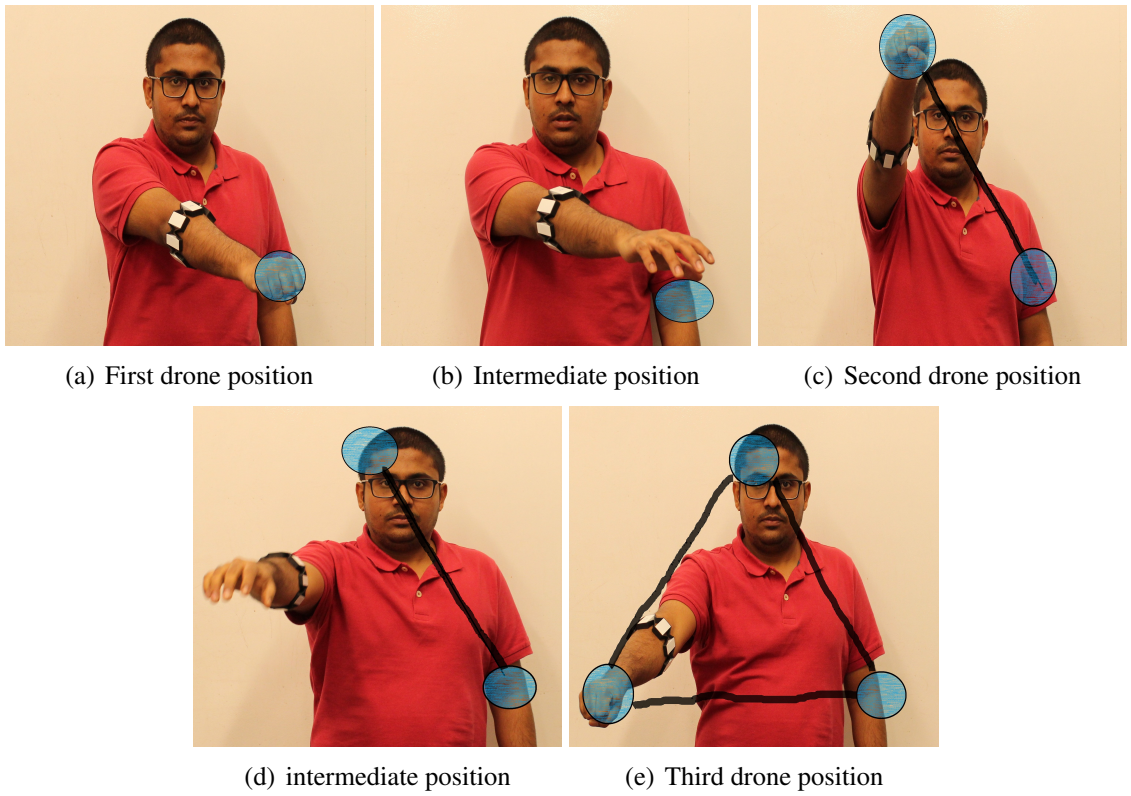


Figure 3-6: (a)-(e) Steps to specify a triangle formation by using fist gesture for drone position and normal gesture during movement

gesture. We train three separate K-NN classifiers for two, three, and four drones with magnetometer readings corresponding to various classifications. During the online phase we feed the magnetometer data to the K-NN classifier based on the number of agents selected in the previous step to determine the formation. Figures 6·9(a), 6·9(b), 6·9(c), 6·9(d), and 6·9(e) show snapshots of the user specifying a triangle formation. We performed rigorous accuracy tests and ground truth comparison to validate our framework, which will be discussed in the results section.

3.6 Tracking Eye Movements

In traditional EEG research, Eye Movement signals are considered as artifacts and are removed. In contrast, we use these signals as inputs for our system to command the direction of travel for the robots. There are various available methods in the literature for detecting and tracking eye movements, which vary considerably (Eggert, 2007). These methods can be broadly categorized into

1. Contact based tracking which offer high accuracy and sophistication,
2. Non-contact based optical tracking methods which measure relative positioning remotely with sensors such as cameras, and
3. Measuring surface electrical potentials from skin, also known as Electrooculogram (EOG), near the eyes.

Our EEG headset detects these EOG signals related to eye movement, hence we can detect eye movement with no additional hardware.

3.6.1 EOG : An Overview

The human eye can be modeled as an electrical dipole whose axis is roughly collinear to the axis of the human eye. The electrical dipole rotates with the rotation of the eye

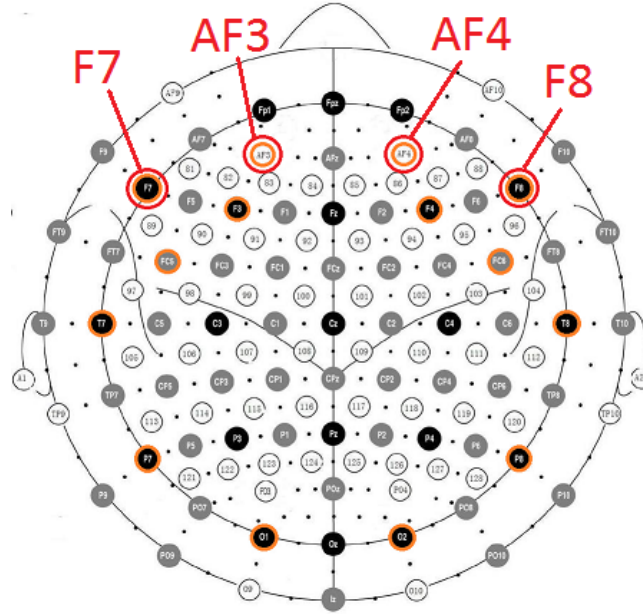


Figure 3.7: EEG sensor placement on the human scalp using the 10-20 system. The locations highlighted in orange depict the locations used by the Emotiv Epoc headset (<http://www.emotiv.com>). We specifically read the red circled locations to get EOG signals for eye movement.

causing small differences (in microvolts) between the electrical potential at the skin surface depending on eye position. The order of magnitude of these signals are much larger than signals due to brain activity evident from Fig. 3.9 and Fig.3.8, hence they can be measured and contrasted easily.

EOG typically uses exclusive electrodes around the eyes to measure movements. But our electrode positions are fixed so we employ the four closest electrodes to the eyes: ‘AF3’, ‘AF4’, ‘F7’ and ‘F8’, according to the 10-20 EEG sensor placement system, as shown in the diagram in Fig. 3.7. Previous methods to detect eye motion have relied on complex classification based algorithms. In contrast, our method uses a simple statistical calculation.

The spatio-temporal signals from these electrodes near the eyes can be described by $u_i(t) \in \mathbb{R}$ where $i \in \{AF3, AF4, F7, F8\}$ denoting the electrodes used. We first normalize the signal by subtracting its mean for each electrode to center the signals about zero. This

can be described by

$$u_i(t) - \frac{\sum_{i=0}^{\tau} u_i}{\tau} \quad (3.12)$$

where τ denotes the number of samples used for the baseline removal.

Horizontal Eye Movement Detection

Electrodes 'F7' and 'F8' are chosen for horizontal eye movement detection as they are the farthest apart in the horizontal plane while being closest to the eyes. Our algorithm for decoding horizontal directional movement depicted in Fig. 3.8 is described in Algorithm 1. In Fig. 3.8 the green ellipses indicate the signal for leftward eye movement and the blue ellipses indicate rightward eye movement. The red ellipse represents blinks which are filtered out.

Algorithm 1 Horizontal Eye Movement Detection

- 1: Remove Baseline with $\tau = 640$ samples.
- 2: Window the data with window size $w \in \mathbb{I}^+$

$$u_i^{\eta}(t) = \eta u_i(w) \quad (3.13)$$

where η represents the window number. We use $w = 128$ samples corresponding to 1 second of data with no overlap.

- 3: Apply 8th order 4 Hz low pass Butterworth filter to the windowed data to isolate the eye movement signals.
- 4: Subtract the resulting signals from both electrodes

$$u_{F7-F8}^{\eta}(t) = u_{F7}^{\eta}(t) - u_{F8}^{\eta}(t) \quad (3.14)$$

- 5: Detect peaks and troughs with threshold magnitude of $200\mu V$ and minimum separation of $w - 1$ samples in $u_{F7-F8}^{\eta}(t)$.
 - 6: Assign Peaks to eye movements to the left $e_l^{\eta} \in \mathbb{Z}^+$ and troughs to eye movements to the right $e_r^{\eta} \in \mathbb{Z}^+$.
-

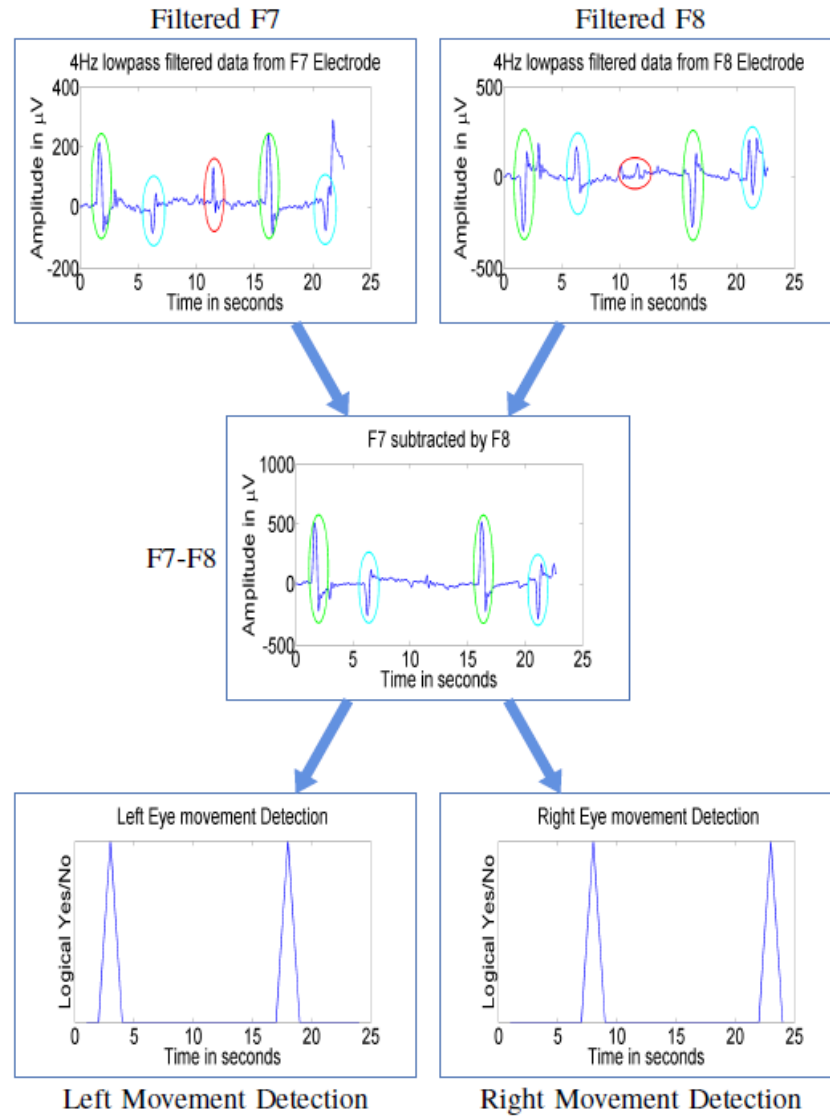


Figure 3-8: Horizontal Eye Movement Tracking. The main steps in Algorithm 1 are shown graphically from top to bottom. The bottom plots show the final extracted left and right eye movements.

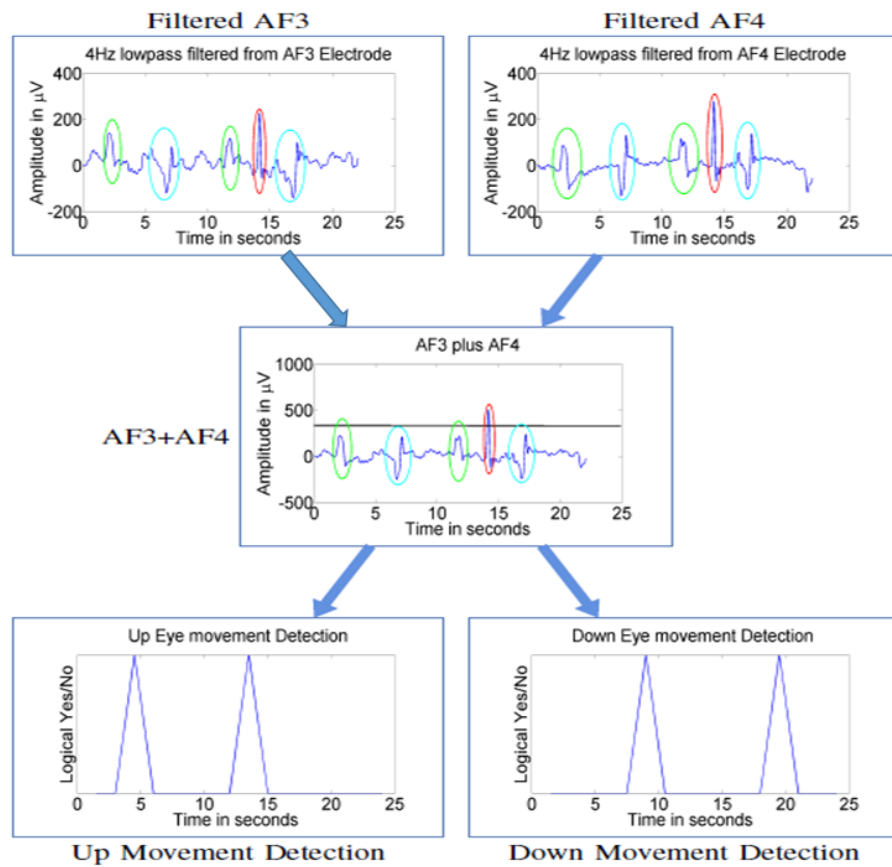


Figure 3-9: Vertical Eye Movement Tracking. The main steps in Algorithm 2 are shown graphically from top to bottom. The bottom plots show the final extracted up and down eye movements.

Vertical Eye movement Detection

Electrodes 'AF3' and 'AF4' are chosen for vertical eye movement detection. The method used is different from horizontal eye movement detection since we do not have any electrode below the eyes to detect the dipoles in the vertical plane of the head. Eye movement upwards results in a positive deflection in both electrodes whereas eye movement downwards has negative deflection for both electrodes. Our algorithm for decoding vertical eye movement is depicted in Fig 3.9 and described in Algorithm 2. In Fig 3.9, the green ellipses indicate the signal for vertical eye movement, and the blue ellipses indicate horizontal movement. The red ellipse represents blinks which are filtered out. It should be

Algorithm 2 Vertical Eye Movement Detection

- 1: Remove Baseline with $\tau = 640$ samples.
- 2: Window the data similar to Equation 3.13
- 3: Same as step 3 in Algorithm 1.
- 4: Add the signals from both the electrodes to get

$$u_{AF3+AF4}^{\eta}(t) = u_{AF3}^{\eta}(t) + u_{AF4}^{\eta}(t) \quad (3.15)$$

- 5: Find peaks and troughs with minimum separation of $w - 1$ and a signal level of $150\mu V - 250\mu V$ from $u_{AF3+AF4}^{\eta}(t)$.
 - 6: Assign Peaks to upward eye movements $e_u^{\eta} \in \mathbb{Z}^+$ and troughs to downward eye movements $e_d^{\eta} \in \mathbb{Z}^+$.
-

noted that both the algorithms above also filter out blinks, which look quite similar to the eye movements (see Figs. 3.9 and 3.8). In the case of horizontal eye movements, the F7 and F8 electrodes both record blinks with almost equal magnitude since they are located approximately at the same distance from the eyes. Hence the signal $u_{F7-F8}^{\eta}(t)$ is automatically devoid of blinks (Eqn 3.15), as evident from Fig 3.8. For the vertical eye movement detection, we introduce an upper threshold to filter out the blinks, which typically are much larger than eye movement signals, as can be seen in Fig. 3.9. Hence, our algorithm effectively discards blinks and measures only the user's intentional eye movements.

Finally, after the left-right and up-down signals have been extracted from the user's eye

movements, these signals are used to control the left-right and forward-backward motion of the robot swarm through the control parameter $v(t)$.

Chapter 4

Swarm Formation Control

Let us begin this chapter by first providing a definition for a Multi-Agent formation. Well intuitively it can be defined as a group of agents with a specific set of inter-agent distances.

This set of inter-agent distances ought to be under the user's control or expectation at any instance of time in most situations. This is the notion of formation control. We come across this concept in many instances of daily life like in traffic for example a line of vehicles are in a formation and you don't want the inter-vehicle distance too close to zero, which will cause collisions. This formation control can be achieved using many techniques, some of which are briefly described in the next section.

4.1 Swarm control techniques overview

The first thing which might come into the readers mind is using methods similar to controlling a group of puppets. Here a centralized system (puppet master) controls a group of agents (puppets) (Ichikawa et al., 1989). The basic problem for this system is the computational complexity involved as group of agents are increased.

Next we can think of a leader follower approach (Ji et al., 2006; Yu et al., 2010), where the followers mimic the leader and follow its movement, like in a parade. This might be effective in certain cases, but may not be appropriate when the individual agents need to move independently. This also brings up the question what happens when leaders fail? These centralized techniques are not very robust and tolerant to faults. They are often computationally expensive, as most of the computation burden is dumped on a single

agent. The next few techniques have a more decentralized approach and intuitively seem more effective and efficient.

Potential functions and fields are another method of controlling multiple agents (Gazi and Passino, 2004). In this methods artificial fields are created in the environment of the agents. These fields which is based on the relative position of the agents govern the motion of agents. Similar to natural fields like gravitational and electro-magnetic fields agents will tend to move in a path which minimizes their energy as defined by the field. This is the key idea behind it, and based on the definition of the field and interactions between agents formation can be created and controlled in an effective and decentralized manner. This is computationally simpler as each agent computes based on its interactions only. This is primarily a distance based approach, where the movement is governed by the inter-agent distances.

There are displacement based approaches as well, where the user specifies directly or indirectly the relative displacements between agents to control the formation of agents (Jadbabaie et al., 2003). In case of these approaches, the agents are required to sense relative positions (displacements) of the other interacting agents and independently compute their control strategy to match the formation requirement.

Graph theory can also be utilized to govern formation. The given space of the agents can be divided into partitions based on the number of agents and the agents can be directed to occupy a particular position like the center of partition etc (Schwager et al., 2011; Cortes et al., 2002). These methods are meant for more optimal formation in terms of coverage. One example is the Centroidal Voronoi partitioning, where a convex space is divided by Voronoi partitions and the agents occupy the center of these partitions.

The methods based on graph theory are computationally and theoretically more complex than the potential field and displacement based method. The scope of this work is limited to the usage of potential function for formation control in case of ground robots,

displacement based formation control strategies in case of aerial vehicles, in the interest of its inherent simplicity.

4.2 Potential Field based controller

The idea of a potential field is taken from nature. For instance a charged particle navigating a magnetic field, or a small ball rolling up/down a hill. The idea is that depending on the strength of the field, or the slope of the hill, the particle, or the ball can arrive to the source of the field or get repelled away from the source, the magnet, or the valley in this example.

In robotics, we can simulate the same effect, by creating an artificial potential field that will attract the robot to the goal and repel it from obstacles. By designing adequate potential field, we can make the robot exhibit simple behaviors.

In potential field approach, we simply create an attractive field going inside the goal and a repulsive field making it move away from obstacles or other agents. The potential field is defined across the entire free space, and in each time step, we calculate the potential field at the agent position, and then calculate the induced force by this field. The agent then should move according to this force.

Let a potential function from points i and j can be represented by $U(i, j) = f(x_i, x_j)$ where x_i, x_j represents the state position vectors at points i and j over a global reference frame.

The force generated by this potential field is given as a function of the gradient of the potential field.

$$F(i, j) = -\nabla U(i, j)$$

This force can be attractive or repulsive based on the user definition. This force can be defined in many ways depending on the application.

The application of potential field in formation control of multi-agents started around late 20th century. Several prominent research publications have been made in this field,

Few of which are : in Giulietti et al. (Giulietti et al., 2000) the authors describe formation control strategies for autonomous air vehicles, Olfati-Saber and Murray (Olfati-Saber and Murray, 2002) describe different approaches for formation control of multi-agent (multi-robot) teams. Similarly, Reif and Wang (Reif and Wang, 1999) consider a distributed control approach for groups of robots, called the social potential fields method. Gazi and Passino (2004) (Gazi and Passino, 2004) describe a class of attraction/repulsion functions for multi-agent convergence and prove its stability. We will primarily be using the work of the latter due to its simplicity and ease of application.

4.2.1 System Description

We consider a swarm of M individuals in an n -dimensional Euclidean space. The position of i_{th} individual is described by $x_i \in \mathbb{R}^n$. All the individuals move simultaneously and know the exact relative position of all the other individuals. i.e. they are synchronous with no time delays. The system dynamics is given by the equation :

$$\dot{x}_i = \sum_{j=1, j \neq i}^M f(x_i - x_j), i = 1, 2, \dots, M \quad (4.1)$$

where x^i is the position vector of an i^{th} agent. $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a function which represents both the attraction and repulsion forces between agents i and j . Let $f(\cdot)$ be defined by :

$$f(y) = -y[f_a(\|y\|) - f_r(\|y\|)] \quad (4.2)$$

where $f_a : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ represents the attraction component and $f_r : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ represents the repulsion component. $\|y\| = \sqrt{y^T y}$ is the euclidean norm.

4.2.2 A few key Notes

From the system definition we can note that :

1. There is a notion of equilibrium distance in which attraction and repulsion forces

balance out which is the unique distance δ . i.e. $f_a(\|\delta\|) = f_r(\|\delta\|)$. Also, $f_a(\|y\|) > f_r(\|y\|)$ for $\|y\| > \|\delta\|$ and $f_a(\|y\|) < f_r(\|y\|)$ for $\|y\| < \|\delta\|$

2. The attraction and repulsion function $g(\cdot)$ is an odd function. i.e. $f(\|y\|) = -f(\| -y\|)$. This leads to an aggregation behavior.
3. The actual forces can be determined by $-yf_a(\|y\|)$ for attraction and $yf_r(\|y\|)$ for repulsion.
4. From our previous definition of potential fields. There exists potentials $J_a : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ and $J_r : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ such that $\nabla_y J_a(\|y\|) = yf_a(\|y\|)$ and $\nabla_y J_r(\|y\|) = yf_r(\|y\|)$
5. Plugging in the previous definitions to equation 4.1, we get

$$\dot{x}_i = \sum_{j=1, j \neq i}^M [\nabla_{x^i} J_a(\|x^i - x^j\|) - \nabla_{x^i} J_r(\|x^i - x^j\|)] \quad (4.3)$$

6. The center of the swarm can be given by $\bar{x} = \frac{\sum_{i=1}^M x^i}{M}$ which remains stationary for all time t . The proofs and stability analysis is given in (Gazi and Passino, 2004) and briefly described in the Appendix.

Controlling Size

Now with a brief review of potential field approach we can develop a suitable controller to satisfy our problem. Recall that the system is described by the two dimensional state space equation for the i^{th} agent

$$\dot{x}_i = \sum_{j=1}^M f_{ab}(x_i, x_j) + v. \quad (4.4)$$

We let the interaction between robots i and j be given by

$$f_{ab}(x_i, x_j) = \frac{a(x_j - x_i)}{(\|x_j - x_i\| - 2r)^2} - \frac{b(x_j - x_i)}{(\|x_j - x_i\| - 2r)^3} \quad (4.5)$$

Where r is the radius of the robot. We can see that the left term provides the attracting field, and the right the repelling field. The r term introduces a safety region around the robots so collision can be avoided.

There is an equilibrium inter-robot distance for this system, in which attraction and repulsion forces balance. Let that equilibrium distance be denoted δ , so that $f_a(\|\delta\|) = f_r(\|\delta\|)$. This δ is governed by the attraction and repulsion gains a and b respectively. In our method we vary the gains to achieve different equilibrium formation. We map the two state output from the HMM to two distinct sets of gains in order to achieve the aggregation and dispersion of the swarm.

4.2.3 Controlling motion

Now to control the motion of the swarm we rely on the output from the eye movement detection which gives us four possible motion commands: Forward, Backward, Left, and Right. We use these command to assign values to the vector v , which drives every robot in the swarm in the same direction.

$$\dot{x}_i = \sum_{j=1}^M f_{ab}(x_i, x_j) + v, \quad (4.6)$$

So now our final controller is given by

$$\dot{x}_i = \frac{a(x_j - x_i)}{(\|x_j - x_i\| - 2r)^2} - \frac{b(x_j - x_i)}{(\|x_j - x_i\| - 2r)^3} + v, \quad (4.7)$$

Depending on the eye movement the vector v is assigned preset values which makes all the agents in the swarm move locally in the direction of v independent of the swarms aggregation or dispersion.

4.3 Displacement based controller

Displacement based controllers are easy to construct and implement. Let $x_i \in \mathbb{R}^n$ denote the position of the i^{th} agent. Using single integrator model the desired formation can be specified as described in (Oh et al., 2015). Let $u_i \in \mathbb{R}^n$ be the control input, then

$$\dot{x}_i = u_i \quad i = 1, \dots, M \quad (4.8)$$

We assume that the following relative displacements are available for agent i

$$x_{ji} = x_j - x_i, \quad (4.9)$$

Let $\star p \in \mathbb{R}^{nM}$ be the desired formation configuration then the objective for the agents are to satisfy the following constraints

$$x_i - x_j = \star p_i - \star p_j, \quad (4.10)$$

$\star p_i$ only specifies the desired displacements, and may not be the actual absolute desired positions for the agents. The desired formation can be defined as :

$$E_x := \{x : x_j - x_i = \star p_j - \star p_i, \quad i, j \in \{1, \dots, M\}\} \quad (4.11)$$

Based on this model we use the swarm controller :

$$u_i = k_p \sum_{j \in \{1 \dots M\}} (x_j - x_i - \star p_j + \star p_i) + v \quad (4.12)$$

where $K_p > 0$ is the proportional gain which is tuned manually to get the desired results and $v \in \mathbb{R}^n$ is the displacement vector used to steer the swarm during and after formation.

Chapter 5

Hardware involved and Signal Acquisition

In this chapter we will illustrate the hardware used to implement our framework, and our signal acquisition methods. We will first describe the hardware used for bio-signal acquisition in the case of EEG and EMG signals, and later describe how we acquire and utilize these signals for further processing. We will also specify the ground and aerial robots used for this thesis.

5.1 EEG Signal Acquisition

We use the Emotiv Epoc headset (www.emotiv.com/), which is an off-the-shelf product to acquire EEG signals and performance metrics. The headset is shown in Fig. 5.1(a), and its technical specification is enlisted in Fig. 5.1(b). There are 14 sensors at fixed locations and two additional sensors which serve as reference present. The sensor locations are according to the 10-20 system of EEG spatial location nomenclature.

5.1.1 Headset setup and signal acquisition

The sensors used in the headset consist of a gold plated metal surface, which is in contact with conductive felt, which in turn is in contact with the scalp. To ensure quality signals are obtained, the felt pad needs to be wetted with a conductive solution. We use standard saline solution in the form of commercially available contact lens solution, to serve as our conductive solution. Also, long and thick hair often impede signal conduction from the scalp, so the user is often advised to have short hair. Emotiv software suite has a signal



(a) An Emotiv Epoc headset used for our experiments.

Number of channels	14 (plus CMS/DRL references)
Channel names (Int. 10-20 locations)	AF3, AF4, F3, F4, F7, F8, FC5, FC6, P7, P8, T7, T8, O1, O2
Sampling method	Sequential sampling, Single ADC
Sampling rate	~128Hz (2048Hz internal)
Resolution	16 bits (14 bits effective) 1 LSB = $1.95\mu\text{V}$
Bandwidth	0.2 - 45Hz, digital notch filters at 40Hz and 60Hz
Dynamic range (input referred)	256mVpp
Coupling mode	AC coupled
Connectivity	Proprietary wireless, 2.4GHz band
Battery type	Li-poly
Battery life (typical)	12 hrs
Impedance measurement	Contact quality using patented system

(b) The Technial specifications of the headset.

Figure 5.1: The Emotiv epoc headset along with its technical specifications is shown.

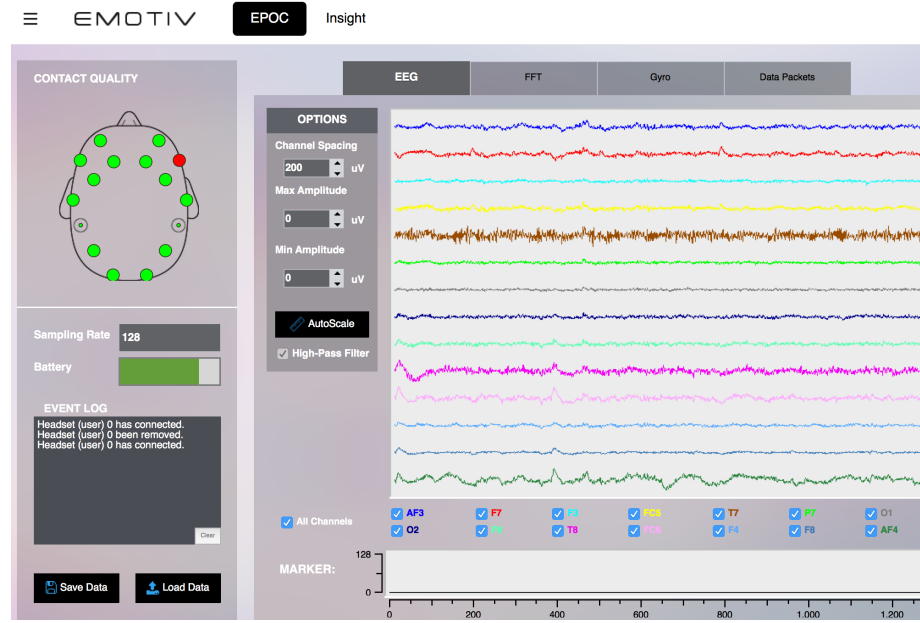


Figure 5.2: A screenshot of Emotiv application. Top-Left corner shown the contact quality information. Raw signals from 14 electrodes is shown in the right half of the picture.

quality tester which informs the user about the contact quality shown in Fig. 5.2. The top left corner of the screenshot (Fig.5.2) shows the signal quality information by a top-down view of the Emotiv headset sensor array. The contact quality of each sensor is represented by a color code from red which implies poor signals, to green which implies good signal quality.

We developed a custom MATLAB code environment to access both the raw signals and the performance metrics from the Emotiv Epoc headset. As described in earlier chapters, we use the raw signals for extracting eye movements and the performance metric signals for thought decoding.

5.2 Signal Acquisition from MYO armband

We use the MYO armband, which is an off-the-shelf product for accessing the EMG and IMU signals from the forearm. The armband is shown in Fig. 5.3. It uses 8 bi-polar

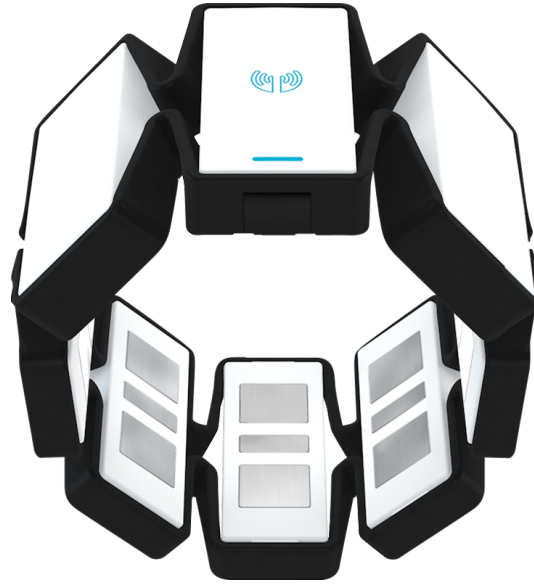


Figure 5.3: Myo armband

stainless steel electrodes to acquire surface EMG signals from the skeletal muscles on the forearm. It can be worn on either arm. It also has a sensitive 9 Degree of Freedom (DoF) Inertial Measurement Unit (IMU) to get motion characteristics from the arm. The IMU comprises of a 3-axis Accelerometer to measure acceleration, 3-axis Gyroscope to measure angular velocities, and 3-axis Magnetometer to measure angular orientations.

We have built a custom Nodejs framework to acquire the EMG and IMU signals from the MYO armband, which communicates wirelessly to a computer through Bluetooth 4.0. We then stream the IMU and EMG signals through TCP/IP data stream, and a MATLAB program is listening to this stream remotely or on the same computer. For EMG signals we window the signals with a 0.5 second window and 0.05 seconds overlap. We then take standard deviations of this window and feed it to our HMM as mentioned in the Chapter 4.

In the case of the magnetometer signals we receive quaternions as input signals. We convert these signals to euler angles and then feed it to our K-NN pipeline as mentioned in chapter 4. The conversion of quaternions to euler angles is a multi-step standard procedure in Robotics and the reader is advised to refer (Dam et al., 1998) for more details.

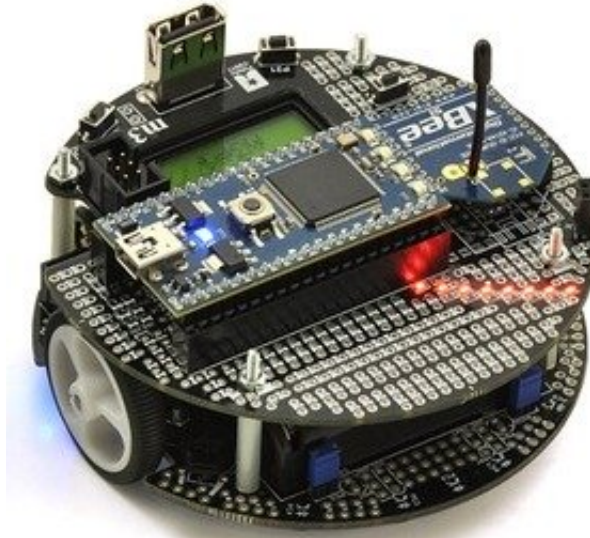


Figure 5-4: M3pi robot platform with Mbed micro-controller and xbee radios.

5.3 Ground Robotic Platform

We use the M3pi robot from Pololu (<https://www.pololu.com/product/2151>) as our ground robot test bed, it is shown in Fig. 5-4 . It is powered by a 32-bit Mbed micro-controller based on ARM architecture. They have an online compiler which uses C language and the user can set individual motor speeds through a program. We can send commands to it remotely through Xbee radios, which use the Zigbee protocol for communication. It is a standard two-wheeled differential drive robot with a third castor wheel. There are no motor encoders or IMU attached. Hence, precise control of this robot is a very difficult task, especially in open loop scenarios. We use the optitrack system to localize these robots and a Matlab program environment to send Zigbee commands remotely to these robots.



Figure 5-5: Assorted parrot mini drones used for our experiments.

5.4 Aerial Drone Platform

We decided to use the Parrot Mini drones as our Unmanned Aerial Vehicle (UAV) platform (<http://www.parrot.com/usa/products/minidrones/>) for performing experiments. These are quadrotors which run on a tiny linux powered PC, with onboard stabilization. These drones have ground facing ultrasonic sensors and a VGA cameras. They implement optical flow techniques to stabilize themselves. This feature is hugely advantageous since we don't have to program the lower level controls, and the drones only need higher level motion commands from the user. Unlike other quadcopters we do not have to worry about tuning gains or other parameters. We use a Node.js platform to send control commands remotely via Bluetooth 4.0. The models we chose are relatively cheap and have mounting blocks to mount lego components. We specifically chose these models to enable us to mount Infrared (IR) reflective markers, which is needed by the Optitrack system for localization. We eliminated the process of having to design and manufacture mounting pads for the markers, which is a time consuming and cumbersome process. A fleet of different Parrot mini drones used in our experiments is shown in Fig. 5-5.

Chapter 6

Results

This chapter gives the reader an insight into the Experimental and simulation results of our proposed system. Accuracy tests for the classification of gestures and formations have also been presented in this chapter to show the efficiency of our classification techniques.

6.1 Brain Swarm Interface

We performed both hardware experiments and hardware in the loop simulations to validate our proposal and framework. The details are mentioned as follows.

6.1.1 Simulations

To demonstrate our brain-swarm interface in simulation, we developed a simulation environment in Matlab. We chose a section of the Boston University campus, with a rectangular path around a campus building, as shown in Fig. 6-2. The path is divided into 4 edges and the swarm has to be driven starting from the left of edge 1 and end on the top of edge 4 following a clockwise motion. At edge 3 (purple path) due to the narrow passageway, the user has to make the swarm aggregate into a tighter swarm by switching thoughts, while in edges 1, 2 and 4 (Blue path) the user makes the swarm disperse.

For the training phase, the user switched between two thoughts at least twice over a period of 60 seconds, during which the EEG signals were recorded and fed into the Baum-Welch Algorithm (Fig. 3-2) to get the model parameters. The thoughts used for simulations and experiment were distinct and repeatable: the disperse state was invoked with a relaxed

neutral thought, while the aggregate state was invoked by a mentally challenging task (in this case calculating the Fibonacci series). Then EEG signals were streamed live and processed as discussed previously to generate the control inputs Θ .

The simulated swarm consisted of 128 point sized holonomic robots. The attraction gain a was fixed at 1 and the repulsion gain b was calculated according to $b(t) = h(t) * M/2.625$, where $h(t)$ is the estimated state sequence from Eqn. (3.11). In our two state HMM case according to the previous formula the user's thought corresponding to state 1 causes the robots to disperse and increase swarm size, and for state 2 causes the robots to converge and aggregate to a smaller size. The swarm reaches its equilibrium size for a particular thought state and stays at that size until the user switches thoughts.

The results of the simulation exercise is summarized in Figs. 6-1 and 6-2. Fig. 6-1 shows the time history of eye movement detection and the mental thought estimation during the motion of the swarm along the 4 legs of the path. It can be seen that the thought estimation remains mostly in the disperse state during legs 1, 2, and 4, and is mostly in the aggregate state in leg 3, as intended. Minor inaccuracies can be attributed not only to the stochastic nature of the HMM, but also to the quality of user's thoughts and noise in the EEG headset. From Fig. 6-2 which shows the path of the centroid of the swarm, we can see the Eye Movement detection is successful in steering the swarm, with a few misclassifications due to the nature of the noisy EEG signals and non-intentional eye movements.

6.1.2 Hardware Experiments

For the hardware experiments we used the m3pi platform with an Mbed controller for mobile swarming robots, and Zigbee radios for communication. The experiments were carried out in an environment with an Optitrack motion capture system to track the motion of the robots (see Fig. 6-3). The control parameters used were $a = 4$ and $b = 80$ for aggregation, and $a = 2$ and $b = 80$ for dispersion. These two wheeled differential drive robots receive individual motor speeds as control inputs from the computer. A proportional point-offset

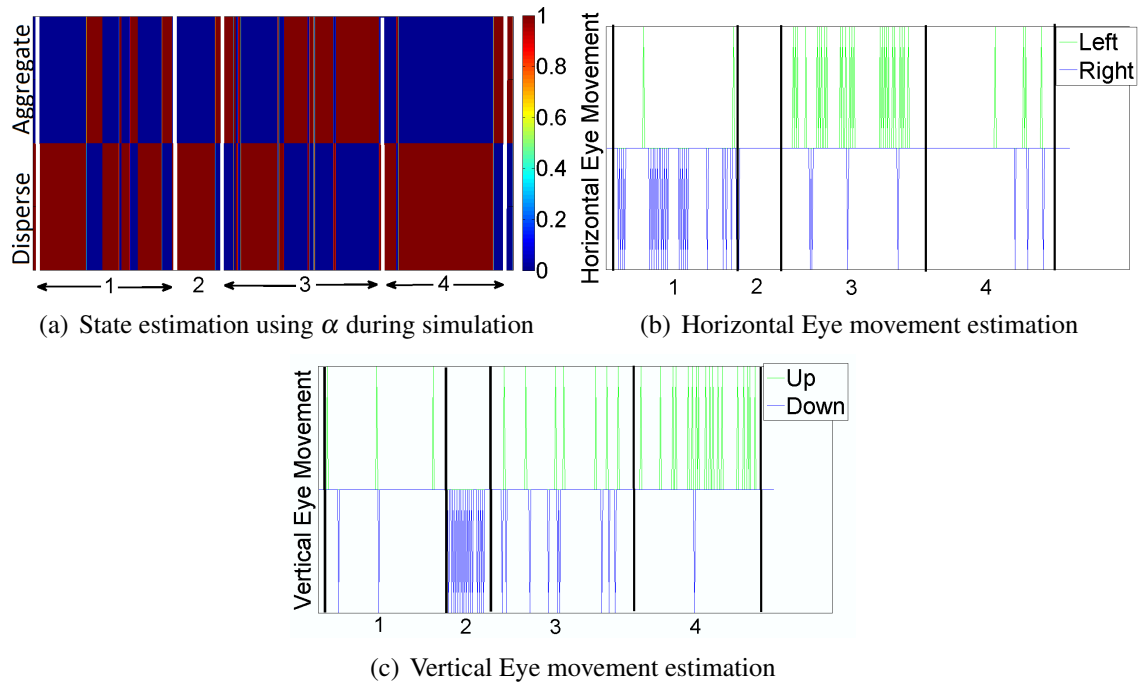


Figure 6.1: Simulation Results. a) Shows the color map of the variable α during the forward procedure. b) Shows the horizontal eye movement. c) Shows the vertical eye movement.

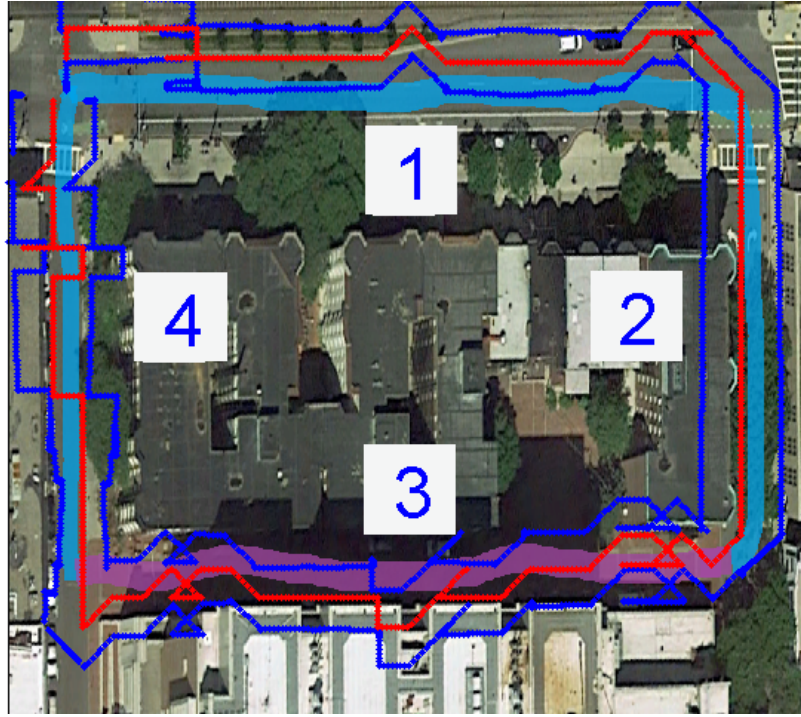


Figure 6-2: Path of the swarm (blue) along with its centroid (red) traveling in a simulated environment with 128 robots.

controller is used to generate the motor speeds from the potential function controller (the details of which can be found in (Pierson and Schwager, 2015)). The control commands for the robots were computed off board the robots, and set to the robots over Zigbee at an update rate of 30 Hz. Due to computational and hardware complexities the computations were divided among 3 computers (one for Optitrack data acquisition, one for controller implementation, and another other for EEG signal processing and video recording) as shown in the experimental setup in Fig. 6-4. The tcp/ip protocol was used for communication among them. The experimental area (Fig. 6-6) was chosen to be a rectangular area divided into 4 legs, similarly to the simulation. In leg 3 the user again must make the swarm aggregate, and in the other legs the swarm should disperse, while navigating in a clockwise manner starting from sector 1. The user obtained visual feedback of the position of the swarm by viewing a live feed of the experimental area from a GoPro camera on a mobile device.

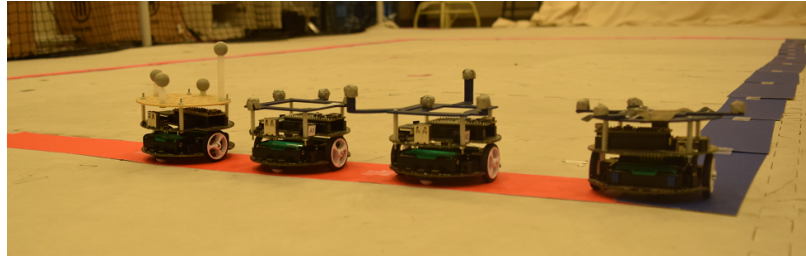


Figure 6.3: M3pi Robots in the Optitrack Arena.

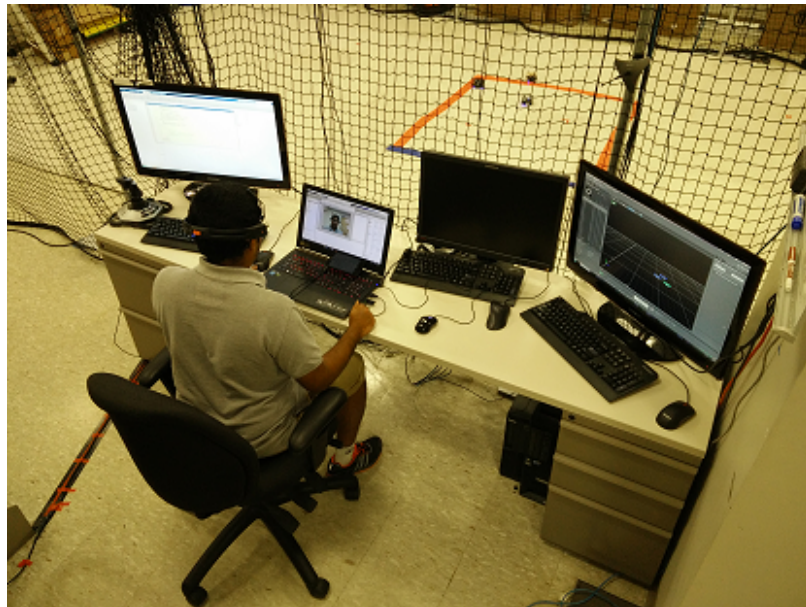


Figure 6.4: Experimental area and setup.

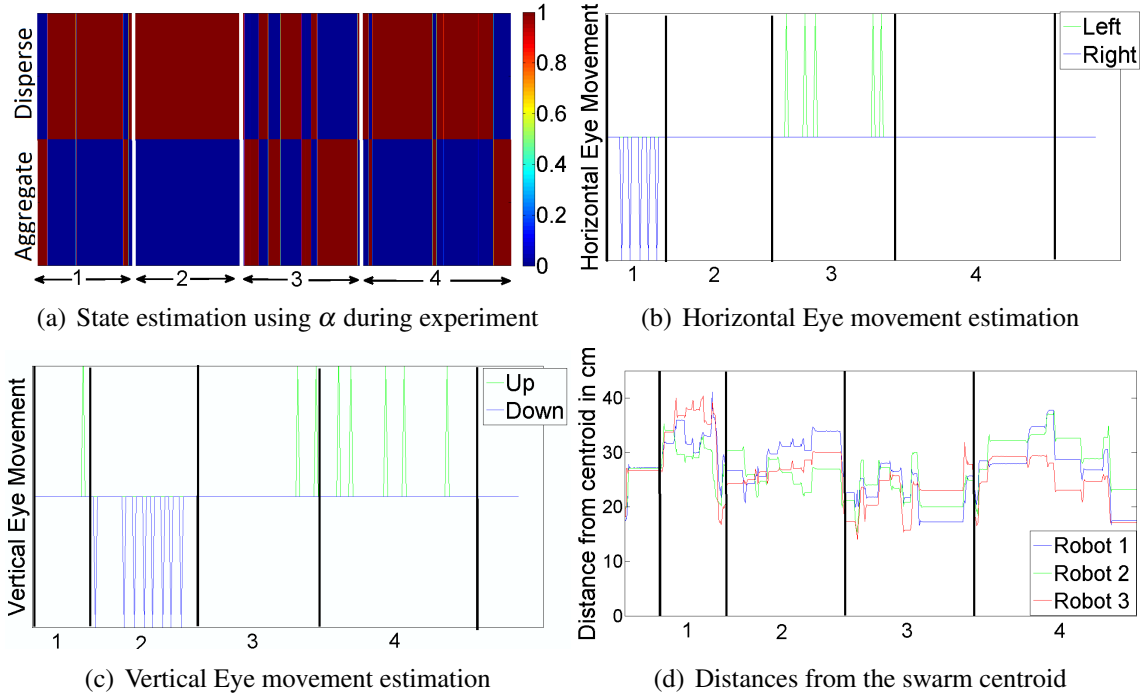


Figure 6.5: Experiment Results. a) Shows the color map of the variable α during the forward procedure. b) Shows the horizontal eye movement. c) Shows the vertical eye movement. d) Shows the distance from centroid of the three robots in centimeter.

The experimental results are summarized in Figs. 6.5 and 6.6. Fig. 6.5 shows the time history of eye movement detection and the thought state estimation throughout the motion of the swarm along the 4 legs. From Fig. 6.2 which shows the path of the robots and their centroid, we can see the eye movement detection is successful in steering the swarm. The color map for the thought state estimation again shows that the HMM is able to reliably determine the user's intention. The system remains in the disperse state with high confidence during legs 1, 2, and 4, and in the aggregate state with high confidence during leg 3.

The biggest challenge in this work is in the integration of this complex system with interacting hardware, communication, software, and human components. We used holonomic dynamics during simulation whereas The M3pis are nonholonomic robots with inefficient actuation and communication. In addition, it was quite a mental challenge for the user to concentrate on thoughts, eye movement, and system monitoring simultaneously. Despite

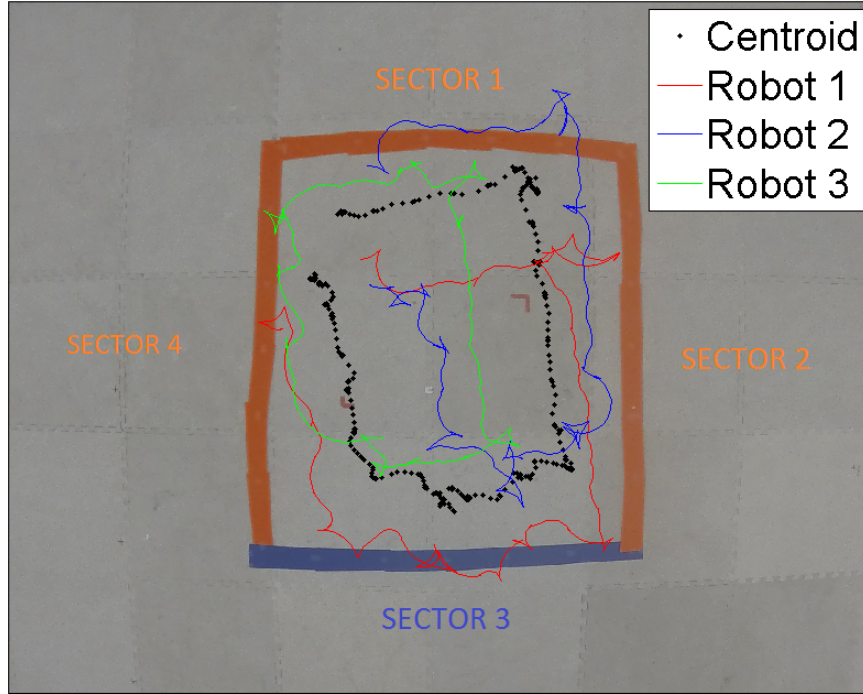


Figure 6.6: Position of the swarm of 3 robots along with their centroid during the experiment.

these challenges, we were able to successfully demonstrate the proposed method.

6.2 Arm Swarm Interface

In this section we describe our experiments and accuracy tests for the ASI. We identify the number of drones according to the pipeline mentioned in Chapter 4. Here we describe our formation recognition pipeline. We have created a formation suite consisting of different formations for different number of drones. We have two formations for two drones ($M = 2$), four different formations for $M = 3$ drones, and three formations for $M = 4$ drones, totaling 9 different formations, as listed in Table 1.

6.2.1 Accuracy Tests

We performed accuracy tests for the five different classifications of gestures and formation as described before. The results are summarized in Fig. 6.7. For detecting M we used a

Number of Drones	Formation shape	Formation number
Two Drones	Horizontal Line	1
	Vertical Line	2
Three Drones	Horizontal Line	3
	Vertical Line	4
	Planar Triangle	5
	3-Dimensional triangle	6
Four Drones	Horizontal Line	7
	Vertical Line	8
	Rectangle	9

Table 6.1: Formation suite for variable number of drones used

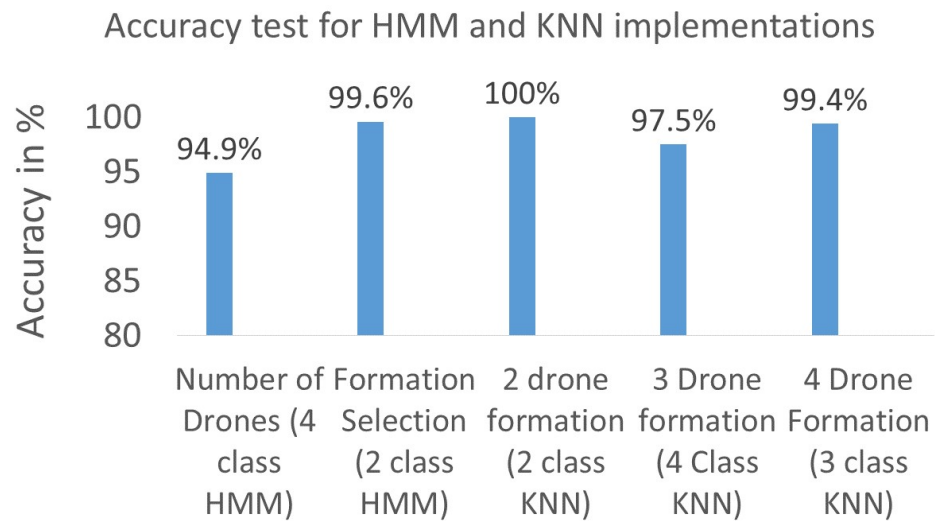


Figure 6-7: Accuracy test for five different HMM and K-NN implementations.

4 class HMM. We trained the model for 144 seconds in which each of the four gestures (Figs. 6-8(a), 6-8(b), 6-8(c), and 6-8(d)) was performed for 3 seconds and repeated 12 times. A timed audio sequence was used to enforce the procedure. There were 20 trials for live classification, each trial was performed for 36 seconds, the four gestures were performed for 3 seconds and repeated 3 times each. Again, a timed audio sequence enforced the user to perform specific sequence of gestures. Inaccuracies in classifications were calculated and reported. The result shown is the average classification accuracy over 20 trials.

The accuracy test for the 2 class HMM to detect drone position during formation phase was performed in a similar manner as the previous case. To detect the formation number using K-NN, 50 samples for each formation were used to train a K-NN classifier. Three different K-NN classifiers were used for the different number of drones. The samples consisted of the 3-axis magnetometer readings corresponding to positions of drones. So for example, for the four drone formation classifier, four magnetometer readings corresponding to the positions of the four drones constituted one sample. Accuracy tests were performed on 20 trials consisting of 8 samples each for the three K-NN classifiers. Accuracy tests for 160 samples overall for each K-NN classifier are reported in Fig. 6-7.

6.2.2 Experiments

Experiments were performed in the Boston University Robotics laboratory test bed. The bed consists of a flight space with IR cameras to track reflective markers on the quadrotors using the OptiTrack system. This system allows for real-time localization of the quadrotors during experiments. Four mini drones from parrot were used to execute the formation controller described earlier.

Our proposed experimental solution is divided into three parts:

- **Phase 1:** We first determine the number of drones M the user wants to control by using by training a four class Hidden Markov Model (HMM) to recognize the number

of fingers the user is showing. The gestures used are shown in figures 6-8(a), 6-8(b), 6-8(c), and 6-8(d).

- **Phase 2:** Next we select the desired formation $\star p$. We use both EMG and IMU data to determine formation of drones. The user draws the formation in the air and makes a fist to place a node (drone) in the formation, the magnetometer readings are recorded when the fist gesture is detected and stored. These recordings are fed to a K-Nearest Neighbour (K-NN) trained classifier, which determines our desired formation $\star p$. Figures 6-8(e) and 6-8(f) show the normal and fist gestures used during formation selection. A two class HMM is used to classify fist and normal gesture. We train three separate K-NN classifiers for two, three, and four drones with magnetometer readings corresponding to various classifications. During the online phase we feed the magnetometer data to the K-NN classifier based on the number of agents selected in the previous step to determine the formation. Figures 6-9(a), 6-9(b), 6-9(c), 6-9(d), and 6-9(e) show snapshots of the user specifying a triangle formation.
- **Phase 3:** Lastly, during flight the user controls the movement of the swarm v . Magnetometer data is recorded and the motion vector v is decoded from it. 3-axis angular rotations (Roll, Pitch and Yaw) measured are assigned to three axis movement (Left-Right, Forward-Backward, and Up-down) of the swarm.

EMG and IMU signals were recorded using the Myo Armband. A Node.js server was used to simultaneously stream the EMG and IMU signals at 200 Hz and 50 Hz respectively. A MATLAB program was used to capture these signals, perform gesture recognition, get position data from Optitrack, and generate control signals for the drones. These control signals were transmitted to the drones via Bluetooth 4.0 protocol using a Node.js pipeline. The eight sensor locations of the MYO armband is fixed. We conducted several preliminary experiments to determine the most effective, at the same time intuitive combination of gestures for our framework. Experimentally we were able to get the formation controller

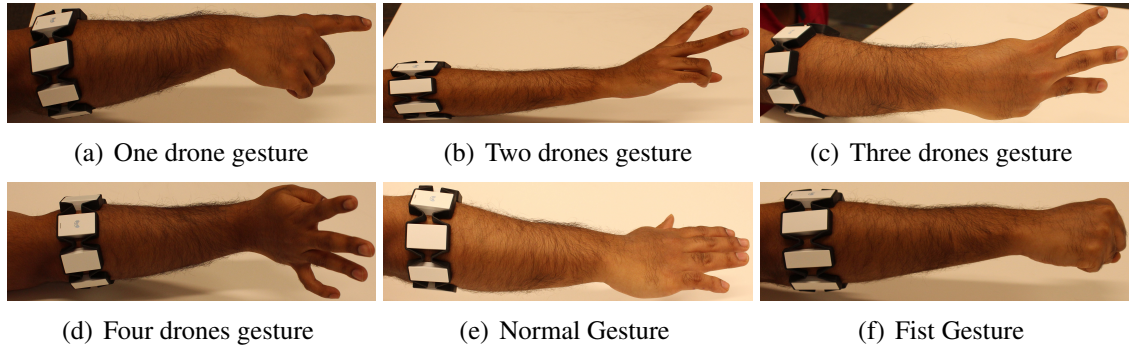


Figure 6.8: Gestures employed during drone and formation selection phase

to work and was evaluated both in MATLAB simulation and hardware experiment with the Parrot mini drones.

A screenshot of the triangle formation with parrot drones corresponding to the triangle formation gesture (Fig. 3-6) is shown in Fig. 6-9(f). We have also shown the sequence for vertical line formation (Fig. 6-10 and rectangle formation (Fig. 6-11 for increasing the reader's understanding of our system.

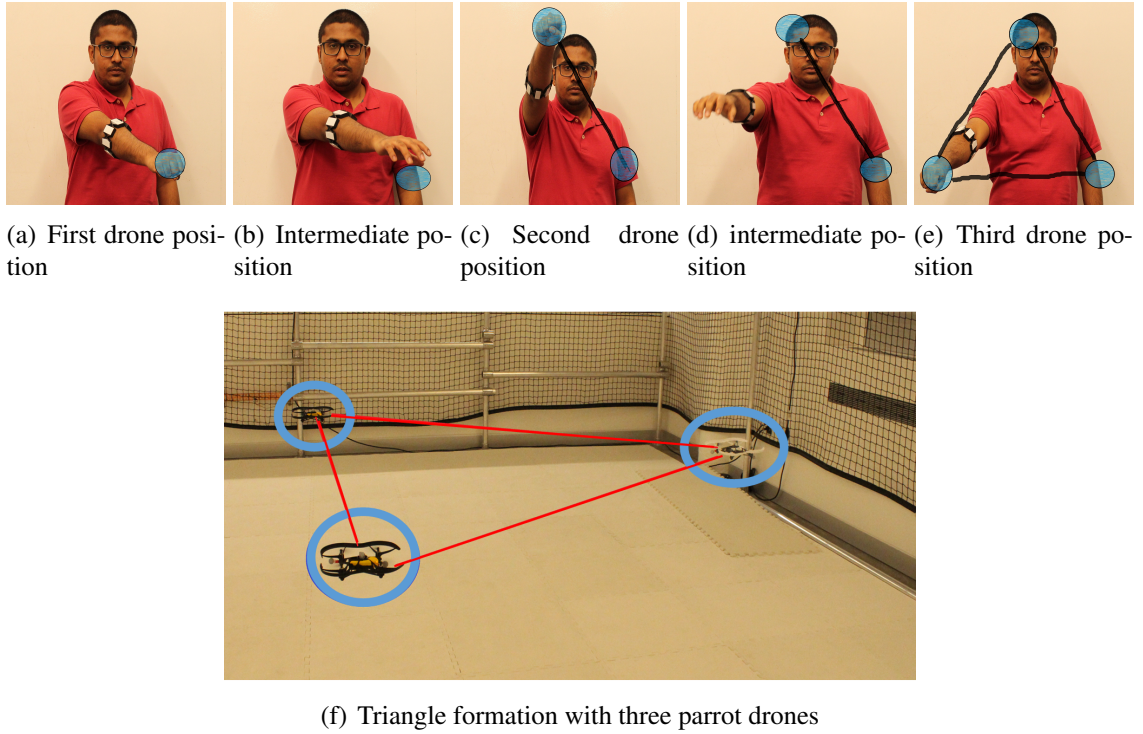


Figure 6-9: (a)-(e) Steps to specify a triangle formation by using fist gesture for drone position and normal gesture during movement, (f) Experimental validation of triangle formation with three mini Parrot drones.

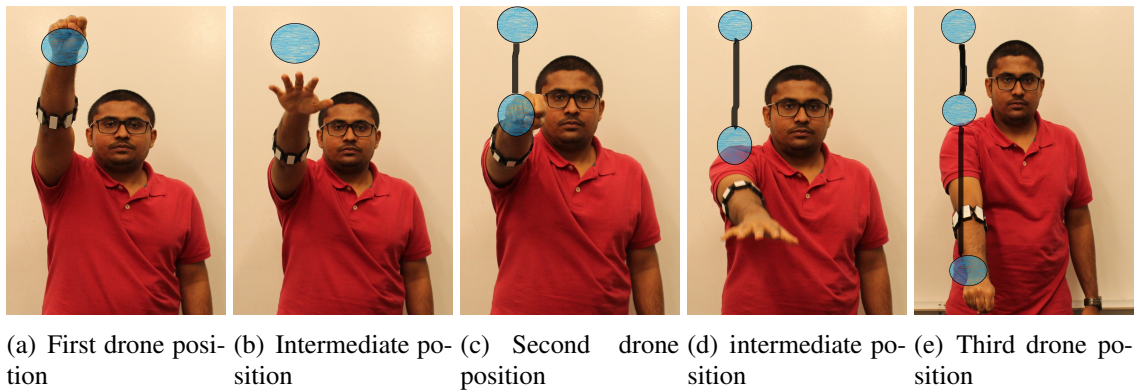
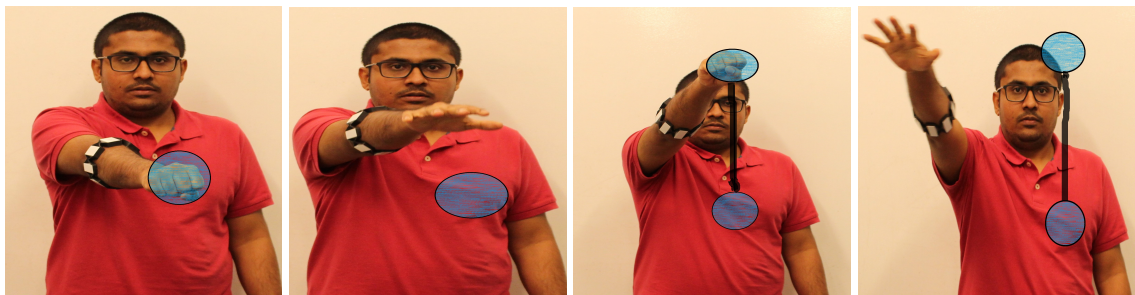
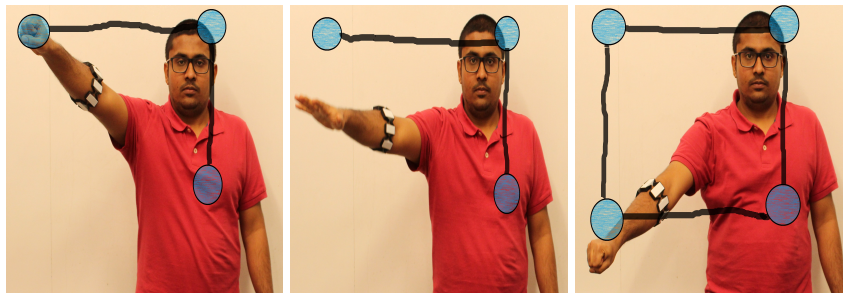


Figure 6-10: Specifying vertical line formation by using fist gesture for drone position and normal gesture during movement



(a) First drone position (b) Intermediate position (c) Second drone position (d) intermediate position



(e) Third drone position (f) intermediate position (g) Fourth drone position

Figure 6.11: Specifying rectangle formation by using fist gesture for drone position and normal gesture during movement

Chapter 7

Summary and Conclusions

7.1 Summary of the thesis

This thesis presents a novel interface to connect humans to swarms of robots, using their Bio-signals, which we call it as Body Swarm Interface (BoSI). We first present a theoretical and practical overview of the various concepts involved in constructing our interface. We define and solve our problem statement in two different ways, which we refer to as the Arm-Swarm Interface and the Brain-Swarm Interface. We describe the signal processing and machine learning pipeline in detail, which we use to decode thoughts, gestures, formations, and eye movement. We then move onto formation control, we illustrate and apply two different techniques based on relative displacements to control the swarms for both the ASI and the BSI application. The hardware used for signal acquisition, and the robotic platforms used for experiments are described in detail. We finally discuss our simulations and experimental results which validate our framework.

7.2 Conclusion

In this paper we propose and successfully demonstrate an online Body-Swarm Interface (BoSI) to control a swarm of ground vehicles and aerial vehicles in simulations and experiments using off-the-shelf hardware. We integrate a variety of engineering and scientific techniques in neuroscience, kinesiology, signal processing, machine learning, control theory, and swarm robotics to construct and implement our system. We successfully navigate a

robotic swarm in simulation and experiment on a given path in case of BSI and demonstrate formation control with aerial vehicles using ASI. The BSI idea and techniques developed in this paper are a proof of concept to demonstrate that a swarm of robots can be controlled by the thoughts and eye movements of a human user. The ASI techniques developed in this thesis illustrate the possibility of a robust and intuitive Human-Swarm Interaction. With impressive classification accuracies, we envision the rise of EMG based HCI methods for communication and control of robots and other devices in the near future. In theory, the Body-Swarm Interface developed in this thesis also gives disabled people the control multiple objects in their environment simultaneously. This work indicates the potential and promise of utilizing bio-signals to control swarms of robots. We hope it inspires the reader to pursue future work in this field.

7.3 Future Work

Although we have demonstrated promising results, the reader can use the platform we have created and our ideas and make improvements to it. We have a few ideas which can be used to improve our current pipeline: Using EMG signals in conjunction with Magnetometer readings to control the motion of the swarms, which would allow a combination of gestures and arm movement to control the swarm motion more intuitively. Including gestures for formation scaling, which would allow the user not only to select the shape of the formation but its scaling in terms of its size as well. Another practical aspect of formation control is the ability to switch between different formation topologies. This option can be implemented by having a ‘next formation’ gesture to switch between formations as listed in Table 1. For the BSI framework access to a better medical grade EEG headset and additional subjects might give some new insights into our framework and provide better results. The use of more sophisticated signal processing and machine learning pipeline could improve the classification accuracy and provide better results.

Appendix A

Lyapunov stability analysis of potential function

We describe the proof as given in (Gazi and Passino, 2004), which applies to our system. In this case intuitively stability can be defined when the system comes to a stand still and the desired formation is reached. The state vector $x \in \mathbb{R}^{nM}$ can be defined as $x = [x^{1T}, x^{2T}, \dots, x^{MT}]^T$.

We can imagine that there will be a invariant or stationary set of equilibrium points depending on the initial conditions which can be denoted by $\Omega_e = \{x : \dot{x} = 0\}$. Which implies that $\dot{x}_i = 0 \forall i = 1, \dots, M$.

Let us prove the stability by using a generalized Lyapunov function. The lyapunov function can be thought as an energy function. Since we are dealing with potential fields the energy of the system should be intuitive to define. Let the Lyapunov function be defined as $J : \mathbb{R}^{nM} \rightarrow \mathbb{R}$.

$$J(x) = \sum_{i=1}^{M-1} \sum_{j=i+1}^M [J_a(\|x_i - x_j\|) - J_r(\|x_i - x_j\|)]. \quad (\text{A.1})$$

Taking the gradient of $J(x)$ w.r.t x_i of an i^{th} individual we get

$$\nabla_{x_i} J(x) = \sum_{j=1, j \neq i}^M [\nabla_{x_i} J_a(\|x_i - x_j\|) - \nabla_{x_i} J_r(\|x_i - x_j\|)] \quad (\text{A.2})$$

To continue to prove stability we need to take the time derivative of the lyapunov function and prove that it is negative definite for all t for asymptotic stability. If it is negative semi definite then La Salle's Invariance theorem can be used to prove stability. The time derivative

of the Lyapunov function is given by:

$$\dot{J}(x) = [\nabla_x J(x)]^T \dot{x} \quad (\text{A.3})$$

$$= \sum_{i=1}^M [\nabla_{x_i} J(x)]^T \dot{x}_i \quad (\text{A.4})$$

$$= \sum_{i=1}^M [-\dot{x}_i]^T \dot{x}_i \text{ from (3)} \quad (\text{A.5})$$

$$= - \sum_{i=1}^M \|\dot{x}_i\|^2 \leq 0 \quad (\text{A.6})$$

This shows that $J(x)$ is negative semi definite. We can't conclude its stability unless its Negative definite. So we move on to La Salle's principle. Notice $J(x) = 0$ is when $\dot{x}_i = 0 \forall i = 1, \dots, M$ if the system is defined such that $\Omega_0 = \{x : J(x) \leq J(x(0))\}$ is compact then we can safely conclude from La Salle's Principle that the chosen system is stable and converges to the largest invariant subset :

$$\Omega_1 = \{x \in \Omega_0 : \dot{J}(x) = 0\} = \{x \in \Omega_0 : \dot{x} = 0\} \subset \Omega_e$$

Now suppose the set is not compact we can still say that for every member i in the group

$$[\nabla_{x_i} J(x)]^T \dot{x}_i = -\|\dot{x}_i\|^2 \leq 0$$

This means that every moves in the direction of decrease in $J(x)$. Therefore, $\Omega_x = \{x(t) : t \geq 0\} \subset \Omega_0$ is compact. So La Salle's theorem can be applied to prove stability which means as $t \rightarrow \infty$ $x(t)$ converges to the largest invariant subset defined by :

$$\Omega_2 = \{x \in \Omega_x : \dot{J}(x) = 0\} = \{x \in \Omega_x : \dot{x} = 0\} \subset \Omega_e$$

We note that Ω_1 and Ω_2 are invariant subsets of Ω_e , we can safely conclude the stability by saying $x(t) \rightarrow \Omega_e$ as $t \rightarrow \infty$.

References

- Akce, A., Johnson, M., and Bretl, T. (2010). Remote teleoperation of an unmanned aircraft with a brain-machine interface: Theory and preliminary results. In *2010 IEEE International Conference on Robotics and Information*, pages 5322 – 5327.
- Alonso-Mora, J., Breitenmoser, A., Rufli, M., Siegwart, R., and Beardsley, P. (2012). Image and animation display with multiple mobile robots. *The International Journal of Robotics Research*, 31(6):753–773.
- Alonso-Mora, J., Haegeli Lohaus, S., Leemann, P., Siegwart, R., and Beardsley, P. (2015). Gesture based human-multi-robot swarm interaction and its application to an interactive display. In *2015 IEEE International Conference on Robotics and Automation*, pages 5948–5953. IEEE.
- Arbib, M. A., Metta, G., and van der Smagt, P. (2008). Neurorobotics: from vision to action. In *Springer Handbook of Robotics*, pages 1453–1480. Springer.
- Asimov, I. (1950). *I, Robot*. Bantam Books.
- Aziz, F., Arof, H., Mokhtar, N., and Mubin, M. (2014). Hmm based automated wheelchair navigation using eeg traces in eeg. *Journal of Neural Engineering*, 11(5):056018.
- Bell, C. J., Shenoy, P., Chalodhorn, R., and Rao, R. P. N. (2008). Control of a humanoid robot by a noninvasive braincomputer interface in humans. *Journal of Neural Engineering*, 5(2).
- Benyus, J. M. (1997). *Biomimicry*. William Morrow New York.
- Bilmes, J. A. (1997). A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, ICSI UC Berkeley.
- Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.
- Cecotti, H. (2010). A self-paced and calibration-less ssvep-based brain–computer interface speller. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(2):127–133.

- Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2002). Coverage control for mobile sensing networks. In *2002 IEEE International Conference on Robotics and Automation.*, volume 2, pages 1327–1332. IEEE.
- Dam, E. B., Koch, M., and Lillholm, M. (1998). Quaternions, interpolation and animation. Technical report, Department of Computer Science, University of Copenhagen. URL: https://archive.org/details/Erik_B_Dam_Martin_Koch_and_Martin_Lillholm___Quaternions_Interpolation_and_Animation.
- Dayan, P. and Abbott, L. (2005). *Theoretical Neuroscience: Computational And Mathematical Modeling of Neural Systems*. Massachusetts Institute of Technology Press.
- DeCandido, K. (2015). *Big Hero 6*. HarperCollins Canada.
- Eggert, T. (2007). Eye movement recordings: Methods. *Neuronal Control of Eye Movements*, 40:1534.
- Faudzi, A. A. M., Ali, M. H. K., Azman, M. A., and Ismail, Z. H. (2012). Real-time hand gestures system for mobile robots control. *Procedia Engineering*, 41:798 – 804. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).
- Galán, F., Nuttin, M., Lew, E., Ferrez, P. W., Vanacker, G., Philips, J., and Millán, J. d. R. (2008). A brain-actuated wheelchair: asynchronous and non-invasive brain-computer interfaces for continuous control of robots. *Clinical Neurophysiology*, 119(9):2159–2169.
- Gazi, V. and Passino, K. M. (2004). A class of attractions/repulsion functions for stable swarm aggregations. *International Journal of Control*, 77(18):1567–1579.
- Giulietti, F., Pollini, L., and Innocenti, M. (2000). Autonomous formation flight. *IEEE Control Systems*, 20(6):34–44.
- Goldsmith, A. (2005). *Wireless Communications*. Cambridge University Press, New York, NY, USA.
- Gray, E. (2004). *Nineteenth-century Torpedoes and Their Inventors*. Naval Institute Press.
- Hochberg, L. R., Serruya, M. D., Friebs, G. M., Mukand, J. A., Saleh, M., Caplan, A. H., Branner, A., Chen, D., Penn, R. D., and Donoghue, J. P. (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442(7099):164–171.
- Howard, A., Mataric, M. J., and Sukhatme, G. S. (2002). Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, pages 299–308.

- Ichikawa, Y., Senoh, M., Suzuki, M., Kamimura, H., Tomizawa, F., Sugiyama, S., and Sasaki, M. (1989). Method and apparatus for master-slave manipulation supplemented by automatic control based on level of operator skill. US Patent 4,837,734.
- Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001.
- Ji, M., Muhammad, A., and Egerstedt, M. (2006). Leader-based multi-agent coordination: Controllability and optimal control. In *Proceedings of the 2006 American Control Conference*, pages 1358–1363.
- Karavas, G. K. and Artemiadis, P. (2015). On the effect of swarm collective behavior on human perception: Towards brain-swarm interfaces. In *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 172–177.
- Kazerooni, H. (1989). Human/robot interaction via the transfer of power and information signals. i. dynamics and control analysis. In *1989 IEEE International Conference on Robotics and Automation*, pages 1632–1640. IEEE.
- Khushaba, R. N., Kodagoda, S., Takruri, M., and Dissanayake, G. (2012). Toward improved control of prosthetic fingers using surface electromyogram (emg) signals. *Expert Systems with Applications*, 39(12):10731–10738.
- LaFleur, K., Cassady, K., Doud, A., Shades, K., Rogin, E., and He, B. (2013). Quadcopter control in three-dimensional space using a noninvasive motor imagery-based braincomputer interface. *Journal of Neural Engineering*, 10(046003).
- Lee, H. and Choi, S. (2003). PCA+HMM+SVM for EEG pattern classification. In *Proceedings of the Seventh International Symposium on Signal Processing and Its Applications, 2003.*, pages 541–544. IEEE.
- Lee, S. and Saridis, G. N. (1984). The control of a prosthetic arm by emg pattern recognition. *IEEE Transactions on Automatic Control*, 29(4):290–302.
- Leeb, R., Friedman, D., Müller-Putz, G. R., Scherer, R., Slater, M., and Pfurtscheller, G. (2007). Self-paced (asynchronous) bci control of a wheelchair in virtual environments: a case study with a tetraplegic. *Computational intelligence and neuroscience*, 2007:79642.
- Li, J., Liang, J., Zhao, Q., Li, J., Hong, K., and Zhang, L. (2013). Design of assistive wheelchair system directly steered by human thoughts. *International journal of neural systems*, 23(03):1350013.

- Liang, H., Bronzino, J., and Peterson, D. (2012). *Biosignal Processing: Principles and Practices*. Taylor & Francis.
- Lotte, F., Congedo, M., Lecuyer, A., Lamarche, F., and Arnaldi, B. A review of classification algorithms for EEG-based braincomputer interfaces. *Journal of Neural Engineering*, 4(2):R1.
- Marzouqi, M. and Jarvis, R. A. (2004). Covert robotics: Hiding in known environments. In *2004 IEEE Conference on Robotics, Automation and Mechatronics*, volume 2, pages 804–809. IEEE.
- McLurkin, J., Smith, J., Frankel, J., Sotkowitz, D., Blau, D., and Schmidt, B. (2006). Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. In *Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, pages 72–75. Association for the Advancement of Artificial Intelligence.
- Merletti, R. and Parker, P. (2004). *Electromyography: Physiology, Engineering, and Non-Invasive Applications*. Wiley.
- Michael, N., Zavlanos, M. M., Kumar, V., and Pappas, G. J. (2008). Distributed multi-robot task assignment and formation control. In *2008 IEEE International Conference on Robotics and Automation*, pages 128–133. IEEE.
- Monajjemi, V. M., Wawerla, J., Vaughan, R., and Mori, G. (2013). Hri in the sky: Creating and commanding teams of uavs with a vision-mediated gestural interface. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 617–623.
- Murata, S. and Kurokawa, H. (2007). Self-reconfigurable robots. *IEEE Robotics & Automation Magazine*, 14(1):71–78.
- Murphy, R. R. (2004). Trial by fire [rescue robots]. *IEEE Robotics & Automation Magazine*, 11(3):50–61.
- Nagi, J., Ducatelle, F., Di Caro, G. A., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., and Gambardella, L. M. (2011). Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 342–347. IEEE.
- Nishikawa, D., Yu, W., Yokoi, H., and Kakazu, Y. (1999). Emg prosthetic hand controller using real-time learning method. In *1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings*, volume 1, pages 153–158. IEEE.
- Obermaier, B., Guger, C., Neuper, C., and Pfurtscheller, G. (2001). Hidden markov models for online classification of single trial EEG data. *Pattern Recognition Letters*, 22(12):1299–1309.

- Oh, K.-K., Park, M.-C., and Ahn, H.-S. (2015). A survey of multi-agent formation control. *Automatica*, 53:424 – 440.
- Olfati-Saber, R. and Murray, R. M. (2002). Distributed cooperative control of multiple vehicle formations using structural potential functions. In *Proceedings of the 15th IFAC World Congress, 2002*. International Federation of Automatic Control.
- Palva, S. and Palva, J. M. (2007). New vistas for α -frequency band oscillations. *Trends in Neurosciences*, 30(4):150 – 158.
- Pierson, A. and Schwager, M. (2015). Bio-inspired non-cooperative multi-robot herding. In *Proceedings of the International Conference on Robotics and Automation (ICRA 15)*, pages 1843–1849.
- Podevijn, G., OGrady, R., Nashed, Y. S., and Dorigo, M. (2013). Gesturing at subswarms: Towards direct human control of robot swarms. In *Towards Autonomous Robotic Systems*, pages 390–403. Springer.
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286.
- Rao, R. P. N. (2013). *Brain-Computer Interfacing: An Introduction*. Cambridge University Press, New York, NY, USA.
- Reif, J. H. and Wang, H. (1999). Social potential fields: A distributed behavioral control for autonomous robots. *Robotics and Autonomous Systems*, 27(3):171–194.
- Rusch, E. (2012). *The Mighty Mars Rovers: The Incredible Adventures of Spirit and Opportunity*. Houghton Mifflin Books for Children.
- Saget, S., Legras, F., and Coppin, G. (2008). Cooperative interface of a swarm of uavs. *arXiv preprint arXiv:0811.0335*.
- Şahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. In *Swarm robotics*, pages 10–20. Springer.
- Santhanam, G., Ryu, S. I., Byron, M. Y., Afshar, A., and Shenoy, K. V. (2006). A high-performance brain–computer interface. *Nature*, 442(7099):195–198.
- Schwager, M., Julian, B., Angermann, M., and Rus, D. (2011). Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561.
- Shimada, N., Shirai, Y., Kuno, Y., and Miura, J. (1998). Hand gesture estimation and model refinement using monocular camera-ambiguity limitation by inequality constraints. In *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998.*, pages 268–273. IEEE.

- Stoica, A., Assad, C., Wolf, M., You, K. S., Pavone, M., Huntsberger, T., and Iwashita, Y. (2012). Using arm and hand gestures to command robots during stealth operations. In *Proceedings of 2012 Society of Photo-Optical Instrumentation Engineers (SPIE) conference series*, volume 8407. Society of Photo-Optical Instrumentation Engineers.
- Stoica, A., Salvioli, F., and Flowers, C. (2014). Remote control of quadrotor teams, using hand gestures. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, pages 296–297, New York, NY, USA. ACM.
- Tenore, F., Ramos, A., Fahmy, A., Acharya, S., Etienne-Cummings, R., and Thakor, N. V. (2007). Towards the control of individual fingers of a prosthetic hand using surface emg signals. In *29th Annual International Conference of the IEEE in Engineering in Medicine and Biology Society, 2007*, pages 6145–6148. IEEE.
- Tse, D. and Viswanath, P. (2005). *Fundamentals of Wireless Communication*. Cambridge University Press.
- Tsui, C. S. L., Jia, P., Gan, J. Q., Hu, H., and Yuan, K. (2007). Emg-based hands-free wheelchair control with eeg attention shift detection. In *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1266–1271. IEEE.
- Turnip, A., Suhendra, M. A., and Mada Sanjaya, W. (2015). Brain-controlled wheelchair based eeg-ssvep signals classified by nonlinear adaptive filter. In *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 905–908. IEEE.
- Vourvopoulos, A. and Liarokapis, F. (2012). Robot navigation using brain-computer interfaces. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1785–1792. IEEE.
- Waldherr, S., Romero, R., and Thrun, S. (2000). A gesture based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173.
- Winner, L. (1977). *Autonomous technology: Technics-out-of-control as a theme in political thought*. MIT Press.
- Wissel, T., Pfeiffer, T., Frysich, R., Knight, R. T., Chang, E. F., Hinrichs, H., Rieger, J. W., and Rose, G. (2013). Hidden markov model and support vector machine based decoding of finger movements using electrocorticography. *Journal of Neural Engineering*, 10(5):056020.
- Wolf, M. T., Assad, C., Vernacchia, M. T., Fromm, J., and Jethani, H. L. (2013). Gesture-based robot control with variable autonomy from the jpl biosleeve. In *2013 International Conference on Robotics and Automation*, Karlsruhe, Germany. IEEE.
- Wolpaw, J. and Wolpaw, E. (2012). *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, USA.

- Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791.
- Yu, W., Chen, G., and Cao, M. (2010). Distributed leader–follower flocking control for multi-agent dynamical systems with time-varying velocities. *Systems & Control Letters*, 59(9):543–552.
- Zecca, M., Micera, S., Carrozza, M., and Dario, P. (2002). Control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical Reviews in Biomedical Engineering*, 30:4–6.
- Zhou, D. and Schwager, M. (2016). Assistive collision avoidance for quadrotor swarm teleoperation. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*.

CURRICULUM VITAE

Aamodh Suresh

57 Brighton Avenue, #11, Allston, MA-02134, aamodh@gmail.com,

www.linkedin.com/in/aamodh,

EDUCATION:

Boston University (BU), College of Engineering Boston, MA

Masters of Science in Mechanical Engineering, May 2016, **GPA: 3.93/4.00**

National Institute of Technology (NITK) Surathkal Karnataka, India

Bachelor of Technology in Mechanical Engineering, May 2014

RESEARCH:

- **Master's Thesis, BU** **September 2014 – Present**
 - Developing an online **Brain Swarm Interface (BSI)** by using principles of Swarm Robotics and Brain Computer Interface (BCI) technology.
Signals from non-invasive EEG headset is used to decipher thoughts and eye movements of the user, which is then used as inputs to control a swarm of robots.
 - Developing a **Body Swarm Interface (BoSI)** by recognising hand gestures from the combination of EMG and IMU signals, and perform formation control on a swarm of aerial drones.
- **Bachelor's Thesis, NITK Surathkal,** **September 2013 - May 2014**
"Multi Robotic Deployment using Voronoi Partitioning with a Single Quadcopter". Used the principles of Voronoi Partitioning for dividing a given

bounded space and performed Multi-Robotic Optimal Deployment by using a single quadcopter in the loop.

- **Ege University, Izmir, Turkey** **July 2013 - August 2013**
Research work in manufacturing and application of Ionic polymer metal composite (IPMC) in Bio- mimetic actuators and developed image processing algorithm using MATLAB for control and feedback of micro robots in 2D frame.

INTERNSHIPS

- **COTMAC Electronics Pvt Ltd, Pune India** **May 2013 - June 2013**
Learnt PLC programming with SIEMENS PLCs, assembling control panels and used it for developing a Human Machine Interface (HMI) for an automated car parking system project.
- **Q-Protech Industries, Bangalore, India** **May 2012 - July 2012**
Received training in the practical aspects of various conventional and non conventional machining processes, heat treatment, surface treatment and industrial management.

PROJECTS

- **Unmanned Aerial Vehicle(UAV) Club, BU, September 2014 - Present**
Developing a controller for quadcopter swarms using EMG signals from the user. Built a Kinect controlled Quadcopter. Design and build of obstacle avoiding quadcopter with ultrasonic sensors and Arduino.
- **Networking in the Physical World (ME544), class project, BU**

Designed a "Mars-Rover" capable of autonomous navigation and obstacle avoidance, localisation about its environment and wireless communication over the internet using "SUNSpots" as the development platform.

- **SOLVE - Centre for system design, NITK, October 2011 - May 2012**

Modelling , simulations and manufacturing of 6 legged walking robot in order to emulate the "crab walk" based on the “Theo Jansen” mechanism.

- **Smart Room, Mechatronics class project, NITK, January 2013 - May 2013**

Built a working scaled model of a "Smart Room" equipped with smart power management, innovative alarm clock and door locking system.

AWARDS

- **Graduate Research Assistantship** **January 2015 – Present**

Presently working at the Multi-Robot Systems Laboratory under Professor Mac Schwager. I am developing novel interfaces for humans to interact with robotics swarms.

- **IAESTE Research Fellow** **July 2013 - August 2013**

Represented my country in the IAESTE Student exchange program. Pursued my research internship at Ege University at Izmir in Turkey.

ASSOCIATIONS

- **Boston University Unmanned Aerial Vehicles (BUUAV) Team**

September 2014 - Present

Position: Graduate student mentor. **Role 1:** Guide and teach physics, controls and programming aspects of UAVs, specifically quadcopters to the team.

Role2: Lead the sub group *Alternate Controls Team* and develop alternative techniques to control swarms of UAVs using Bio-signals and computer vision.