EVALUATION OF THE OPENBCI NEURAL INTERFACE FOR CONTROLLING A QUADROTOR SIMULATION

An Undergraduate Research Scholars Thesis

by

ALEJANDRO FRANCISCO AZOCAR

Submitted to Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Research Advisor: Aaron D. Ames

May 2015

Major: Aerospace Engineering

TABLE OF CONTENTS

Page

ABSTRACT

Evaluation of the OpenBCI Neural Interface for Controlling a Quadrotor Simulation. (May 2015)

Alejandro Francisco Azocar
Department of Aerospace Engineering
Texas A&M University

Research Advisor: Dr. Aaron D. Ames
Department of Mechanical Engineering

This thesis presents an initial analysis on the use of electroencephalography and electromyography to control the thrust settings of a quadrotor. The OpenBCI neural interface is used to sample muscle activity on a subject's face. Signal processing and event detection algorithms are implemented to identify eyewinks, and these wink events modify the thrust commands in a high fidelity, nonlinear quadrotor simulation. Currently only right and left wink events are detected; these can be mapped to two quadrotor commands such as fly up and down, roll right and left, pitch up and down, or yaw right and left. The ultimate goal of this project is to create a low-cost brain-machine interface system to fully control a real quadrotor using only bioelectrical signals such as electroencephalography and electromyography. A successful demonstration of the OpenBCI system may result in brain-machine interfaces that can be used in the development of low-cost prosthetic arms and legs.

DEDICATION

For my parents. This work is the culmination of my ever-changing interests. You have always encouraged my curiosity in math, science, and the arts, and my success comes from both of you.

Thank you for always supporting my crazy dreams.

Te quiero mucho.

ACKNOWLEDGMENTS

I have been supported by many great faculty members throughout my undergraduate career. I would like to thank Dr. John Hurtado for taking me into his lab when I was a freshman. He introduced me to the world of research and encouraged me to pursue a graduate degree. Dr. John Valasek has been a great mentor to me, pushing me to exceed my own expectations in research. Finally, I would like to thank Dr. Aaron Ames. He took me into his lab, gave me the freedom to pursue my own interests, and gave me incredible advice throughout the graduate school application process. This thesis would not be possible without his support.

I am also thankful for Dr. Casey Ricketts and Dr. Kristi Shryock. I would not be at Texas A&M without their help.

NOMENCLATURE

ALS	Amyotrophic Lateral Sclerosis
BMI	Brain-machine Interface
DCM	Direction-Cosine Matrix
DOF	Degrees of Freedom
ECoG	Electrocorticogram/Electrocorticography
EEG	Electroencephalogram/Electroencephalography
EKG	Electrocardiogram
EMG	Electromyography
fMRI	Functional Magnetic Resonance Imaging
ICMS	Intracortical Microstimulation
iEEG	Intracranial Electroencephalography
LFP	Local Field Potential
MEA	Microelectrode Array
NED	North, East, Down
RPM	Revolutions per Minute
SNR	Signal-to-Noise Ratio
VEP	Visually Evoked Potential

CHAPTER I

INTRODUCTION

Neurological injuries and limb loss have led millions of people to suffer from mobility related disabilities. The Christopher & Dana Reeve Foundation reports that nearly 1 in 50 Americans, approximately 6 million total, are suffering from paralysis; the Amputee Coalition estimates that nearly 2 million are suffering from limb loss [1, 2]. Current treatments for paralysis are generally ineffective, and typical prostheses are not capable of giving a patient true mobile independence. Brain-machine interface (BMI) technology, coupled with neural prostheses and exoskeletons, may eventually restore mobility and independence in amputees and patients suffering from impaired motor function due to spinal cord injury, strokes, cerebral palsy, and amyotrophic lateral sclerosis (ALS) [3]. In addition to the rehabilitation potential of BMIs, this technology may also be used to augment the lives of healthy subjects [4].

In a typical paralysis or limb loss case, the motor cortex in the brain is still healthy and able command movement; however, movement is not accomplished due to a disconnect between the nervous system and the body's effectors. BMIs are capable of bypassing this disconnect by recording the user's brain activity, predicting the user's intent from the brain signals, and sending commands to computers, robotic prostheses, and exoskeletons [4]. This process is usually a closed system, taking advantage of visual feedback as shown in Figure I.1. In the past, BMIs have been successfully used to type with a virtual keyboard and control wheelchairs; however, there are many challenges to solve before the ultimate goal of accurately controlling robotic arms, legs, and exoskeletons can be accomplished.

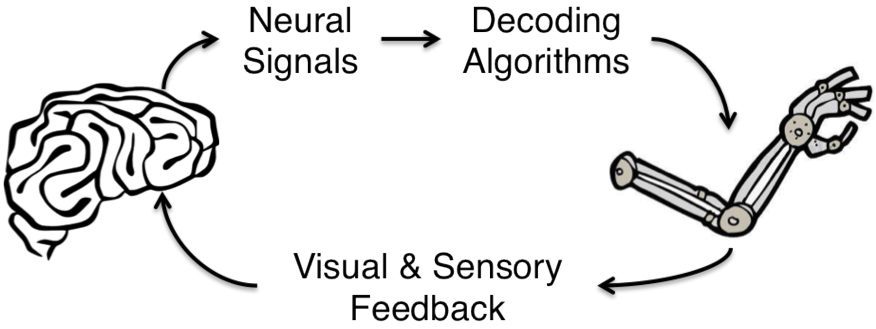


Fig. I.1. Component overview of a simple BMI.

Elements of a Brain-machine Interface

There are four major tasks that a BMI should be able to accomplish: BMIs must interface with the brain for real-time recording, extract and interpret noisy brain signals, command the signals to various kinds of actuators, and supply some form of feedback to the user. Each of these tasks is explored in more detail in the following sections.

Neuronal Recording

Obtaining neural signals requires a trade-off between resolution and level of invasiveness. The fundamental unit of neuronal activity in an individual neuron is the action potential discharge; the sum of the individual discharges creates an electrical field called a local field potential (LFP) [5]. To record single-unit activity, an electrode must be implanted in the brain. In order to accurately characterize the motor intentions of the brain, and to offer redundancy in control, multi-neuron recordings are necessary [6]. Microelectrode arrays (MEAs) offer this increased accuracy at the cost of invasive implants. On the opposite end of the spectrum is the electroencephalogram (EEG), which is non-invasively attached to the scalp. Electroencephalography (EEG) integrates the activity of neurons over a much larger area than MEAs do; as a result, EEG signals exhibit much lower resolution. Electrocorticography (ECoG), or intracranial EEG (iEEG), offers a compromise between MEAs and EEG. Since electrocorticograms (ECoGs) are placed directly on the surface of the brain, they measure signals at a higher resolution than EEG does; at the same time, they are less invasive than MEAs because they do not penetrate the blood-brain barrier [7]. Figure I.2 shows examples and the progression between EEG, ECoG, and MEAs.

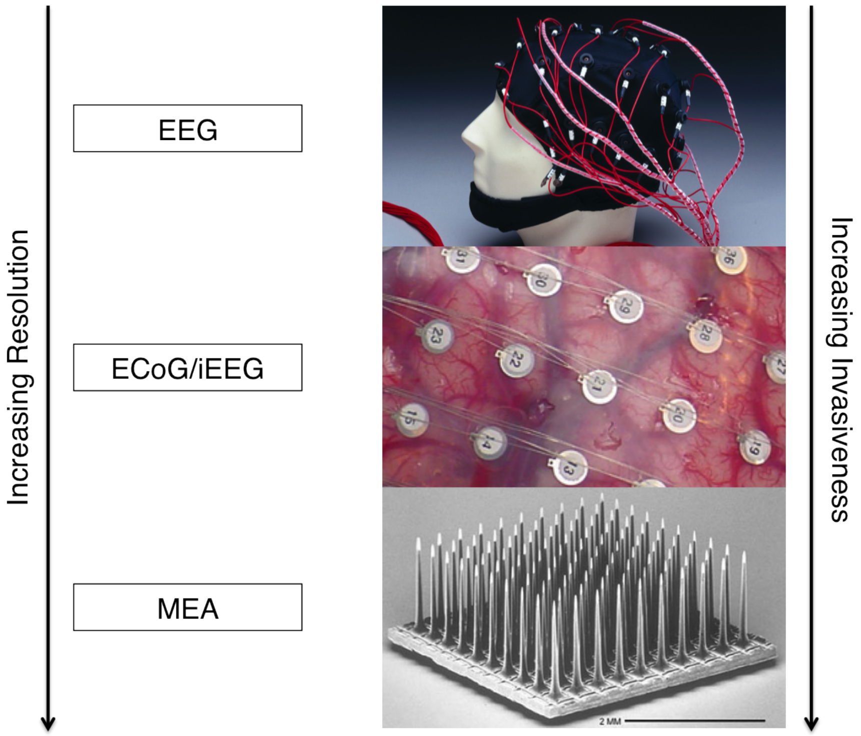


Fig. I.2. Resolution and invasiveness comparison of neuronal signal recording methods.

Other advantages and disadvantages of invasive and non-invasive BMIs are further discussed in the following sections.

Invasive BMIs

Although invasive BMIs are out of the scope of this work, they are worth discussing due to the opportunities and challenges presented by their implementation in BMIs. Due to their high resolution, MEAs are the only current method that can decode the subject's intended limb movements with high accuracy and reliability [4]. Both invasive and non-invasive methods have traditionally relied on visual feedback to close the loop; however, MEAs are capable of bi-directional data flow. This is currently being explored through intracortical microstimulation (ICMS)- stimulation of neurons via small electrical currents [4]. ICMS can be used reduce the time needed to learn new tasks with a neuroprosthesis, increase the accuracy of such tasks, artificially convey a sense of touch, and provide the brain with knowledge of the prosthesis' relative location to the rest of the body (proprioception). ICMS will allow neuroprostheses to feel like natural appendages instead of artificial actuators.

The biggest challenge faced by invasive BMIs is achieving stable, long-term recordings (over many years) of thousands of neurons from multiple areas in the brain [3]. Since a foreign object is inserted into the brain, the body reacts by creating a layer of scar tissue around the MEAs; over time this results in deterioration of the signal-to-noise ratio (SNR) of the electrodes [4]. Another challenge faced by invasive techniques is reliable, high-speed, wireless data transmission between the electrodes and a computer [3]. Both of these challenges must be addressed before invasive methods become robust enough to be used outside of research settings.

Non-invasive BMIs

The research presented in this thesis uses a combination of EEG and facial electromyography (EMG); therefore, we will focus on the many aspects of non-invasive BMIs. The clear advantage of non-invasive BMIs is that patients avoid the risks of brain surgery and the biocompatibility challenges that result from implanted MEAs. Unlike MEAs that measure neurons in specific locations, EEGs interpret a subject's intentions through the overall electrical activity of an enormous amount of neurons. As a result, EEG is typically used to detect brain activity related to visual stimuli, gaze angle, voluntary intentions, and cognitive states [3]. A particularly effective example of visual stimuli for brain activity modulation is the use of visually evoked potentials (VEP), or changes in the brain signals that occur when subjects looks at particular items on a screen [3].

A significant challenge for EEG-based BMIs is the amount of time required to train subjects to control their own brain activity [4]. Visual feedback from the actuator is essential for effective training; this provides an avenue for the implementation of virtual reality systems in BMIs in order to reduce training time [3]. A critical requirement for BMIs is the ability to control actuators while simultaneously performing unrelated tasks; with the proper training, some subjects have successfully controlled BMIs while speaking [4]. Since EEGs are placed on the scalp, they are subject to interference from muscular and ocular movements, as well as other electromagnetic effects [3]. Finally, most EEG electrodes require a conductive gel in order to function; this gel takes a significant amount of time to apply, limits the amount of sensors capable of being placed on the scalp, and eventually dries out [8]. This is a small challenge, but may limit recording time in longer tests. Dry-contact electrodes are under development, but are generally even lower resolution than typical electrodes.

Other non-invasive recording techniques have also been explored, including the detection of magnetic fields produced by ionic current flows and detection of changes in blood flow via functional magnetic resonance imaging (fMRI) [5]. A major drawback to using these systems, particularly fMRI, is the lack of mobility resulting from large, bulky hardware. EMG, recording the electrical signals produced by muscle contractions, has also been shown to control bionic legs and arms [9, 10].

Signal Extraction and Control Algorithms

The nature of signals corresponding to motor activity is not yet fully understood; however, the successful design of BMIs does not rely on exact knowledge of the motor activity in the brain [3]. The standard operating concept in the field is to repeatedly perform a specific task, construct a neural model for that task, and identify the neuronal characteristics associated with the task [5]. The recurring patterns resulting from motor tasks have been successfully analyzed using linear regressions and the Unscented Kalman Filter, among others [6]. The quantity and quality of data available depends on the recording method, bandwidth, frequency, and SNR. Specific signal processing techniques used in this project are discussed in more detail in the subsequent chapters.

Although non-invasive systems have traditionally been regarded as insufficient to control prostheses with multiple degrees of freedom (DOF), advances in electrodes and control systems are making this a more feasible option. Sharing control of a BMI between a subject's brain activity and automatic controllers eases the burden on the subject when attempting to control multiple DOF prostheses. In shared BMI control, the brain is in charge of high-order motor control, such as deciding where to move and initiating movement, whereas a robotic controller directs low-level motor functions, such as accuracy, machine learning, and stability [6]. A continuing challenge in shared BMI control is the development of computationally efficient algorithms.

Neural Prostheses and Interfaces

A variety of neural interfaces have been developed. The signals from the brain can be used to command simple, low DOF computer programs and complex, high DOF robotic prostheses. Neurally controlled computer keyboards and internet browsers have allowed some subjects to communicate with the outside world and brain controlled wheelchairs have restored a certain amount of mobility in paralyzed patients and amputees [4]. These solutions have made significant improvements in the lives of patients; however, there is still much more that can be done.

The current goal for BMIs is to create robust, high DOF, prosthetic arms, legs, and exoskeletons for amputees and paralyzed patients. This is much more challenging, because such neuroprostheses require a much higher rate of information flow and more advanced control systems [5]. It is important to be able to create a variety of neuroprostheses; paraplegic patients choose walking and trunk stability among the most desired motor functions whereas quadriplegic patients prioritize arm and hand function [6]. Although the main ideas behind upper limb, lower limb, and whole-body prostheses are all similar, they face different challenges. Lower limb and whole-body systems have to account for postural control, while prosthetic arms are generally concerned with high dexterity.

Sensory Feedback

Sensor and actuator systems inherently build up errors in measurement and in commanding outputs; therefore, feedback must be introduced into the system in order to ensure robust performance. Visual feedback is currently the most common form of feedback used in BMI applications [5]. While visual feedback is suitable in a laboratory setting, visual stimuli will not always be available in real-world settings. In a healthy body, tactile and proprioceptive signals provide crucial information about one's limbs. A possible solution is to mount pressure and vibration sensors on prostheses and use haptic interfaces to transmit the sensed values via direct vibrations on the body

[3]. While visual and haptic feedback may help to reduce errors in the system, they do not solve a larger, more abstract problem.

Ideally, a prosthesis should feel like a natural extension of the human body instead of an artificial limb. In order for this to occur, both tactile and proprioceptive information must be sent directly to the brain. Neuroplasticty and ICMS are areas of research that hold promise for sensory feedback [4]. Neuroplasticity allows for modification to the neural circuits, or cortical motor maps, associated with a given task. In addition to modifying the biological components of a BMI, machine-learning techniques may also be used to modify and improve the control systems used to control the neuroprostheses [5]. Neuroprostheses with advanced sensory feedback technology will incorporate an artificial sense of touch and position through the bionic limb. This will allow patients to reach a subconscious level of control over the prostheses, a crucial step towards the body's acceptance of a foreign actuator [3].

Brain-machine Interface Challenges

The development of effective BMIs is clearly an interdisciplinary subject, requiring expertise in design, electromechanics, computer science, neuroscience, biology, and chemistry. Due to the interdisciplinary nature of BMI, several challenges will have to be overcome before BMIs are robust and safe enough for daily use beyond the lab. Some of the major challenges include: creating implantable and biocompatible recording devices, increasing the number of neurons to record from, improving real-time signal processing and control algorithms, integrating accurate sensory feedback into the system, and building more advanced neuroprostheses.

Overview

The ultimate goal of this project is to create a quadrotor which can be flown by a human using only his or her thoughts. In order to achieve this goal, a reliable BMI system must be developed. The recently developed OpenBCI system was selected as data recording system for it's low cost and open-source capabilities. Before OpenBCI can be safely used with a real quadrotor, it must be tested with a quadrotor simulation.

This thesis presents preliminary work towards achieving this goal. The OpenBCI system was used in conjunction with a Python signal processing program to detect a user's right and left eyewinks. The eyewinks were used to send the quadrotor different voltage commands. These voltages translate to different thrust values, which allow the simulated quadrotor to take flight.

Chapter II focuses on the OpenBCI system and the signal processing algorithms. Chapter III goes into details about the development of a high-fidelity quadrotor simulation. This chapter includes derivations of the equations of motion, voltage-thrust relationships, gyroscopic effects, and simulation verification. Chapter IV highlights some of the results of integrating the OpenBCI system, the Python signal processing code, and the MATLAB quadrotor simulation. Chapter V discusses future improvements to this work, and the next steps required to control a real quadrotor by thoughts alone.

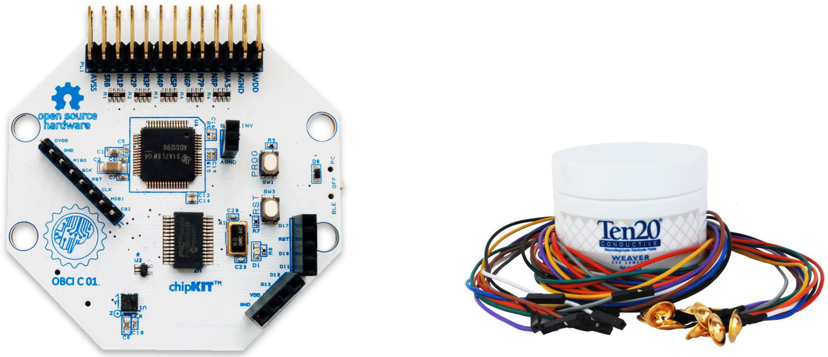
CHAPTER II

SIGNAL EXTRACTION

Essential to successful BMIs are the signal processing algorithms. This chapter details the first two elements of the BMI under development: the recording system and signal processing techniques used to extract meaningful data from the raw signal.

Data Acquisition Technology

The EMG signals are acquired using gold-cup, passive electrodes connected to the 32-bit OpenBCI board; this is a chipKIT-compatible, 8-channel neural interface with a 32-bit processor which implements the Microchip PIC32MX250F128B microcontroller. The board communicates with a computer via a Bluetooth enabled USB dongle, which is based on the RFDuino radio module. Ten20 conductive paste is applied to the electrodes in order to adhere them to the body while also improving the signal quality. The OpenBCI board and electrodes are seen in Figure II.1 [11].



(a) OpenBCI board.

(b) Electrodes and paste.

Fig. II.1. 32-bit OpenBCI board and electrodes. The USB dongle is not shown.

The OpenBCI board and electrodes were selected as the data acquisition devices due to their low cost and open-source capabilities. The board can be used to as an EEG, EMG, and an electrocardiogram (EKG).

The board records raw data in “counts”. In order to convert from counts to volts, a scale factor, SF must be used. The scale factor is dependent on the gain of the amplifier, G. For the sample data, the maximum gain of 24 was used, which results in a scale factor of 0.02235 microvolts per count. The scale factor varies with gain as shown below.

$$SF = \frac{4.5}{(2^{23} - 1)G} \tag{II.1}$$

Signal Processing

In order to capture eyewinks with EMG, electrodes are placed on both the right and left temporal muscles, near the eyeballs. This is seen in Figure II.2 [12].

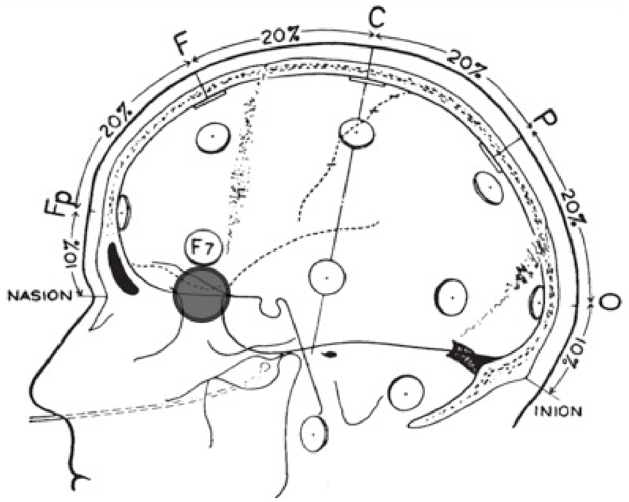


Fig. II.2. EMG electrode location is marked by the darkened circle. Located near left eye, below the F7 node. Opposite electrode located near right eye, below the F8 node (adapted from Klem et al.).

In addition to the two recording electrodes, ground and bias electrodes are attached to the two ear lobes, where minimal electrical activity is present. The OpenBCI board streams the EMG data to a computer via bluetooth and a Python program performs real-time signal processing and wink detection.

Signal Filtering

The OpenBCI board attenuates most of the 60 Hz electrical noise resulting from the alternating current in the power supply; therefore, the real-time signal processing in Python focuses on extracting information from the desired frequency bands. As with any biosignal recording instrument, OpenBCI suffers from baseline wander; this drift contaminates the lower frequencies of the data and makes it difficult to extract meaningful information. To remove baseline wander, a 0.5 Hz high-pass filter is applied using a digital biquadratic filter.

$$K = \tan(\omega_c \pi) \tag{II.2}$$

$$Z = \frac{1}{1 + K/Q + K^2} \tag{II.3}$$

$$a_0 = Z \tag{II.4}$$

$$a_1 = -2a_0 \tag{II.5}$$

$$a_2 = a_0 \tag{II.6}$$

$$b_1 = 2(K^2 - 1)Z \tag{II.7}$$

$$b_2 = (1 - K/Q + K^2)Z \tag{II.8}$$

Where ω_c is the cutoff frequency (normalized by sampling rate), Q is the quality factor, and a_x, b_x are the filter coefficients. Each sample detected by OpenBCI is filtered in real-time using these coefficients.

$$s_f = s_r a_0 + z_1 \tag{II.9}$$

$$z_1 = s_r a_1 + z_2 - b_1 s_f \tag{II.10}$$

$$z_2 = s_r a_2 - b_2 s_f \tag{II.11}$$

Where s_r is the raw sample, s_f is the filtered sample, and z_x are intermediate filter variables. Figure II.3 shows data obtained from the right temporal muscle.

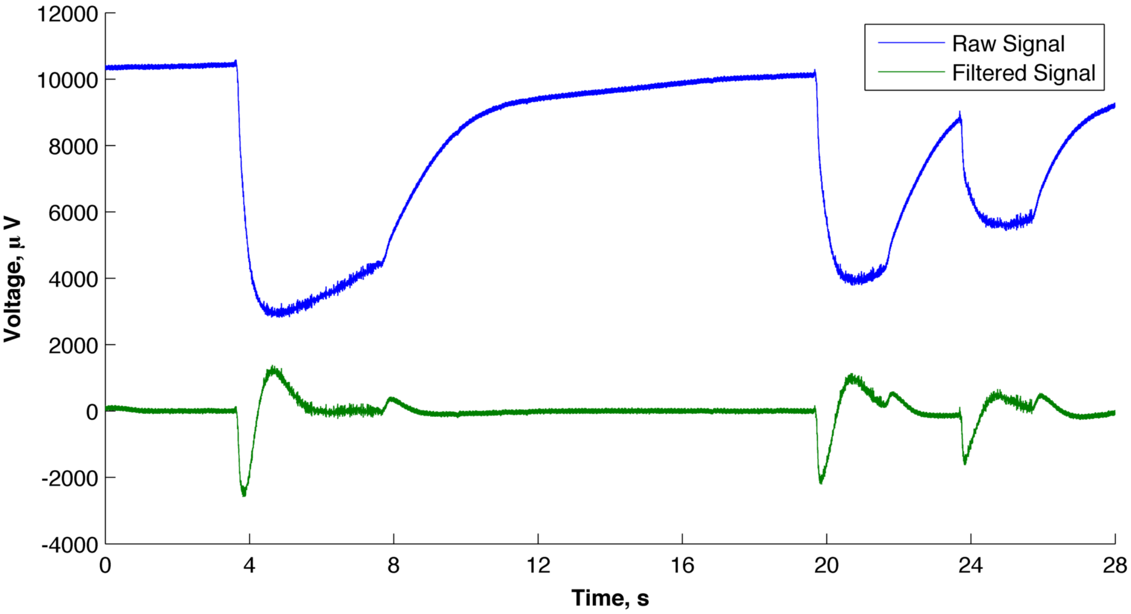
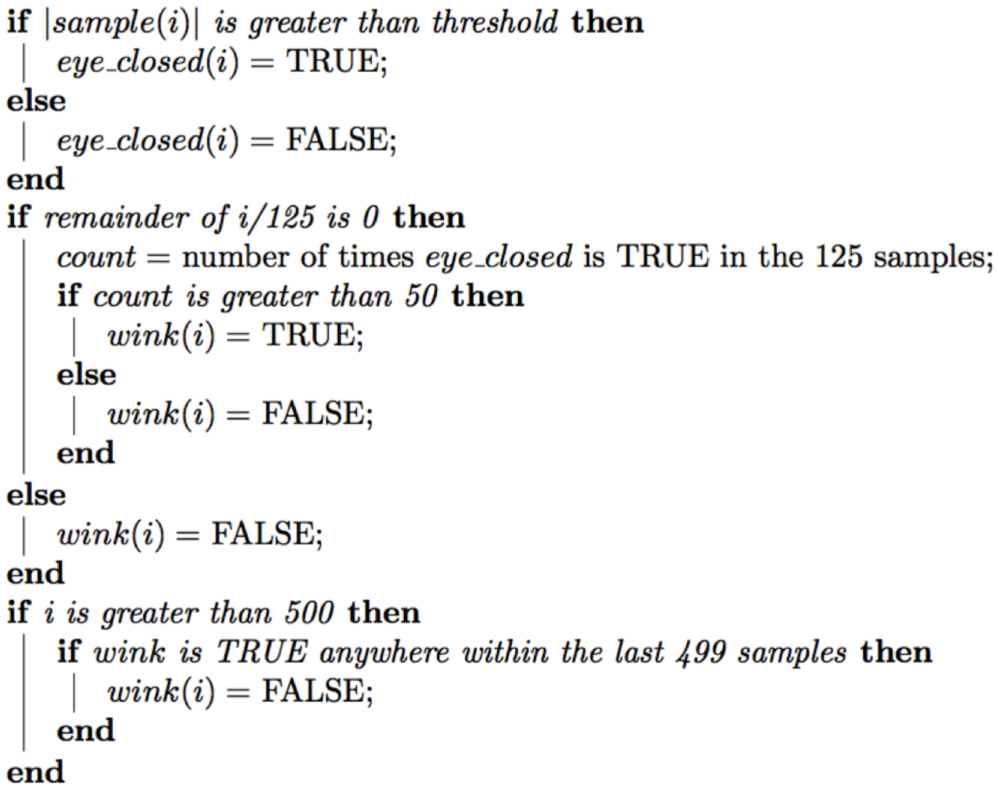


Fig. II.3. Comparison of signal before and after applying biquadratic filter. The eye is closed at 4 s, opened at 8 s, and winked at 20 and 24 s.

The raw signal shows significant baseline wander; this wander continues to increase over time. In addition to baseline wander, the raw signal does not accurately capture the complete act of closing and opening the eye during a wink. On the other hand, the filtered signal is centered around 0 μV, removing the baseline wander. In addition, the entire winking action can be identified. The large initial pulse seen at 4 s corresponds to the subject closing his right eye, whereas the smaller pulse at 8 s corresponds to the subject opening his eye again. Similar close actions can be identified at 20 s and 24 s whereas the corresponding open actions occur at 22 s and 26 s.

Wink Detection Logic

As Figure II.3 shows, the act of closing or winking an eye results in a significant amplitude change. In order to detect a wink in real-time, the following algorithm is used, where i is the sample number. The OpenBCI board detects 250 samples per second.



```
if $|sample(i)|$ is greater than threshold then
    $eye\_closed(i)$ = TRUE;
else
    $eye\_closed(i)$ = FALSE;
end
if remainder of $i/125$ is 0 then
    $count$ = number of times $eye\_closed$ is TRUE in the 125 samples;
    if $count$ is greater than 50 then
        $wink(i)$ = TRUE;
    else
        $wink(i)$ = FALSE;
    end
else
    $wink(i)$ = FALSE;
end
if $i$ is greater than 500 then
    if $wink$ is TRUE anywhere within the last 499 samples then
        $wink(i)$ = FALSE;
    end
end
```

Fig. II.4. Wink detection algorithm.

Part **A** of the algorithm simply detects when the signal is above or below a certain threshold, in this case the threshold is ±500 μV. The TRUE instances can be seen in Figure II.5, as a black x. Part **B** is invoked every half second and assigns a trigger variable to the current sample if at least 50 samples have exceeded the threshold in that time span. At this point in the algorithm, the trigger variable may be invoked more than once by a single winking action, if the initial pulse is large and long enough. Part **C** ensures that there is only a single trigger per wink by setting the trigger variable to FALSE if there is already a TRUE value within the last 2 seconds. The final trigger can be seen in Figure II.5, as a red dot. The drawback to using this method is that a wink performed within 2 seconds of a previous wink will not be recognized.

Many improvements are planned for this algorithm. The ±500 μV threshold works for the subject in the experiment; however, that threshold may not always be correct for different subjects. In order

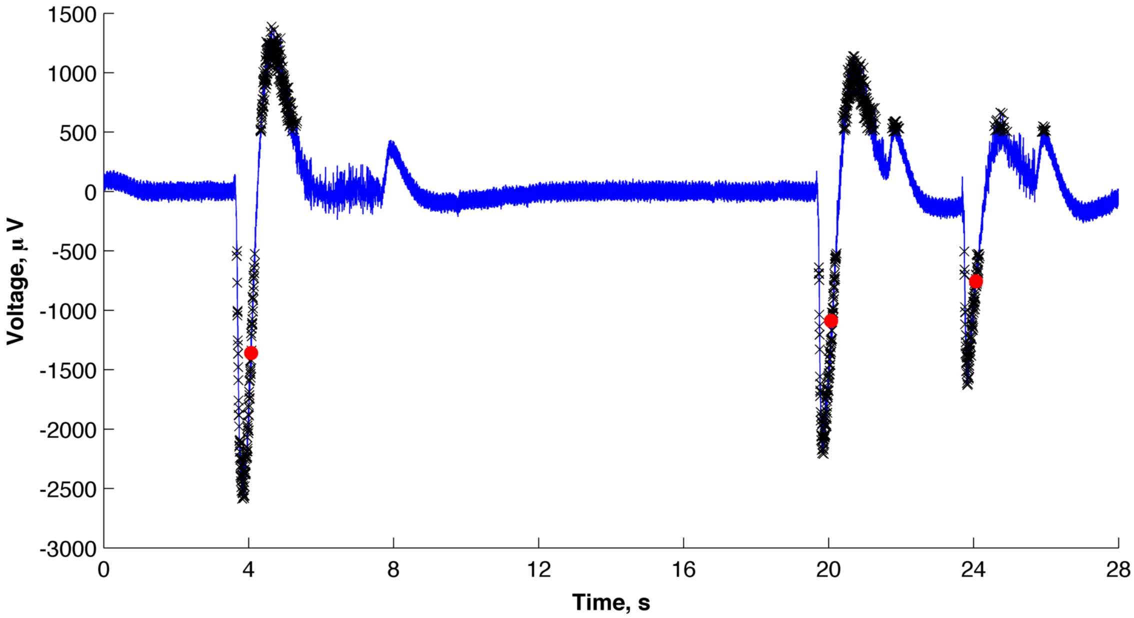


Fig. II.5. Results of the wink detection algorithm. Samples above/below the threshold are represented by a black x. The final wink trigger is shown as a red dot.

to make this algorithm robust, the threshold must be calculated for each subject automatically. This will be accomplished by analyzing the signal during a baseline period, in which the subject is staring straight forward without blinking. The winks can then be detected by comparing the waveform to the baseline signal. Another future improvement is to identify winks by detecting the peaks in the pulses, not just points above a threshold. Advanced debouncing techniques will also be implemented in order to eliminate the 2-second window in between possible detections.

CHAPTER III

QUADROTOR SIMULATION

This chapter details the third element of the BMI under development: the device being controlled by the neural signals. An open-loop quadrotor simulation was developed; the simulation takes four voltage values as input in order to fly a mathematical model of a real quadrotor.

A quadrotor is a rotorcraft with two sets of fixed-pitch propellers which are connected via four arms. Changes to the input voltage of each rotor lead to changes in the thrust provided by the rotor. A top view of the quadrotor is seen in Figure III.1. In order to balance the torques created by the individual rotors, the odd-numbered rotors rotate clockwise, whereas the even-numbered rotors rotate counter-clockwise.

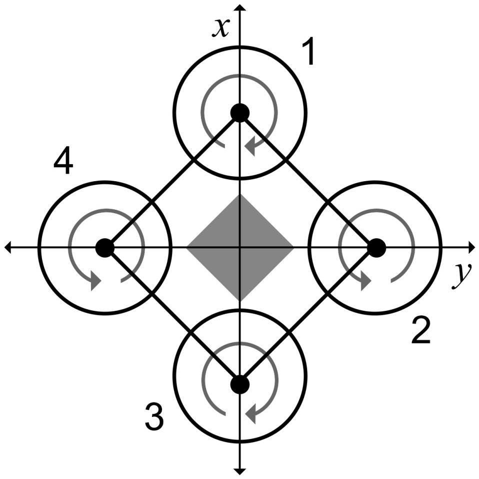


Fig. III.1. Quadrotor top view. The arrows represent the angular velocity vector of each rotor.

Table III.1 presents critical mass properties of the quadrotor. The arm length l, is used to calculate the moments due to rotor thrust. The moments of inertia I_{XX}, I_{YY}, and I_{ZZ} are used in deriving the rotational equations of motion, and the rotor moment of inertia J_r, is used to calculate gyroscopic effects.

Table III.1
Quadrotor mass properties and dimensional characteristics.

Parameter	**Value**
Mass, kg	0.5
Arm Length, m	0.2
I_{XX}, kg-m^2	7.5 x 10^{-3}
I_{YY}, kg-m^2	7.5 x 10^{-3}
I_{ZZ}, kg-m^2	1.3 x 10^{-2}
I_{XY}, kg-m^2	0
I_{XZ}, kg-m^2	0
I_{YZ}, kg-m^2	0
J_r, kg-m^2	6.5 x 10^{-5}

Simulation Structure

The MATLAB simulation is based on a pre-developed aircraft simulation [13]. It is divided into a main script file and two supporting functions. The executive script, `quadrotor.m`, calls the supporting functions. The initial state vector and control vector are defined here as well. The state vector consists of body-axis velocities, body-axis rates, Euler angles, and inertial positions. The control vector consists of the input voltages for each rotor.MATLAB's `ode45.m` integrates the equations of motion and returns important state information that can be plotted.

The primary supporting function, `eqOfMotion.m`, contains the twelve equations of motion and returns the state visualization history. These equations are discussed later. This function also changes the rotor voltages at the specified times. The direction-cosine matrix (DCM), which is shown in Eq. (III.1), is contained in `DCM.m`. This function contains the matrix that relates vectors in the body frame to the inertial, earth-fixed frame.

Deriving the Equations of Motion

Assumptions

The quadrotor model is treated as a rigid body of constant mass. All internal forces are assumed to be in equilibrium; therefore, aeroelasticity effects are ignored. As can be shown by the products of inertia in Table III.1, the quadrotor is symmetric about the XZ and YZ planes. The Flat Earth model is used, Earth is treated as inertially fixed in space, and the gravitational field is uniform and constant with altitude.

Propulsive forces and gravity are the source of the forces and moments acting on the quadrotor. Gyroscopic effects due to the rotors are included in the analysis. The quadrotor body frame follows the usual convention of North, East, Down (NED).

Orientation

In order to describe the motion of the quadrotor in terms of the inertial frame $\mathcal{N}$, where Newton's laws of motion are valid, a relationship must be found between the quadrotor body frame $\mathcal{B}$, and the inertial frame. The inertial reference frame can be rotated to coincide with the body frame using three rotation, or Euler, angles. First, the inertial frame is rotated about the $\mathbf{\hat{n}_3}$ axis. This angle of rotation is defined as yaw ψ. This new reference frame is denoted by $\mathcal{A}$. A rotation about the $\mathbf{\hat{a}_2}$ axis, or pitch θ, follows and this reference frame is $\mathcal{C}$. Finally, a rotation about the $\mathbf{\hat{c}_1}$ axis, or roll ϕ, occurs. After rotating the inertial frame in this order, it will coincide with the body frame $\mathcal{B}$. These Euler rotations are shown in Fig. III.2.

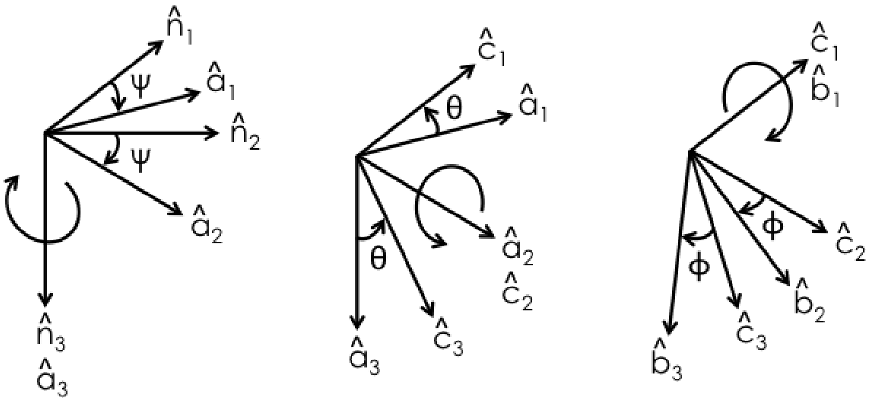


Fig. III.2. Euler rotations from inertial frame to body frame.

The DCM can be computed by multiplying the individual rotation matrices.

$$\begin{bmatrix} \hat{\mathbf{b}}_\mathbf{1} \\ \hat{\mathbf{b}}_\mathbf{1} \\ \hat{\mathbf{b}}_\mathbf{3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{n}}_\mathbf{1} \\ \hat{\mathbf{n}}_\mathbf{1} \\ \hat{\mathbf{n}}_\mathbf{3} \end{bmatrix}$$

$$\begin{bmatrix} \hat{\mathbf{b}}_\mathbf{1} \\ \hat{\mathbf{b}}_\mathbf{1} \\ \hat{\mathbf{b}}_\mathbf{3} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \hat{\mathbf{n}}_\mathbf{1} \\ \hat{\mathbf{n}}_\mathbf{1} \\ \hat{\mathbf{n}}_\mathbf{3} \end{bmatrix} \tag{III.1}$$

The DCM describes the body frame in terms of the inertial frame and is shown in Eq. (III.1). Since this is an orthogonal matrix, the transpose of the DCM is also its inverse. Therefore, in order to represent the inertial frame in terms of the body frame, the rotation matrix is simply transposed.

Equations of Motion

In order to derive the equations of motion, the body velocity vector $\mathbf{V_B}$, and angular velocity vector $\boldsymbol{\omega}$, must be defined.

$$\mathbf{V_B} = u\hat{\mathbf{b}}_\mathbf{1} + v\hat{\mathbf{b}}_\mathbf{3} + w\hat{\mathbf{b}}_\mathbf{3} \tag{III.2}$$

$$\boldsymbol{\omega} = p\hat{\mathbf{b}}_\mathbf{1} + q\hat{\mathbf{b}}_\mathbf{2} + r\hat{\mathbf{b}}_\mathbf{3} \tag{III.3}$$

The translational equations of motion in the body frame are:

$$\dot{u} = rv - qw + \frac{F_X}{m} - g\sin\theta \tag{III.4}$$

$$\dot{v} = pw - ru + \frac{F_Y}{m} + g\cos\theta\sin\phi \tag{III.5}$$

$$\dot{w} = qu - pv + \frac{F_Z}{m} + g\cos\theta\cos\phi \tag{III.6}$$

where F_X, F_Y, and F_Z are the forces acting on the quadrotor. Since the only force acting on the quadrotor is thrust, F_X and F_Y are zero. F_Z is negative of the sum of thrust for all four rotors.

The rotational equations of motion in the body frame are:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix}^{-1} \begin{bmatrix} L - qr(I_{ZZ} - I_{YY}) \\ M - rp(I_{XX} - I_{ZZ}) \\ N - pq(I_{YY} - I_{XX}) \end{bmatrix} \tag{III.7}$$

where L, M, and N are the rolling moment, pitching moment, and yawing moment, respectively.

The angular velocity components can be expressed using the rotational kinematic equations, which relate Euler angular rates to body-axis angular rates:

$$\dot{\phi} = p + \tan\theta(q\sin\phi + r\cos\phi) \tag{III.8}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{III.9}$$

$$\dot{\psi} = \sec\theta(q\sin\phi + r\cos\phi) \tag{III.10}$$

Equations (III.8)-(III.10) describe the time rate of change of the quadrotor attitude. These calculations are necessary in order to solve the translational equations of motion.

In order to describe the location of the quadrotor in the inertial frame, the transpose of Eq. (III.1) is needed. The inertial velocity vector $\mathbf{V_N}$, must also be defined:

$$\mathbf{V_N} = \dot{X}\mathbf{\hat{n}_1} + \dot{Y}\mathbf{\hat{n}_2} + \dot{h}\mathbf{\hat{n}_3} \tag{III.11}$$

Using these two relationships, the inertial velocities are found in terms of the body velocities:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ \sin\theta & -\sin\phi\cos\theta & -\cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{III.12}$$

Equations (III.4)-(III.10) and (III.12) are the twelve nonlinear equations of motion for the simulation. These equation are heavily coupled, nonlinear, ordinary differential equations which are extremely difficult to solve analytically. Instead, these equations are integrated and solved using `ode45.m` in MATLAB in order to describe the position and orientation of the quadrotor throughout time.

Voltage Model

As the voltage, V, supplied to each rotor is varied, the revolutions per minute (RPM) of the rotors change. Voltage is positively correlated with both rotor RPM and thrust. Rotationally induced drag on the rotors is also dependent on the input voltage. Applying these relations to the equations of motion, the quadrotor can be controlled solely by varying the input voltage between 0 V and 4 V.

The relationship between rotor angular velocity, Ω, and input voltage is given by:

$$\Omega = 625V^2 \tag{III.13}$$

The relationship between rotor thrust, T, and rotor angular velocity is:

$$T = 4.905 \times 10^{-8}\Omega^2 \tag{III.14}$$

The relationship between rotor drag, D_r, and rotor angular velocity is:

$$D_r = 9.81 \times 10^{-10}\Omega^2 \tag{III.15}$$

The voltage relations from Equations (III.13)-(III.15) are shown in Fig. III.3.

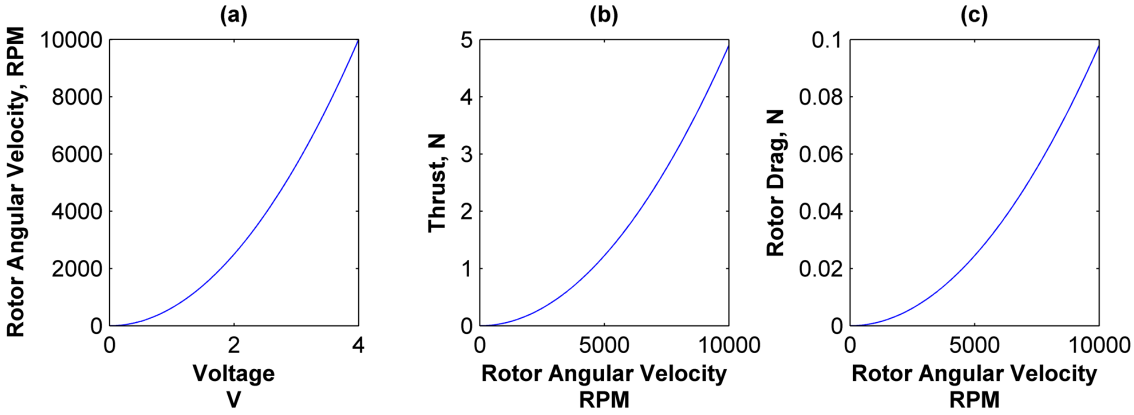


Fig. III.3. Relationships between rotor angular velocity and voltage (a), rotor thrust and angular velocity (b), and rotor drag and angular velocity (c).

Gyroscopic Effects

There are several different gyroscopic and actuator actions that produce moments on the quadrotor. The first is the body gyroscopic effect, BGE. The cross-product of body angular rates results in a net gyroscopic effect. Eqs. (III.16) show the relationship between the body rates and the BGE.

$$\begin{bmatrix} BGE_X \\ BGE_Y \\ BGE_Z \end{bmatrix} = \begin{bmatrix} \dot{\theta}\dot{\phi}(I_{YY} - I_{ZZ}) \\ \dot{\phi}\dot{\psi}(I_{ZZ} - I_{XX}) \\ \dot{\theta}\dot{\phi}(I_{XX} - I_{YY}) \end{bmatrix} \tag{III.16}$$

Note that the BGE_Z is zero since I_{XX} and I_{YY} are of equal magnitude.

Each rotor rotates about its own axis of rotation, creating moments about all three body axes of the quadrotor. The moments produced in the body X and Y axes are due to the propeller-gyroscope effect, PGE. The propeller-gyroscope effect is neglected in the Z-body axis since the counter-torque imbalance (explained later) is far more influential.

$$\begin{bmatrix} PGE_X \\ PGE_Y \\ PGE_Z \end{bmatrix} = \begin{bmatrix} J_r\dot{\theta}\Omega_r \\ J_r\dot{\phi}\Omega_r \\ 0 \end{bmatrix} \tag{III.17}$$

Where J_r is the rotor moment of inertia about its axis of rotation. Note that this equation is the PGE for a single rotor. Ultimately, all four $PGEs$ are added together, resulting in the net PGE. The counter-clockwise rotation of rotors 2 and 4 is accounted for by assigning them negative angular velocities.

The counter-torque imbalance, CTI, causes the quadrotor to yaw. The CTI is caused by drag on the rotor. The result is a drag moment that translates to the entire quadrotor. As Eqs. (III.13)-(III.15) show, the drag moment on each rotor is quadratically proportional to its angular velocity. As a rotor rotates in a particular direction, the quadrotor will have a tendency to rotate in the opposite direction.

$$CTI = (-1)^i \sum_{i=1}^{4} D_{r_i} \hat{\mathbf{b}}_{\mathbf{3}} \tag{III.18}$$

The roll and pitch actuator action moments, AA, are the main sources of rolling and pitching moment. Increasing the voltage on one of the rotors will result in a moment, which is the cross-product of l and T_i.

$$\begin{bmatrix} AA_X \\ AA_Y \\ AA_Z \end{bmatrix} = \begin{bmatrix} l(T_4 - T_2) \\ l(T_1 - T_3) \\ 0 \end{bmatrix} \tag{III.19}$$

The summation of Eqs. (III.16)-(III.19) results in the rolling, pitching, and yawing moments on the quadrotor, which are applied in Eqs. (III.7).

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} BGE_X + PGE_X + AA_X \\ BGE_Y + PGE_Y + AA_Y \\ CTI \end{bmatrix} \tag{III.20}$$

Model Verification

The 0^{th} second model verification depicted in Table III.2 is a unique method of verifying known, hand-calculated values against the values that MATLAB is producing during the simulation. Multiple voltage values are shown against the different responses; these responses are then compared to the expected physical response. For example, if no voltage (no thrust) is provided, one would expect the acceleration in the Z-direction to match gravitational acceleration. That is exactly the case in the first row. Using the 0^{th} second model verification technique, the certainty of MATLAB performing the calculations correctly and the code working as expected can be assured.

Table III.2
0^{th} second model verification in the body frame (voltage input vs. response).

V_1	V_2	V_3	V_4	$\dot{u}$	$\dot{v}$	$\dot{w}$	$\dot{p}$	$\dot{q}$	$\dot{r}$
0	0	0	0	0	0	9.81	0	0	0
2.8284	2.8284	2.8284	2.8284	0	0	0	0	0	0
2.8284	0	2.8284	0	0	0	4.9050	0	0	-3.7731
0	2.8284	0	2.8284	0	0	4.9050	0	0	3.7731
3	3	3	3	0	0	-2.6058	0	0	0
3.5	3.5	3.5	3.5	0	0	-13.192	0	0	0
1	1	1	1	0	0	9.6567	0	0	0
2.80634	2.85	2.80634	2.85	0	0	0	0	0	0.2329
2.6183	2.6183	3	3	0	0	0	17.373	-17.373	0
3	3	2.6183	2.6183	0	0	0	-17.373	17.373	0

Table III.2 shows many expected outcomes from the quadrotor. The first row shows that when zero voltage is supplied to the quadrotor, the vehicles falls with the acceleration of gravity. The second row shows that when each rotor receives 2.8284 V, the vehicle hovers in place without any acceleration. The third and fourth row show that when two rotors receive 2.8284 V, the vehicle falls with half the acceleration of gravity. In addition to falling, a yawing motion is caused by the counter-torque imbalances. The final two rows show a set of voltage values that cause the quadrotor to hover in place while rolling and pitching; this effect is caused by the actuator action moments.

This quadrotor simulation serves as an excellent starting point to test different control systems, including BMI systems. The simulation is currently still open-loop, which means that there is no feedback supplied to the system. Each quadrotor motion (such as initiating motion, maintaining a current state, or deceleration) must be manually controlled by the user. In the future, a controller will be integrated in order to reduce the workload of the user.

CHAPTER IV

OPENBCI TESTING

Experimental Setup

In the experiment, which was approved by the Texas A&M Institutional Review Board, the subject closed and winked his eyes in an arbitrary pattern. The test took place in an engineering laboratory, as opposed to a Faraday cage. A Faraday cage was not used because relying on a cage to reduce electromagnetic interference is not practical in a mobile BMI system. Electrodes were attached to the temporal muscles, as previously mentioned; channel 1 corresponds to the right side whereas channel 2 corresponds to the left side.

The OpenBCI system recorded and transmitted the biosignal to a computer at 250 Hz while a Python program filtered the signal in real-time. The program also performed wink detection at a half second delay. Once the experiment finished, a file with a time history of the detected right and left winks was generated.

The MATLAB quadrotor simulation uses the time history file to determine the voltages to apply to each rotor. For example, a right wink corresponds to a temporary increase in voltage (increase in thrust) in order to ascend for a few seconds, and then return to a hover. Similarly, a left wink corresponds to a temporary decrease in voltage (decrease in thrust) in order to descend and eventually return to a hover. The simulation always commands the quadrotor to ascend at $t = 0$ in case the user decides to descend (left wink) first.

Results

Figures IV.1 and IV.2 show the results of the test. As stated before, the quadrotor simulation commands an ascent followed by a hover command at the start of the simulation. At approximately 4 s the user closes his right eye, causing the quadrotor to ascend and subsequently return to a hover. At approximately 12 s the user closes his left eye, causing the quadrotor to descend. Finally, at 20 s and 24 s the user winks his right eye, leading to two more ascents. Although only ascent and descent maneuvers are shown here, the same time history file could be used for other maneuvers such as roll right and left, pitch up and down, or yaw right and left.

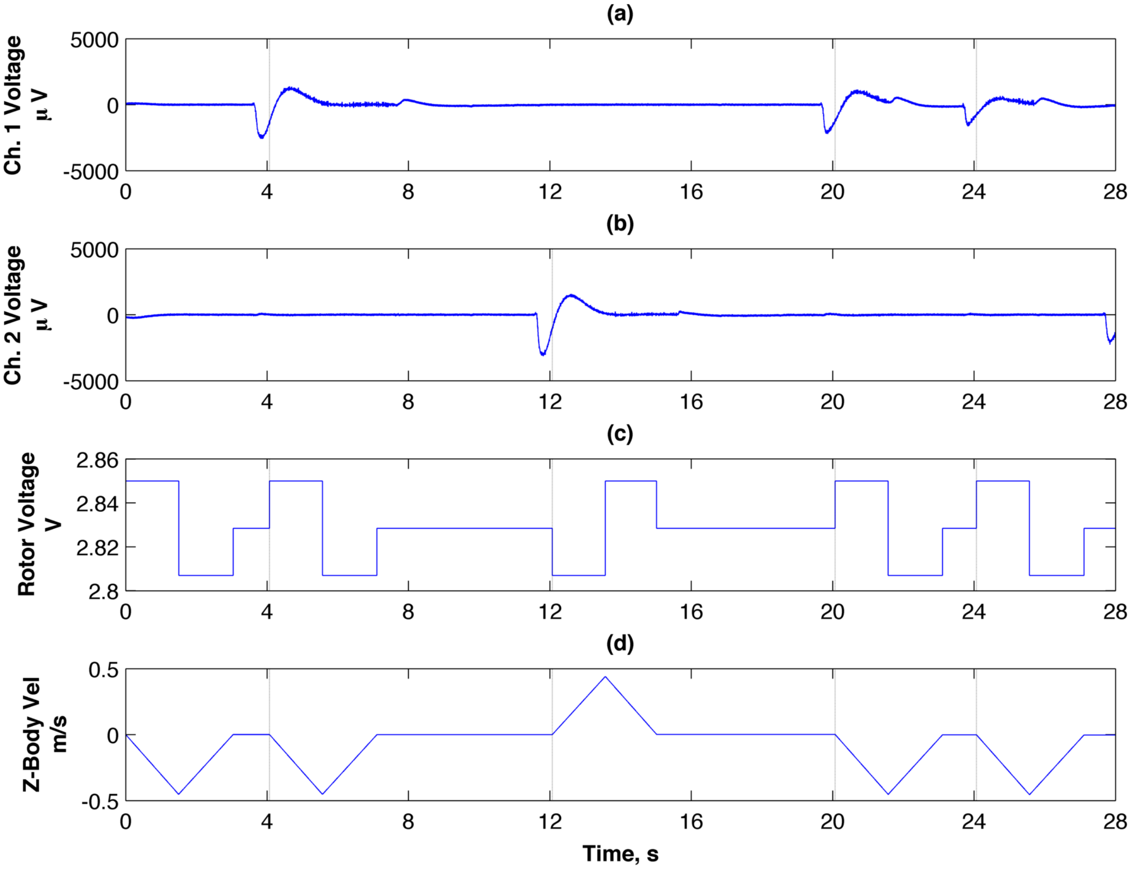


Fig. IV.1. Right eye (a) and left eye (b) signals, and resulting velocity (c) and altitude (d) change. The vertical lines represent when a wink is detected.

The detected winks can be identified by the vertical bars in the figures. Note that the accelerate up, accelerate down, and hover commands are generated by individual quadrotor voltages of 2.85 V, 2.807 V, and 2.82843 V, respectively. The quadrotor velocities that are generated from these voltage commands and result in an altitude change can be seen in Figure IV.1 (d). Note that using the NED reference frame, a negative velocity is actually an ascent velocity whereas a positive velocity is a descent velocity.

Figure IV.2 shows that the OpenBCI system, along with the Python algorithm is able to successfully command the vertical motion of a simulated quadrotor. Further work will involve increasing the amount of motions detected. One way to do this is to add recognition of double and triple wink sequences.

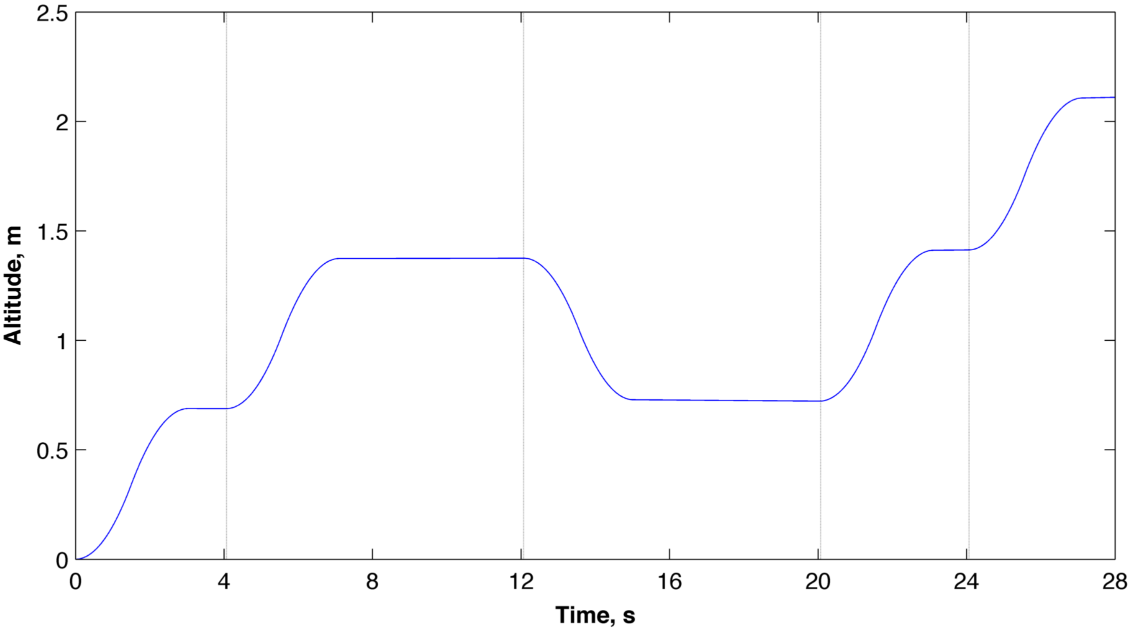


Fig. IV.2. Quadrotor altitude throughout the experiment.

Since the simulation is still open-loop, the ascent maneuver requires the simulation to generate an increase in voltage above the hover voltage in order to accelerate, a decrease in voltage below the hover voltage in order to decelerate (1.5 s later), and finally a return to the hover voltage once the vertical velocity is zero again (3.04 s later). Similarly, the descent maneuver requires a decrease in voltage, followed by an increase in voltage (1.5 s later), and finally a return to the hover voltage (2.95 s later).

CHAPTER V

CONCLUSIONS

This thesis outlined the early stages of developing a low-cost BMI system, which integrates EEG and EMG to control a quadrotor. The OpenBCI system was used to record neuromuscular activity when winking the eyes. Real-time signal processing and wink detection algorithms were developed in Python and integrated with the OpenBCI board. An open-loop quadrotor simulation was created for testing. This simulation is a physically accurate model of a quadrotor, which will eventually be built to fly in real life. The experimental results showed that it is possible to control at least the ascent and descent motion of a quadrotor using winks only.

Future Work

Various improvements to the system are currently underway. The quadrotor simulation will be ported to Python in order to integrate it with the Python signal processing and wink detection algorithms. This will result in a truly real-time BMI system, because the quadrotor simulation will be running alongside the OpenBCI signal detection, instead of afterwards. Feedback will be added to the simulation in order to close the loop and increase the controllability of the quadrotor. Real-time control using OpenBCI has been conducted successfully in the lab; the same experiment was conducted with the Aldebaran Nao robot in which right and left winks corresponded to walking right and left, respectively.

Peak detection algorithms will be used alongside advanced debouncing techniques in order to eliminate the 2-second window in between possible wink detections. In addition, the program will check for winks more often than every half second. This will increase the number of available commands by allowing subjects to take advantage of single, double, or even triple winks.

An alternative BMI setup that is currently planned will use VEP in addition to winks. VEPs can be recording by placing electrodes on the scalp along the occipital lobe. If a subject concentrates on a screen which is flashing at a particularly frequency, the waveforms picked up around the occipital lobe will match that frequency. A subject concentrating on a screen sectioned into four different flashing frequencies can choose a particular frequency based on the desired action. Combining this with the basic wink commands, a four-frequency screen can result in 8 different commands.

The ultimate goal is to attach electrodes to the motor cortex in order to send commands by thought alone. As mentioned before, this training may take months to complete, so this goal is much more challenging. Once pure thought can be demonstrated to be robust enough to control the simulation, it will be tested on a real quadrotor.

Although the OpenBCI system is being used to control a quadrotor, the success of this project has much larger implications. If successful, OpenBCI can be used to create low-cost EMG and EEG controlled prostheses. This will be particularly useful for child amputees. Since children's bodies grow so quickly, it is not feasible to constantly purchase new expensive prostheses. OpenBCI could be used in conjunction with 3D printing to create custom neurally controlled prostheses that may be modified as the child grows. As a low-cost BMI, OpenBCI has the potential to revolutionize the prostheses industry.

REFERENCES

[1] One degree of separation: Paralysis and spinal cord injury in the united states. Print, Christopher & Dana Reeve Foundation, 2009.

[2] Kathryn Ziegler-Graham, Ellen J MacKenzie, Patti L Ephraim, Thomas G Travison, and Ron Brookmeyer. Estimating the prevalence of limb loss in the united states: 2005 to 2050. *Archives of physical medicine and rehabilitation*, 89(3):422–429, 2008.

[3] Mikhail A Lebedev and Miguel AL Nicolelis. Brain–machine interfaces: past, present and future. *TRENDS in Neurosciences*, 29(9):536–546, 2006.

[4] José del R Millán and J Carmena. Invasive or noninvasive: understanding brain-machine interface technology. *IEEE Engineering in Medicine and Biology Magazine*, 29(EPFL-ARTICLE-150426):16–22, 2010.

[5] Parag G Patil and Dennis A Turner. The development of brain-machine interface neuroprosthetic devices. *Neurotherapeutics*, 5(1):137–146, 2008.

[6] Mikhail A Lebedev, Andrew J Tate, Timothy L Hanson, Zheng Li, Joseph E O'Doherty, Jesse A Winans, Peter J Ifft, Katie Z Zhuang, Nathan A Fitzsimmons, David A Schwarz, et al. Future developments in brain-machine interface research. *Clinics*, 66:25–32, 2011.

[7] Rajesh PN Rao. *Brain-computer Interfacing: An Introduction*. Cambridge University Press, 2013.

[8] Thomas J Sullivan, Stephen R Deiss, Tzyy-Ping Jung, and Gert Cauwenberghs. A brain-machine interface using dry-contact, low-noise EEG sensors. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 1986–1989. IEEE, 2008.

[9] Jing Wang, Oliver A Kannape, and Hugh M Herr. Proportional emg control of ankle plantar flexion in a powered transtibial prosthesis. In *IEEE International conference on rehabilitation robotics (ICORR), Seattle, USA*, 2013.

[10] Todd A Kuiken, Guanglin Li, Blair A Lock, Robert D Lipschutz, Laura A Miller, Kathy A Stubblefield, and Kevin B Englehart. Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms. *Jama*, 301(6):619–628, 2009.

[11] Openbci 32-bit board kit (chipkit$^{\text{TM}}$-compatible), 2015. [Online; accessed 12-March-2015].

[12] George H Klem, Hans Otto Lüders, HH Jasper, and C Elger. The ten-twenty electrode system of the international federation. *Electroencephalogr Clin Neurophysiol*, 52(suppl.):3, 1999.

[13] Alejandro F Azocar and John Valasek. High fidelity simulation of a nonlinear aircraft. In *Proceedings of the AIAA Science and Technology Forum and Exposition 2014: 52nd Aerospace Sciences Meeting, National Harbor, MD, USA*, 2014.