Title

Classification, Tracking, and Suppression of Parkinsonian Tremor

Author

Eliahu, Daniel Shalom

Publication Date

2017

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

CLASSIFICATION, TRACKING, AND SUPPRESSION OF PARKINSONIAN TREMOR

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING
with an emphasis in ROBOTICS & CONTROL

by

Daniel S. Eliahu

September 2017

The Thesis of Daniel S. Eliahu
is approved:

Professor Gabriel H. Elkaim, Chair

Professor Dejan Milutinovic

Professor Renwick E. Curry

Doctor Lin Zhang

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Contents

List of Figures

List of Tables

Abstract

Classification, Tracking, and Suppression of Parkinsonian Tremor

by

Daniel S. Eliahu

Approximately 11 million Americans live with Parkinson's Disease (PD) or Essential Tremor (ET), and the current standard of care is inadequate. Diagnostic methods rely on qualitative observations rather than quantitative metrics and treatments are often prohibitively expensive or riddled with side effects. This work first develops a non-intrusive wearable device that can perform an automated diagnosis of PD and ET. Next, it develops an algorithm for attitude estimation of the human wrist. Finally, it develops a prototype for a wearable tremor suppression device. The diagnostic device was tested with a group of 30 healthy volunteers who had been educated on the motion characteristics of PD and ET. A 99.9% classification accuracy was achieved during this test. The attitude estimation algorithm was tested with the help of a mechanical test rig that simulated tremor. During a highly dynamic oscillation, the attitude estimate achieved an RMS error of less than 1 degree. This same mechanical test rig was used to evaluate the effectiveness of the tremor suppression prototype. Settings on the rig were adjusted to test tremors of varying amplitude and frequency. The device was able to attenuate tremor amplitude by an average of 60.96% across the configurations tested.

Acknowledgments

This work would not have been possible without the support of many amazing individuals and organizations. I would first like to thank Citris and the Banatao Institute for generously funding this research.

My sincere gratitude also goes to my thesis advisor, Professor Gabriel Elkaim, for always inspiring me to work harder than I thought I could. When I needed his advice most, the door to Gabe's office was always open, and his immense knowledge never failed to steer me in the right direction.

A warm thank you also goes to Professor Dejan Milutinovic, for his insightful comments, hard questions, and commitment to teaching. To Professor Renwick Curry, thank you for the patience and support you gave me, as well as the selfless contributions you made to all members of the Autonomous Systems Lab.

I would also like to thank Dr. Lin Zhang and his team of experts at the UC Davis Medical Center. Lin's insight and expertise were invaluable during all phases of this work. Without his dedication, we would still be at square one.

To the entire Autonomous Systems Lab, thank you for your encouragement, ideas, tea, and coffee.

Finally, an enormous thank you to my family and friends. Aba, Ema, David, and Moran, your unconditional love and support have made this milestone possible. To my friends new and old, thank you for always blessing me with an excuse to take a break. Lastly, thank you to Pat for being an editor extraordinaire.

Chapter 1

Introduction

Approximately 11 million Americans live with Parkinson's Disease (PD) or Essential Tremor (ET), and unfortunately, current treatment options have yet to return full quality of life to these patients [1]. No treatments exist to cure or reverse the effects of these debilitating, chronic diseases; the current standard of care involves only symptomatic management. Perhaps the most notorious of these symptoms is the oscillatory hand, wrist, and arm tremor present in 80% of autopsy-confirmed diagnoses of ET and 70% of PD [11, 12]. In the case of PD, the accompanying bradykinesia (slowness of movement), rigidity (muscle stiffness), and postural instability make it difficult for patients to maintain their independence; a problem that is aggravated by the increasing cost of senior care. ET patients, on the other hand, suffer from kinetic and/or postural tremors in their arms and hands which significantly affects their ability to perform daily activities. The Parkinson's Disease Foundation estimates that the combined direct and indirect cost of PD in the United States is approximately $25 billion [1]. Therapeutic interventions, which account for a substantial portion of this cost, fall into two categories:

1. **Pharmacologic treatments:** The primary focus of these treatments is on correcting the dopaminergic disturbances in the brain. However, even the most time-tested dopaminergic drugs can frequently cause side effects such as hallucinations, dizziness, nausea,

insomnia, and compulsiveness [2], which rules out their use for many PD patients. For patients that can tolerate the side effects, these drugs are still unable to completely correct the behavior of the basal ganglia and return fully stabilized motor function [19]. Over time, the effectiveness of dopaminergic treatments decreases and sensitive adjustment of medication dosage is required. For these reasons, the currently available drugs are not considered an ideal long-term solution for the treatment of PD. Pharmacologic treatments of ET require the use of tremor suppressants, which usually have modest benefits in tremor control at the expense of poor tolerability. Common side effects include drowsiness, cardiac implications, and cognitive blunting.

2. **Surgical interventions:** Surgical interventions for both PD and ET are still being developed. The most promising option to date is deep brain stimulation (DBS) [32]. This procedure must be performed by an experienced neurosurgeon and is both costly and not medically suitable for many patients due to its invasive nature. DBS is also only effective in 71% of patients [16]. Therefore, despite yielding remarkable results when effective, DBS is not currently able to service the entire population of patients.

The current attitude in the medical community is that neither treatment method adequately and conveniently addresses patient needs. Moreover, both types of treatment rely on a correct diagnosis, but even this can be difficult to attain. PD and ET share a number of common symptoms, and this overlap leads to many misdiagnoses, especially in the early stages of disease onset.

Parkinsonian tremor initially presents itself as a 3-7Hz “pill-rolling” motion of the fingers, then slowly progresses to forearm pronation/supination and finally elbow flexion/extension [11]. These motion profiles are pictured in Fig. 1.1. Many cases of PD are also characterized by other, non-tremor symptoms. These can include bradykinesia, rigidity, postural instability, lack of facial expression, and in some cases dementia. The presence of these other symptoms can aid a doctor in making a confident diagnosis. However, in their absence, diagnoses must be made based on tremor characteristics and family history alone.

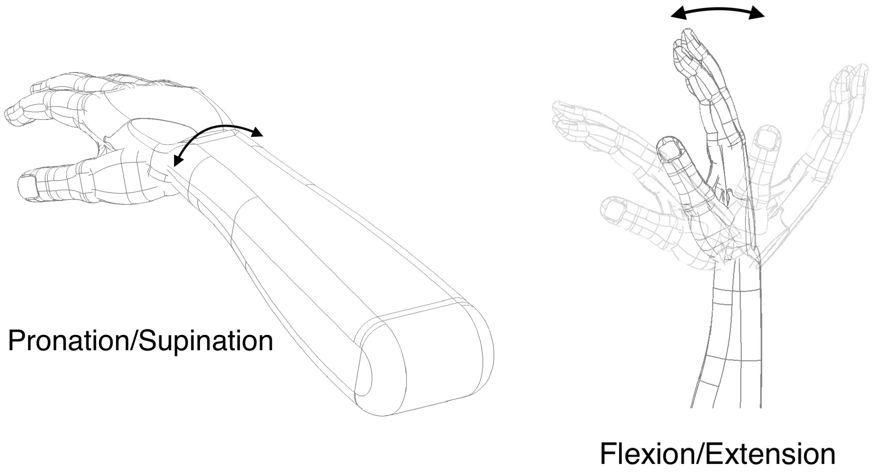


Figure 1.1: Illustration of two common forms of tremor induced motion. Pronation and supination are most commonly associated Parkinson’s Disease, while flexion and extension are most commonly associated with Essential Tremor.

ET tremors cause similar movement patterns, but with one distinct difference: ET presents as flexion/extension of both the forearm and elbow at a frequency of 4-12Hz (most commonly 8-12Hz), in contrast to the forearm pronation/supination in PD. Furthermore, while the tremor can spread or intensify, ET has no accompanying symptoms.

In cases where a patient has an arm or wrist tremor with no accompanying symptoms, doctors strive to determine whether the forearm is in flexion/extension or pronation/supination. Unfortunately, if the tremor is small in amplitude, this crucial distinction can be hard to discern with the naked eye. As a result, it is common for doctors and patients to wait until the tremor has intensified before starting treatment. A frequently pursued alternative is to start a patient on a course of treatment specific to either PD or ET. If a patient responds well to the treatment, doctors can deduce which disease the patient has and reach a diagnosis.

Neither of these diagnostic methods inspires confidence. Waiting for a tremor to intensify is the more conservative approach, but it delays treatment and allows the patient's condition to deteriorate. Such a delay ultimately reduces the effectiveness of treatment once a diagnosis is made. On the other hand, the prescription of drugs to preempt a diagnosis can also carry significant harm. Despite pharmacological advancements, the most advanced drugs on the market still cause uncomfortable or unbearable side effects, whether or not the drug is successful at treating the original condition.

The current shortcomings, in both the diagnosis and treatment of idiopathic tremor, are the primary motivation behind this work. While the underlying tremor-causing diseases are rooted in chemical, biological, and genetic aberrations, their symptoms are largely mechanical in nature. Thus, the decades of knowledge built around the identification, tracking, and stabilization of mechanical systems can be applied directly toward the effort of improving medical care.

The first chapter of this thesis addresses the challenge of tremor diagnosis and quantification. It uses inertial measurement units (IMUs) to develop a non-intrusive, wearable sensor system. The system comprises four IMUs, a microcontroller, data logger, and power supply all mounted to a flexible glove and wristband. The system is capable of measuring and recording the complete motion characteristics of the wrist and hand.

To validate the system, healthy volunteers were recruited and asked to mimic the motion of PD and ET. Procedures were then developed to extract key features from the raw and unprocessed data. These features were used to train classification algorithms with the goal of correctly classifying unknown samples. Different classification algorithms are compared and evaluated for their accuracy and diagnostic practicality.

Using the hardware developed for the diagnostic device, Chapter 3 then develops a new algorithm for attitude estimation of the human wrist. It leverages the work of several researchers in the aerospace community and makes multiple domain specific improvements. The attitude filter is designed as a two-stage, multiplicative Kalman filter. It receives raw measurements from a six degree-of-freedom IMU and outputs an estimate of the sensor attitude. This estimator is validated offline with experiments in Matlab and tested in real time on a microcontroller.

Finally, Chapter 4 of this thesis begins work toward a tremor suppression device. The device mimics the design of momentum-based reaction wheels commonly found onboard orbiting satellites. To test the device while avoiding risk to patients, a mechanical rig is developed to simulate the tremor induced by Parkinson’s disease. At the time of writing, the tremor suppression device is still a very early stage prototype. Its primary purpose is to show that the current design merits further inquiry.

Chapter 2

Tremor Classification

The diagnostic standard of care for the differential diagnosis of Parkinson's Disease (PD) and Essential Tremor (ET) is currently not adequate. The shortcomings are caused by a lack of diagnostic tests, due in part to an insufficient understanding of the underlying diseases and their associated symptoms. Current knowledge indicates that PD typically presents as a 3-7Hz tremor and ET as a 4-12Hz tremor, but the specific motion characteristics of individual limb segments have not been well quantified. Doctors have qualitatively observed that PD is most often characterized by a pronation/supination of the wrist and ET by a flexion/extension of the wrist (see Fig. 1.1). It would, however, be an important step forward if doctors could reach diagnoses based on more quantitative data.

Toward this goal, researchers have attempted to enhance their understanding of the diseases by analyzing surface electromyography (sEMG) data from tremor patients [33]. These results are encouraging, but the use of sEMG sensors is limited for two main reasons. First, the muscle groups responsible for causing upper limb tremor are distributed throughout the hand, forearm, and upper arm. One or more sEMG sensors would need to be placed over each of these muscle groups to obtain a complete picture of the tremor, making the system intrusive and uncomfortable for a patient. The setup time required for such a system would also limit its utility in a typical clinical setting. A second disadvantage to the use of sEMG sensors is the cost; a single clinical grade sEMG sensor a factor of 10 more expensive than the sensors used in this work.

While diagnostic aids utilizing other technologies are sometimes available to doctors [11], they are often expensive and/or invasive. This chapter develops a new means for the objective quantification of Parkinsonian Tremor as well as an algorithm for the automated diagnosis of PD and ET. It leverages low-cost, non-intrusive sensors that are capable of measuring the key motion parameters of the limb. They are small, safe, and well suited for use in a clinical setting. Moreover, the sensors only need to be placed on the hand and wrist, yet are still able to record the full characteristics of tremor-induced motion.

2.1 Background

In order to improve the diagnosis of tremor, inertial measurement units are used to detect and record the motion of the hand and wrist. Many of the design choices within this chapter are driven by the specific characteristics of these sensors. Therefore, in order to leverage these sensors completely, a detailed understanding of both their advantages and limitations is needed. This background section provides a brief introduction to inertial measurement units as a foundation for the prototype device designed in this work.

After data has been collected from the inertial measurement units, an algorithm must be developed that can perform an accurate diagnosis of the underlying disease. At the current stage of this research, the goal is to perform a differential diagnosis between PD and ET. Therefore, the algorithm must be able to classify unknown samples into one of these two categories. This task falls under the well-researched field of study known as data classification. An introduction to the methods and practices of data classification is also provided in this background section.

2.1.1 Inertial Measurement Units

Inertial measurement units (IMUs) are electronic components that combine many sensors into one package. Together, these sensors are capable of measuring specific force, angular rotation rates, and the effects of an external magnetic field (depending on the specific configuration). The demand for IMUs, both in the aerospace community and in the consumer electronic market, has led to the widespread availability of micro electro-mechanical systems (MEMs) IMUs. Current manufacturing technology allows electronics suppliers to place a three-axis accelerometer, three-axis gyroscope, and a three-axis magnetometer together in a single, low-cost package. Such integrated sensors can be purchased for less than USD 5.00 (quantity 1000). The affordability, small size, portability, and convenience of these sensors make them an ideal choice for use on a clinical device.

While MEMs IMUs are popular for their small size and low cost, they have a number of limitations. At the low end of the cost spectrum, IMUs typically suffer from relatively high noise and bias levels. These noise characteristics differ between the individual sensors that compose the IMU.

2.1.1.1 Accelerometers

Accelerometers are capable of measuring the specific force applied to the sensor. An accelerometer reading, given by

$$\vec{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{2.1}$$

measures the inertial accelerations on the sensor (in the coordinate frame of the sensor) minus the force of gravity. This relationship is given by the equation below:

$$\vec{a}_{meas} = \vec{a}_{true} - \vec{g} \tag{2.2}$$

Where $\vec{a}_{meas}$ is the measured acceleration, $\vec{a}_{true}$ is the inertial acceleration and $\vec{g}$ is the acceleration of gravity. Therefore, if the sensor is not accelerating in inertial space, an ideal accelerometer will measure "$-g$". Likewise, if an ideal accelerometer were allowed to fall freely in a vacuum, the recorded measurement would be zero.

A typical MEMs accelerometer is corrupted by DC bias, scaling errors, axis misalignment errors, and noise that is assumed to be Gaussian in nature. Therefore, equation (2.2) becomes

$$a_{meas} = [S_F][R](a_{true} - g) + b_0 + v_a \tag{2.3}$$

Where $[S_F]$ is a diagonal matrix of scaling factors, $[R]$ is the axis misalignment, b_0 is the DC bias, and v_a is the Gaussian noise. If the accelerometer is calibrated offline as described by Elkaim et al. [14], the scale factor and axis misalignment can be removed. Moreover, for the methods used in the chapter, the DC bias is removed during feature extraction process. Thus, equation (2.3) becomes:

$$a_{meas} = a_{true} - g + v_a \tag{2.4}$$

Where v_a is the Gaussian noise.

2.1.1.2 Gyroscopes

MEMs gyroscopes measure inertial rotation rates. In the case of PD, a disease that causes involuntary pronation and supination of the wrist, the ability to directly measure the rotation rate of the limb is advantageous. However, a MEMs gyroscope measurement suffers from axis misalignment errors as well as DC bias and Gaussian noise. As before, the axis misalignment can be calibrated offline. Unlike accelerometers, the DC bias is time-varying and highly temperature dependent. Therefore, it is not sufficient to calibrate this quantity offline. The tracking of this bias will become relevant in Chapter 3.

With this in mind, a gyroscope measurement, written as

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.5}$$

can be expressed by:

$$\omega_{meas} = \omega_{true} + b(t, T, c) + v_\omega \tag{2.6}$$

where ω_{true} is the true rotation rate, $b(t, T, c)$ is a function representing the sensor bias, and v_ω is the Gaussian noise. The bias term $b(t, T, c)$ is a function of time, temperature, and cost of the gyroscope. It may also contain a DC component. While the DC bias can be calibrated before data collection, the time and temperature varying components cannot be so easily removed. In order to obtain an accurate gyroscope measurement, the gyroscope bias must be estimated in real time. For the methods used in the chapter, however, all errors other than the Gaussian noise are assumed to be removed during feature extraction process. Therefore, equation (2.6) becomes:

$$\omega_{meas} = \omega_{true} + v_\omega \tag{2.7}$$

2.1.2 Data Classification

Data classification is the practice of separating data samples into one or more categories. It is a well-researched field of study with tools and techniques that have been thoroughly refined by the data mining and machine learning communities. It is relevant to this work because the ultimate goal of this chapter is to develop a device that can automatically classify chronic tremor. Once data has been collected by the IMUs described above, the raw measurements will be processed with many different data classification strategies.

As a necessary prerequisite to a successful classification algorithm, feature extraction must first be used to extract relevant features or attributes from the high volume of raw data. There are a number of ways to extract features from a dataset, but the process is primarily guided by domain-specific intuition. All subsequent computations and analyses are then conducted using the extracted feature vectors.

After feature extraction is complete, the task becomes designing an algorithm to perform an accurate classification of the feature vectors. The remainder of this section will present a background on supervised binary classification. In this context, the term "binary" indicates that every sample can only belong to one of two classes (as opposed to multi-class classification). The term "supervised" indicates that all training data is pre-tagged with a corresponding class. It should also be noted that different feature extraction methods may lead to drastically different results during classification.

There are two general categories of classification algorithm available to a data scientist: parametric and non-parametric. As the name implies, parametric schemes fit a parametric model to the data. Training data is used to learn a set of parameters (or weights) that are then used to classify unknown samples. This method is advantageous because there are usually fewer parameters to learn than features in the original dataset. In contrast, non-parametric methods do not fit a model to the data. Instead, training data is stored or "memorized," and predictions are made by comparing a test sample with the stored training samples. Non-parametric algorithms usually require more computational memory but often provide an improved classification of non-linear data. This thesis will investigate the use of both parametric and non-parametric algorithms in an effort to maximize classification accuracy. Specifically, a K Nearest Neighbor algorithm will be implemented, and a new parametric algorithm will be developed.

Before testing the classification algorithm with real data, it is important to segment the data into three sets: training, validation, and testing. The training and validation sets are used to explore the effects of different techniques on the data. Researchers have access to these sets throughout the development of an algorithm and use them to discover relationships in the data. In the context of this thesis, various classification algorithms will be tested and tuned with the training and validation sets.

The test data is kept separate from the rest of the data and is only used as a final benchmark for the developed algorithms. This approach is advantageous because it guards against over fitting an algorithm to the training data. If the test data is selected at random, the assumption is made that it is independent of, but similarly distributed to the training/validation set. By keeping the test data segregated throughout the tuning of the algorithms, researchers can use the test set to check if the final algorithm is over-fit to the training data. Moreover, because the test set has never been seen by the algorithm, it can be used to make generalizable conclusions as to the effectiveness of an algorithm.

2.2 Sensor System

With the necessary principles behind IMUs and data classification established, the development of a system for the diagnosis of Parkinsonian tremor becomes the primary focus. In this section, a prototype device is built that can measure and record tremor data in the arm, hand, and wrist of a patient. The device uses low-cost IMUs and is designed to be comfortable and non-intrusive. The design of this sensor system is split into hardware components and software modules. This section describes the hardware and software systems required to collect data. The algorithms that process this data are explored in Section 2.5.

2.2.1 Sensor System Hardware

The sensor system must be able to measure the motion caused by Parkinsonian Tremor precisely. Substantial care is taken to make the device comfortable, lightweight, and non-intrusive. The device is also completely passive, as it simply records data. Wherever possible, soft, flexible fabrics are used. In order to make the device as comfortable as possible, it is split into three separate components: an armband, wristband, and glove. The full system is pictured in Figs. 2.1 - 2.2 and each component is explained in detail below.

2.2.1.1 Arm Band

The “brain” of the system is composed of a microcontroller and a data-logger. Together, these components are approximately the size of a deck of cards. They are mounted to a

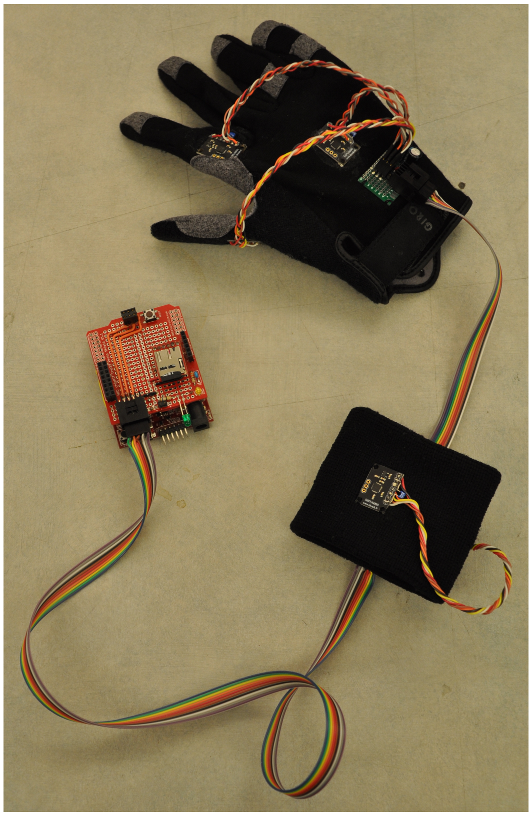


Figure 2.1: The full sensor system. Pictured here is the glove, wristband and microcontroller. Together these components are able to measure and record the complete motion of a users hand and wrist.

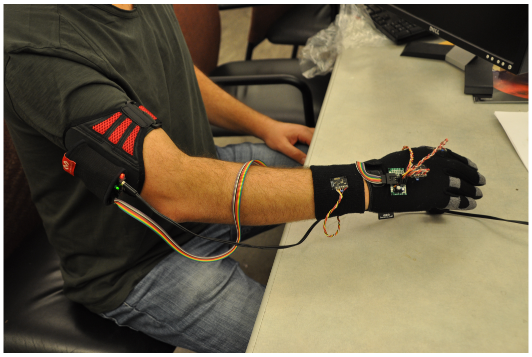


Figure 2.2: The full sensor system as worn by a student volunteer. The system measures the motion of the hand and wrist in a lightweight, comfortable, and non-intrusive manner.

lightweight armband that fits comfortably around the upper arm of the user. The microcontroller used is a Microchip PIC32MX320F128H. The Chipkit UNO32 development board was used as a convenient package for the microcontroller, featured in Fig. 2.3.

Figure 2.3: UNO32 development board. This board contains a Microchip PIC32MX320F128H and provides a convenient interface with the chip.

An UNO32 protoshield was utilized to provide an easy interface to the development board. It is a simple shield that fits on top of the UNO32 development and provides an area to create custom circuitry. It is pictured in Fig. 2.4

The Sparkfun OpenLog is a simple data logging device used to save data to a microSD card. It has many configuration settings, but for this project, it is set to immediately record all information it receives. Data is sent to the OpenLog via a serial communication protocol. The completed proto shield and data logger coupled to the UNO32 board is shown in Fig. 2.5.

2.2.1.2 Wristband

The second component is a lightweight, flexible wristband with a single Inertial Measurement Unit (IMU) mounted on it. A breakout board was used to interface with this chip

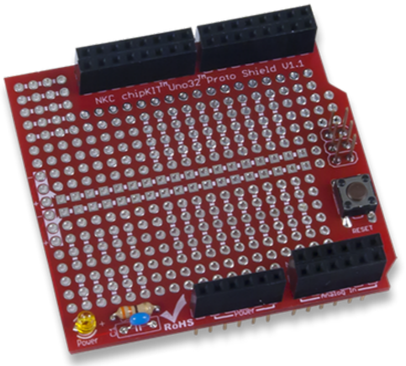


Figure 2.4: UNO32 protoshield. This component is a simple pre-made shield that fits on top of the UNO32 and provides a development area to create custom circuitry.

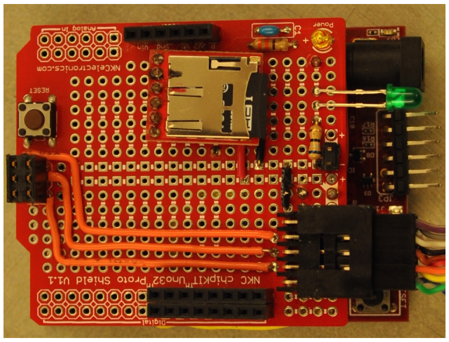


Figure 2.5: Customized shield attached to the UNO32 development board. The silver component toward the top of the board is the OpenLog data logger. The black terminal in the bottom right corner of the board is the port connecting this board to the sensors in the glove and wristband. Together with the microcontroller, these components orchestrate all timing and data sampling within the system.

(Invensense MPU6000). It is approximately 0.5in x 1in and is equipped with a 3-axis accelerometer and a 3-axis gyroscope to measure the acceleration and rotation rates of the wrist.

The MPU6000 communicates via an SPI protocol. In order to route this communication signal from the microcontroller to the IMU, a custom circuit board is also mounted to the wristband. This board sends the necessary signals to the IMU and passes the rest of the signals to the third component of the system (a glove which is described below). There is a flexible, detachable wire that connects the armband to the board on the wristband. The completed wristband can be seen in Fig. 2.6-2.7.

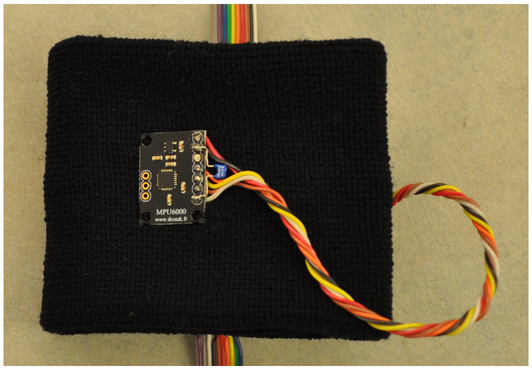


Figure 2.6: Top side of the completed wristband. A single IMU is mounted to this band and is used to record the motion parameters of the wrist. The band itself is made from a soft flexible fabric.

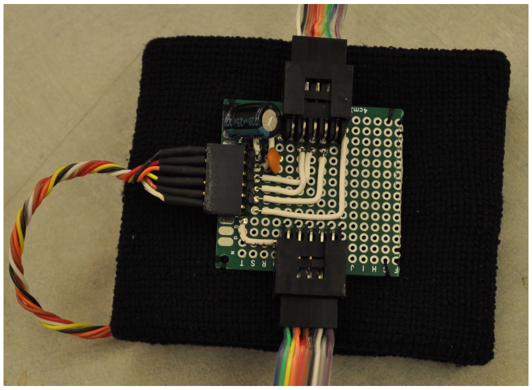


Figure 2.7: Bottom side of the completed wristband. A custom circuit board is mounted to this band. This board sends the necessary signals to the IMU and passes the rest of the signals to the glove.

2.2.1.3 Glove

The third component is a lightweight, flexible glove. The glove is outfitted with three IMUs located on the back of the glove, the index finger, and the thumb. In order to route this communication signal from the wristband to the IMUs, a custom circuit board is also mounted to the glove. There is a flexible, detachable wire that connects the glove to the wristband. The completed glove is shown in Fig. 2.8.

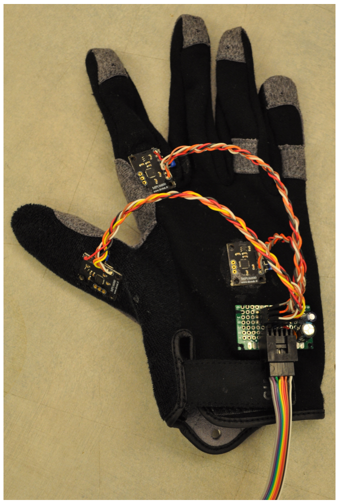


Figure 2.8: The completed glove. There is an IMU mounted on the back of the hand, index finger, and thumb. These sensors are able to measure the motion parameters of a patient's hand.

2.2.2 Sensor System Software

The hardware design of the sensor system provides all the physical elements necessary for the recording of tremor data, but software is needed to orchestrate the interaction between the system components. The software must be able to sample data from the IMUs, save the data to the on board memory card, and extract the data to a computer once a test is complete. It also must ensure that these events are properly timed and synchronized. The software required to perform these tasks is split into four modules described below. The IMU Communication module, Data Logging module, and Timing module are written in C and run in real time on the microcontroller. The Data Decoding module is written in python and runs offline on a desktop or laptop computer.

2.2.2.1 IMU Communication

The MPU6000 IMU can communicate via SPI or I2C communication protocols, but for this project only SPI is viable. When using I2C, only one bit of the sensor address can be manually set. Therefore, a single I2C network can include a no more than of two of these sensors, and this project requires the use of 4 IMUs.

The MPU communication module is responsible for initializing each sensor and querying the sensors for data. Sensor initialization must be completed every time the system is powered on. The initialization process requires the sensors to be placed in SPI mode, configured, and then reset. Data sampling is performed on demand with a simple SPI command followed by a number of SPI read operations.

2.2.2.2 Data Logging

The data logging module is responsible for sending all data to the OpenLog data logger as efficiently as possible. All data is recorded in binary format in order to maximize the amount of data per second that can be saved. This format is the most space efficient but is least readable by a human. Since the binary file also lacks any structure, it is impossible to detect if any information is lost during transmission. To address this challenge, all data corresponding to a single sampling time is wrapped with a header and footer. The binary file is then decoded offline by a different script described below.

2.2.2.3 Data Decoding

All four IMUs are queried during each sampling interval. Each sensor responds with a 2-byte measurement for each of its axes. As discussed previously, the sensors have six measurement axes (three for the accelerometer and three for the gyroscope). Therefore, each measurement generates 48 bytes of data and is wrapped by a header and a footer.

When decoding the raw data, this script looks for a header and then inspects the byte exactly 48 bytes in front of the header. If this byte is not the footer, it concludes that some amount of data was lost during the transmission, discards the information, and begins looking for the next header. This procedure maintains the integrity of the data while still using space-efficient binary files.

Although this method is effective, the header and footer are nothing more than a sequence of bits, and there does exist a risk they will appear spontaneously in the data stream. Because the decoding algorithm ignores all bits between the header and the expected footer, a sequence of data bits matching the header would not affect the normal operation of the system. If the system lost information, however, and did not find the footer in the expected location, it is possible to identify a false byte when looking for a new header. Since this would also cause a mismatch at the new expected footer, however, the algorithm will ultimately ignore this false header and again begin looking for a valid one.

2.2.2.4 Timing and Interrupt Service Routine

As mentioned above, each sample interval generates 48 bytes of data, plus a 2-byte header and a 2-byte footer. Thus, 52 bytes of information must be saved at each sampling interval. Tests of the OpenLog data logger indicate that it will consistently lose data when attempting to save more than 7kB per second. In order to stay under this limit, the system can not sample faster than 134Hz. However, due to the critical nature of the data, the sampling rate is reduced by a factor of 2; the sampling rate of the system is therefore set to 70Hz. This sampling rate is also sufficiently greater than the expected frequency of the tremors that will be recorded. In general, the tremors will not exceed 7Hz, and thus a sampling rate of 70Hz will adequately prevent aliasing.

An interrupt service routine (ISR) is set up on the microcontroller with a 70Hz timer. This routine uses the IMU communication module to sample the data and then uses the data logging module to save the data to the microSD card. This ISR runs as long as the device is powered.

2.3 Data Acquisition

Once completed and tested, the sensor system was used to evaluate potential data classification algorithms. While it would have been ideal to collect data from diagnosed patients, regulatory constraints dictated that only data from healthy volunteers could be used during the initial tests of the device. Therefore, 30 healthy volunteers were educated on the detailed motion characteristics of PD and ET. While wearing the device, the volunteers were then asked to mimic these motions to the best of their ability. All the data was recorded to a microSD memory card and processed offline with the Data Decoding software module described above. This raw data was then used to test a variety of data classification algorithms. As described in Section

2.1.2, the first step to data classification is feature extraction.

2.4 Feature Extraction

Based on dialogue with medical experts at the UC Davis School of Medicine, it was determined that the most diagnostically relevant attributes of a tremor are the amplitude and the frequency of the motion. The feature extraction method used in this work is only one of many possibilities. In reality, there are an infinite number of ways to define features from the raw dataset. As mentioned in Section 2.1.2, the feature extraction method can have a significant effect on the success of a classification algorithm. Therefore, future work may include the investigation of alternative feature extraction methods.

With this in mind, a procedure was developed to extract relevant features from the raw data. Recall that the raw data has 24 individual measurements (three axes for each gyroscope/accelerometer, and four IMUs in total), each sampled at 70Hz for the entire length of a trial. The original 70Hz sampling rate used to collect the raw data causes the system's Nyquist frequency to be $F_N = 35Hz$. In order to extract the dominant frequency and amplitude from this data, each measurement of each sensor is passed through a fast Fourier transform (FFT) with 378 frequency bins. Therefore, the frequency resolution of the FFT can be computed as follows:

$$\text{Freq. Resolution} = \frac{F_N}{378} = 0.0926 Hz/Bin \tag{2.8}$$

After computing and plotting the FFT, the highest peak then corresponds to the dominant frequency of the tremor. In practice, this peak is always located at a frequency of 0Hz, because there is a steady state component to all of the time-series data. Therefore, only frequencies above 1.5Hz are considered valid.

The resulting new dominant frequency and the magnitude of the tremor at this frequency are extracted for each measurement of the raw data. This results in a set of 48 numbers (because there are 24 measurements in the raw data and each will have both a magnitude and frequency). These 48 numbers are placed in a vector which is called the feature vector in all further analysis.

A key advantage to extracting features in this manner is that the features become invariant to the sensor biases as described in the Section 2.1.1. Only the frequency and magnitude are extracted from the FFT, and the FFT operation inherently removes any DC-bias from the measurements. The bias on the measurements can therefore be ignored, making the feature vectors ready for further analysis.

2.5 Analysis of Classification Algorithms

Before performing any analysis or training with the data, it is separated into a training/validation set and a testing set. The training/validation size is selected to be 44, and the testing size is selected to be 16. In order to maintain generalizable results, none of the testing data is used or viewed in any way until the end of the analysis. All initial investigations are performed with the training data. In order to properly tune the various models, an exhaustive Leave-p-Out cross-validation is used with $p = 4$.

Leave-p-out cross-validation (LpO CV) uses p observations as a validation set and the remaining observations as a training set [4]. This is repeated with every possible combination that splits the original dataset into a validation set of p observations, requiring C_n^p trials. By using this method, many different machine learning methods are thoroughly tested on the training data while leaving the testing set untouched. Once evaluated with the test set, no new classification methods can be introduced.

2.5.1 Evaluation Criteria and Notation

While exploring various methods and techniques, each algorithm is evaluated based on the following criteria (note that cross-validation is only conducted on the training data):

1. Average classification accuracy across all iterations of a leave-4-out cross-validation.
2. Minimum classification accuracy across all iterations of a leave-4-out cross-validation.
3. Standard deviation of classification accuracy across all iterations of a leave-4-out cross-validation.

After sufficient analysis and fine-tuning was completed with the training data, the best algorithms were evaluated on the test data. When using the test data, algorithms are evaluated only based on their classification accuracy.

In order to clarify the symbols and variables used throughout the data classification process, the following notation will be adopted for the remainder of the chapter.

- $\mathbf{x_i}$ – The i^{th} feature vector in the dataset.

- $\mathbf{y_i}$ – The class label associated with the i^{th} feature vector in the dataset. $y_i = 0$ if the sample is from a volunteer mimicking PD and $y_i = 1$ if the sample is from a volunteer mimicking ET.

- $\mathbf{X}$ – The concatenation of all feature vectors into a matrix. The dimension of this matrix is $[t \times n]$ where t is the number of vectors in the dataset and n is the number of features in each vector. In this work $n = 48$.

- $\mathbf{X_{(i,:)}}$ – The i^{th} row of the matrix X.

- $\mathbf{\hat{y}}$ – The predicted class of an unknown sample.

2.5.2 Implementation and Training Results

Once the training/validation data has been isolated, many data classification algorithms are tested and evaluated based on the criteria outlined above. This section explores the effectiveness of these algorithms. The first algorithm tested is called K nearest neighbors (KNN) and is a non-parametric classification algorithm. The KNN algorithm also has a number of parameters that can have an impact on the classification accuracy. These parameters are explored and optimized. After thoroughly testing the KNN algorithm, a parametric algorithm is also developed. This parametric algorithm is based off simple classification scheme and its effectiveness is compared to that of the KNN algorithm.

2.5.2.1 Full Feature KNN

The first algorithm implemented is K nearest neighbors. KNN is a non-parametric method for data classification, which means at no point is a parametric model fit to the data [3]. Instead, training data is stored or “memorized”, and predictions are made by comparing a test sample with the stored training samples. The algorithm is simplest to understand in the case where $K = 1$. It can be summarized as follows:

Algorithm 1 KNN Algorithm

```
1: **Training Step:** Store all training samples where the set of training samples is denoted $S = \{(x_1,y_1),(x_2,y_2),\dots,(x_t,y_t)\}$
2: **Classification Step:** Given a test sample $x_0$, find the training pair $(x_i,y_i) \in S$ that minimizes the distance $d(x_0,x_i)$ between $x_0$ and $x_i$. The predicted output is then $\hat{y}(x_0) = y_i$.
```

The most common way to tune this algorithm is to adjust the value of K. When $K > 1$, the K nearest training samples are consulted using a majority vote system to determine the output prediction. A second, way to adjust this algorithm is to alter the function used for computing the distance.

KNN with L_2 distance

The first test evaluates the performance of the KNN algorithm for all values of $K \in \{1,3,5,7\}$ using the L_2 distance metric. The L_2, or Euclidean distance, is defined as:

$$d_{L_2}(\vec{p},\vec{q}) = \sqrt{||\vec{p}-\vec{q}||} \tag{2.9}$$

	K = 1	K = 3	K = 5	K = 7
Mean Accuracy	0.957	0.962	**0.974**	0.971
Min Accuracy	0.50	0.25	0.25	0.25
Std of Accuracy	0.096	0.091	0.078	0.082

Table 2.1: KNN algorithm using L_2 distance metric. Calculations are taken across all iterations of a LpO cross-validation ($p = 4$). From this result, it can be seen that a value of $K = 5$ generates the best accuracy and standard deviation when using the L_2 distance metric.

Table 2.1 suggests that performance using the L_2 distance metric was optimal with a value of $K = 5$ resulting in a 97.4% mean classification accuracy during a LpO cross-validation. While this is a good initial result, further exploration may reveal room for improvement.

KNN with L_∞ distance

In an attempt to improve the results, different distance metrics are evaluated. The second such metric is the L_∞ distance, defined as:

$$d_{L_\infty}(\vec{p},\vec{q}) = \max_i |\vec{p}_i - \vec{q}_i| \tag{2.10}$$

As before, the KNN algorithm with an L_∞ distance metric is evaluated for all values of $K \in \{1,3,5,7\}$, the results of which can be seen in Table 2.2.

	K = 1	K = 3	K = 5	K = 7
Mean Accuracy	0.932	0.967	**0.993**	0.989
Min Accuracy	0.25	0.50	0.50	0.50
Std of Accuracy	0.123	0.090	0.072	0.056

Table 2.2: KNN algorithm using L_∞ distance metric. Calculations are taken across all iterations of a LpO cross-validation ($p = 4$). From this result, it can be seen that a value of $K = 5$ generates the best accuracy the L_∞ distance metric. The standard deviation was smallest with a value of $K = 7$. This indicates that the higher K value may lead to more consistent results.

Table 2.2 reveals that performance using the L_∞ distance metric was also optimal with a value of $K = 5$. This is the same optimal K value for the L_2 distance metric. With $K = 5$ the L_∞ distance metric achieved a mean cross-validation accuracy of 99.3%. This is an outstanding result and is far superior to the L_2 results. However, it is interesting to note that the performance of the L_∞ solution is worse than the L_2 solution at $K = 1$, possibly indicating that the L_∞ solution may be less robust to a change in parameter K.

KNN with Cosine Similarity

The final distance metric used is the angular distance or cosine similarity. This is a common metric used by the machine learning community when analyzing the similarity between two text documents [21], but is also applicable in other domains. The rationale behind using it in this context is that the magnitude of a sample vector may be less relevant than its direction in an n-dimensional feature space. Therefore, one can compute how similar two sample vectors are by evaluating the angle between them. Mathematically speaking, this is represented by the following equation:

$$\angle(\vec{p},\vec{q}) = \frac{\vec{p}^T\vec{q}}{||\vec{p}||\ ||\vec{q}||} \tag{2.11}$$

However, a slight modification is required in order to use the cosine similarity as a proper distance metric. As defined in equation 2.11, the cosine similarity ranges from zero to one, approaching one if the two vectors are closer together. For the KNN algorithm to use cosine similarity as a distance, the metric should approach zero when the vectors are closer together. Therefore the d_{cos} distance metric is defined as:

$$d_{cos}(\vec{p},\vec{q}) = 1 - (\angle(\vec{p},\vec{q})) = 1 - \frac{\vec{p}\cdot\vec{q}}{||\vec{p}||\ ||\vec{q}||} \tag{2.12}$$

Thus $d_{cos}(\vec{p},\vec{q}) = 0$ when the vectors $\vec{p}$ and $\vec{q}$ are identical and $d_{cos}(\vec{p},\vec{q}) = 1$ when the vectors are perpendicular. It can also be proved that d_{cos} satisfies the triangle inequality for a proper distance metric. However, the triangle inequality is not a requirement for a distance metric to be used in the KNN algorithm.

The KNN algorithm with the d_{cos} distance metric is again evaluated for all values of $K \in \{1,3,5,7\}$. The results of this test can be seen in Table 2.3.

	K = 1	K = 3	K = 5	K = 7
Mean Accuracy	0.957	0.962	**0.974**	0.971
Min Accuracy	0.5	0.25	0.25	0.25
Std of Accuracy	0.096	0.091	0.078	0.082

Table 2.3: KNN algorithm using d_{cos} distance metric. Calculations are taken across all iterations of a LpO cross-validation ($p = 4$). From this result, it can be seen that a value of $K = 5$ generates the best accuracy and standard deviation when using the d_{cos} distance metric.

As seen in Table 2.3, the performance of the d_{cos} distance metric was identical to the performance of the L_2 distance metric (Table 2.1), with an optimal accuracy of 97.4% when $K = 5$. While this is a fairly good result, it remains inferior to the L_∞ distance metric.

2.5.3 Principal Component Analysis (PCA)

In order to investigate the feature space further, this work also seeks to determine whether all 48 features are critical when making a classification. It is possible that some of the dimensions within a sample are more important than others. In order to determine whether there is sparsity that can be exploited, a principal component analysis (PCA) is conducted.

PCA attempts to optimally reduce the n-dimensional data into a d-dimensional space. It can be conceptualized by attempting to minimize the reconstruction error of a rank-d approximation of the data. This rank-d approximation is denoted by the symbol Φ where Φ has dimension $[t \times d]$. The matrix V is of dimension $[d \times n]$ and is the matrix that maps the rank-d approximation back into the n dimensional feature space. This is represented by the following equation:

$$\min_{\Phi,V} \sum_{i=1}^{t} ||X_{(i,:)} - \hat{X}_{(i,:)}||_2^2 \quad \hat{X} = \Phi_{t\times d} V_{d\times n} \tag{2.13}$$

In order to perform the PCA each data sample is first centered by subtracting the sample mean from each feature of the sample. Then the empirical covariance matrix is computed as $C = X^T X$ where X is the matrix of centered data samples. Next, the singular value decomposition of the matrix C is computed. This is given as $C = U \Sigma U^T$. After computing the singular value decomposition, the matrix Σ will contain the singular values of the covariance matrix. By selecting the singular vectors with the d largest singular values, the n-dimensional data can be optimally mapped to a d-dimensional space.

After completing this process on the training data, the largest 15 singular values are plotted in Figure 2.9. It is observed that the top three eigenvalues are larger than the rest. Therefore the parameter d was selected as $d = 3$ for the remainder of the PCA computations. Further research should be conducted to investigate the effects of choosing different values for d.

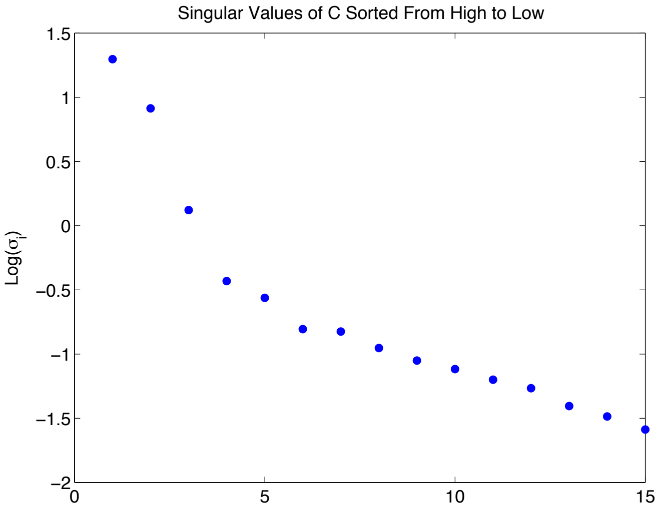


Figure 2.9: Singular values of C sorted from high to low. C is the empirical covariance matrix constructed from the training/validation data. This plot reveals that a dimension of 3 is a reasonable choice when performing the principal component analysis

Once the dimension $d = 3$ is selected, a matrix U_{reduce} is created by discarding all singular vectors except those associated with the largest three singular values. Then the feature matrix X is replaced by the matrix Φ which is defined as $\Phi = XU^T$. The dimension of Φ is therefore $[48 \times 3]$ where each of the 48 training samples has been reduced to a 3-dimensional vector.

In order to evaluate this reduced feature set, the KNN algorithm described above is implemented again with the matrix Φ as the feature matrix. As before, all three of the distance methods are implemented with varying values of K. The accuracy was then compared to the previously achieved accuracy with the full feature set. These results are displayed in Tables 2.4-2.6.

	K = 1	**K = 3**	**K = 5**	**K = 7**
Mean Accuracy w/o PCA	0.957	0.962	0.974	0.971
Mean Accuracy with PCA	0.949	**0.977**	**0.977**	0.972
Min Accuracy w/o PCA	0.5	0.25	0.25	0.25
Min Accuracy with PCA	0.250	0.250	0.500	0.500
Std of Accuracy w/o PCA	0.096	0.091	0.078	0.082
Std of Accuracy with PCA	0.105	0.077	0.072	0.079

Table 2.4: KNN algorithm using L_2 distance metric with and without PCA. Calculations are taken across all iterations of a LpO cross-validation ($p = 4$). This table reveals that the PCA reduced feature set is able to generate superior results to the full feature set. The best results were achieved with a value of $K = 3$ or $K = 5$.

	K = 1	**K = 3**	**K = 5**	**K = 7**
Mean Accuracy w/o PCA	0.932	0.967	0.993	0.989
Mean Accuracy with PCA	0.969	**0.997**	0.977	0.976
Min Accuracy w/o PCA	0.250	0.500	0.500	0.500
Min Accuracy with PCA	0.500	0.500	0.500	0.500
Std of Accuracy w/o PCA	0.123	0.090	0.043	0.056
Std of Accuracy with PCA	0.084	0.029	0.072	0.074

Table 2.5: KNN algorithm using L_∞ distance metric with and without PCA. Calculations are taken across all iterations of a LpO cross-validation ($p = 4$). This table reveals that the PCA reduced feature set is able to generate superior results to the full feature set. The best results were achieved with a value of $K = 3$.

	K = 1	**K = 3**	**K = 5**	**K = 7**
Mean Accuracy w/o PCA	0.957	0.962	0.974	0.971
Mean Accuracy with PCA	0.982	**0.996**	0.977	0.977
Min Accuracy w/o PCA	0.500	0.250	0.250	0.25
Min Accuracy with PCA	0.500	0.500	0.500	0.750
Std of Accuracy w/o PCA	0.096	0.091	0.078	0.082
Std of Accuracy with PCA	0.065	0.032	0.073	0.072

Table 2.6: KNN algorithm using d_{cos} distance metric with and without PCA. Calculations are taken across all iterations of a LpO cross-validation ($p = 4$). This table reveals that the PCA reduced feature set is able to generate superior results to the full feature set. The best results were achieved with a value of $K = 3$.

An inspection of these results indicates an overall performance increase when using the reduced feature set. This is not expected, as one would usually expect to lose information by reducing the dimensionality of the feature space through PCA. The fact that using PCA improves the accuracy indicates that there was significant noise in many of the initial features, inhibiting the ability of the KNN algorithm to correctly classify the data. It is well known that in the presence of noisy samples, increasing the parameter K will improve the performance of a KNN classifier. This is reflected by the fact that the optimal value of K is 5 for all trials involving the full feature set. After performing the PCA, however, the optimal value of K is 3 for all distance metrics tested. This indicates that the PCA analysis effectively eliminated the high-noise dimensions from the data. Once the sample noise is reduced the KNN algorithm can achieve higher performance with a reduced value for K.

2.5.4 Developing a Parametric Method

The accuracy achieved using the non-parametric KNN algorithm is sufficient, but the method is not ideal in terms of usability. In order to effectively use the KNN algorithm in practice, researchers would need to save every data sample ever recorded. The dataset could periodically be pruned to remove redundant data points and save storage space. However, this would be both time consuming and challenging. Instead, it is more advantageous to use a parametric algorithm. Recall that parametric classification algorithms learn a set of weights that can then be used to classify any unknown sample. In this section, a parametric method will be developed based on the findings from the non-parametric method above.

The design of the algorithm can be summarized by the following steps:

Algorithm 2 Parametric Classification Algorithm

```
1: Training Step: Use the training data to find two vectors. One that represents the average of all PD patients, and one that represents the average of all ET patients. These vectors are called the PD reference vector and the ET reference vector respectively.
2: Classification Step: Given an unknown sample, determine whether it is closer to the PD reference vector or the ET reference vector. The closest option is the prediction of the class for the unknown sample.
```

The training step requires finding a PD reference vector and an ET reference vector. Although non-traditional, these vectors represent the “parameters” that are learned by the classification algorithm. Recall from Section 2.1.2 that a standard parametric algorithm fits an underlying model to the training data. In this case, there is no underlying model. However, because the reference vectors are relatively short, there are still a small number of parameters that must be stored by the algorithm. Therefore, the advantages of a standard parametric algorithm are also realized by the method developed in this work.

The reference vectors can be computed in many ways. At the most basic, an arithmetic average of the raw feature vectors can be used. This is computed by first normalizing all the feature vectors (so that they have unit norm), then calculating an arithmetic average. This is done on both the full feature set and the PCA-reduced feature set developed in Section 2.5.3.

The classification step requires the comparison of the PD and ET reference vectors to an unclassified sample. As with the KNN approach, several methods can be used to determine how close one vector is to another. All three distance metrics that were tested with the KNN algorithm are also evaluated with the parametric algorithm. To remain consistent with previous tests, a LpO cross-validation with $p = 4$ is used to evaluate performance. The results of this experimentation can be seen in Table 2.7.

	Mean Accuracy	**Min Accuracy**	**Std of Accuracy**
L_2 w/o PCA	0.977	0.75	0.0720
L_2 with PCA	0.977	0.75	0.0720
L_∞ w/o PCA	0.999	0.75	0.0164
L_∞ with PCA	0.977	0.75	0.0720
d_{cos} w/o PCA	0.977	0.75	0.0720
d_{cos} with PCA	0.977	0.75	0.0720

Table 2.7: Vector average algorithm using all distance metrics with and without PCA. Calculations are taken across all iterations of a LpO cross-validation ($p = 4$). In no trial did the PCA reduced feature set produce an improved accuracy over the full feature set. The best results were achieved with the full feature set and the L_∞ distance metric.

When conducting a leave-4-out cross-validation with this new algorithm, five out of the six trials performed identically. In no trial did the PCA reduced feature set produce an improved accuracy over the full feature set. This result is surprising because it has already been established that the PCA reduced feature set has less noise. The algorithm performs no worse with the noisier full feature set; this an indication that the vector average method developed in this work is naturally robust to a noisy dataset.

The most interesting aspect of these results is the performance of the L_∞ distance metric on the full feature set. This configuration produced a mean accuracy of 0.999 - a result better than any algorithm thus far. This result also fits nicely with the insights gained when using the KNN algorithm. As seen in Tables 2.4-2.6, the L_∞ distance metric generated the best accuracy regardless of the use of PCA.

2.5.5 Accuracy on Test Data

The best algorithm before performing PCA (determined by an exhaustive cross-validation on the training data as described above) is full feature KNN with an L_∞ distance metric and $K = 5$. This achieved an average training accuracy of 0.993. After reducing the data to a 3-dimensional feature space via PCA, the best algorithms are L_∞ with $K = 3$ and d_{cos} with $K = 3$. These achieved an average training accuracy of 0.997 and 0.996 respectively. The parametric algorithm achieves an accuracy of 0.999 when used with the L_∞ distance metric. These four methods were the most promising options during the training phase, and are all evaluated on the test data. The test data consists of 16 samples that have not previously been used in any of the training procedures. The results of this classification are shown below.

	Accuracy on Test Data
KNN L_∞ (K = 5)	1.0
KNN L_∞ PCA (K = 3)	1.0
KNN d_{cos} PCA (K = 3)	1.0
Vector Average L_∞	1.0

Table 2.8: Results of the best algorithms on the test data. All algorithms achieved such high accuracy that a test set of 16 samples is not sufficient to determine which is truly the best.

Table 2.8 reveals that all the algorithms achieved test results of 100% accuracy. While this is exceptional, it limits the insights that can be gained from the test data. All the algorithms achieved such high accuracy that a test set of 16 samples was not sufficient to determine which was best. Therefore, the more revealing metric when evaluating the algorithms is the mean classification accuracy during the LpO cross-validation.

2.6 Conclusions

The experiments performed answer important questions about the classification of tremor. In general, it can be concluded from this experiment that KNN is an effective approach when attempting to classify this dataset. In the best case, a mean cross-validation accuracy of .997 was achieved. It can also be concluded that the choice of K has a moderate impact on the overall classification accuracy. In the worst case, altering K can cause a 0.02 decrease in performance. Depending on the system requirements this could be a significant margin.

It is also clear from this investigation that the L_2 distance metric is significantly worse than both L_∞ and d_{cos}. Moreover, while the L_∞ and d_{cos} distance metrics have similar performance, the L_∞ is slightly superior. When a parametric approach is developed, the L_∞ distance becomes a more clear choice as it was the only option to achieve an accuracy greater than 0.977. In fact, the parametric algorithm with an L_∞ distance metric is the most accurate of any algorithm tested, achieving an accuracy of 0.999 during a leave-4-out cross-validation.

All four methods that were implemented on the test dataset achieved 100% accuracy. However, this metric is not statistically significant because the test dataset contained only 16 samples. The more revealing metric is the accuracy achieved during cross-validation of the training data.

A principal component analysis is able to reduce the feature space to 3 dimensions without loss of performance. When the KNN algorithms are re-tested with the reduced feature space, their performance improved. This can likely be attributed to noise in the original dataset.

Overall, the analysis shows that the use of this system for the diagnosis of PD and ET may be feasible. Although the healthy volunteers cannot mimic the two diseases perfectly, they are able to produce repeatable and distinct motion patterns based on the instruction of the researchers. This work proves that the device can record the parameters of this motion with sufficient accuracy and the data classification algorithms developed show promise. While the data from real patients will look different, classifying the signals of the healthy volunteers serves as a necessary prerequisite to pursuing clinical trials. At the time of writing a clinical trial is underway to evaluate the clinical effectiveness of the device. Lastly, no volunteers reported any discomfort at all during or after the trial. This is another key validation that the system design is indeed comfortable and non-intrusive.

Chapter 3

Attitude Estimation

As advancements are made in assistive, rehabilitative, and diagnostic devices, the challenge of accurately and efficiently measuring the motion parameters of human limbs becomes increasingly relevant. In many cases (e.g., gait detection [31], or human body motion tracking [36]) the orientation or trajectory of a limb contains information necessary for the proper operation of a robotic or diagnostic device. There are many ways to track an object's motion through space. Options include optical tracking, inertial measurement, acoustic sensing, magnetic sensing, and others. Advanced camera based tracking systems have the advantage of providing a direct measure of an object's location and orientation. They require very little post-processing of data and can achieve sub-millimeter precision. However, such systems are costly to set-up and maintain, and require that the object being tracked remain within the field of view of the cameras at all times. Such systems are often large and are rarely mobile. These systems also require constant maintenance and re-calibration, which would add substantial overhead in a clinical setting. Moreover, during the evaluation of Parkinson's Disease, it is common practice to ask a patient to walk several yards away from and toward a medical professional. Tracking such a procedure with an external camera-based system would require that a large space be dedicated for this purpose. The primary goal of this chapter is to develop an algorithm that can track the attitude of the human wrist. Attitude, defined as an objects 3D orientation in space, will also become a critical parameter when designing a tremor suppression device in Chapter 4.

3.1 Background

Within the aerospace community, inertial measurement units (IMUs) are a well-researched means for measuring the raw data needed for attitude estimation. Extended Kalman filters are one common method for fusing raw sensor information into a workable attitude estimate, however several other techniques exist. Crassidis et al. [10] gives a thorough survey of several nonlinear attitude estimation techniques and covers the use of unscented Kalman filters such as the one he developed [9].

Yun et al. [35] developed a quaternion-based extended Kalman filter (EKF) for the attitude estimation of human limbs. One of his contributions was a validated dynamic model to describe the rotation rates of human limbs. However, the filter developed in Yun's work uses an additive update step which was shown by Markley [26] to be mathematically incorrect when using quaternions as an attitude parameterization. Hall et al. [18] developed an incrementally-updated quaternion-based multiplicative Kalman filter (MEKF) for use on miniature aerial vehicles (MAVs). This filter, based on the work of Markley [26], tracks the Euler angle error as its state and externally updates a reference quaternion.

Neither of these filters, however, included dynamic tracking of the gyroscope bias. It is well known that microelectromechanical (MEMs) gyroscopes are susceptible to a DC offset that varies slowly with time (see Section 2.1.1). This bias is highly temperature dependent and is not consistent between power cycles of the device. In practice, the commonly made assumption that gyroscope bias terms are estimated offline would require a user to re-calibrate the device every time it was powered. In a clinical setting, this would be impractical.

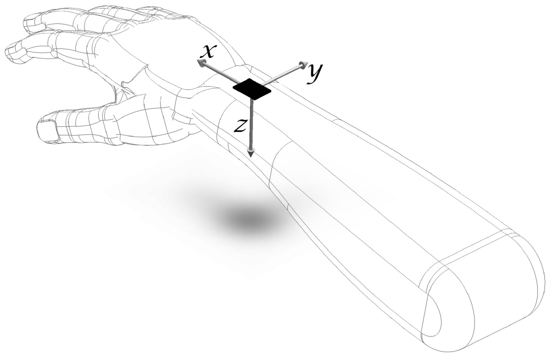


Figure 3.1: Rendering of an inertial measurement unit mounted on a human wrist. The sensor axes are labeled with the same conventions used in the aerospace community. [14]

This chapter develops a two-stage quaternion MEKF in the same style as Markley and Hall [18, 26], but with several domain-specific improvements. While the filter presented was specifically designed to track wrist attitude, it can be easily modified for other limbs. Previous filters which have tracked the attitude of human limbs have used an additive update when using quaternions. Unlike previous work, this filter also leverages the specific geometry of the human wrist to improve the performance of the accelerometer update. It incorporates assumptions about limb rotation rates validated by Yun [35] to smooth the raw gyroscope measurements. Finally, it introduces a method for the dynamic, real-time tracking of gyroscope bias.

The technical content in this chapter is based heavily on the framework of Kalman filters. Developed by Rudolf Kalman, these filters have been applied to many different engineering disciplines. Because these filters are at the core of the algorithm developed in this chapter, a basic understanding of their structure and operation is necessary. Section 3.1.1 provides this background.

3.1.1 Kalman Filters

A Kalman filter is a real time algorithm that uses noisy sensor measurements to estimate the value of an unknown variable, called the state variable. The estimation is based on a pre-defined linear process model of the physical system that is being measured, and a linear measurement model describing the relationship of the sensor measurements to the state variable. In this thesis, two different Kalman filters will be used to estimate the attitude of a human wrist.

A standard Kalman filter functions as a two step process. First, the prediction step uses information about the system dynamics, as well as the previous estimate of the state variable to predict the current value of the state variable. Second, the update step uses a sensor measurement to correct the prediction of the state variable.

In order to write this algorithm mathematically, one must first define a process model and a measurement model. To do this, the vector $\vec{x} \in \mathbf{R}^n$ is defined as the state variable of a system, and the discrete state transition matrix is defined as the matrix $\Phi \in \mathbf{R}^{n \times n}$. In the process model, the matrix Φ describes how to propagate the state variable from time step k to time step $k+1$. Thus, the dynamic process model is defined as:

$$\vec{x}_{k+1} = \Phi \vec{x}_k + \Gamma \vec{\mathrm{w}}_k \tag{3.1}$$

Where $\vec{\mathrm{w}}_k$ is a Gaussian white noise process and Γ is the cross correlation matrix of $\vec{\mathrm{w}}$. If $\vec{\mathrm{w}}$ is a zero-mean, uncorrelated process, Γ becomes an identity matrix. Based on the process model defined in 3.1, the process noise covariance matrix, Q, is defined as:

$$Q = E\{\vec{\mathrm{w}}^T\vec{\mathrm{w}}\} \tag{3.2}$$

In order to define a measurement model, it is assumed that the elements in the vector $\vec{z} \in \mathbf{R}^m$ are directly measurable by the sensors of the system. Therefore, if a matrix $H \in \mathbf{R}^{m\times n}$ can be constructed that relates $\vec{x}$ to $\vec{z}$, a measurement model is defined as:

$$\vec{z}_k = H\vec{x}_k + \vec{v}_k \tag{3.3}$$

Where $\vec{v}_k$ is a Gaussian white noise process. If the measurement or the process noise are not white, a coloring term can be added to equation 3.3. Based on this measurement equation, the measurement noise covariance matrix, R, is defined as:

$$R = E\{\vec{v}^T\vec{v}\} \tag{3.4}$$

With the models above, a Kalman filter is able to provide the optimal estimate of the state variable $\vec{x}$ given the measurements $\vec{z}$ with equations 3.5-3.9. In these equations, the matrix P is used to denote the state covariance matrix, which is a measure of the estimated accuracy of the state estimate.

Prediction Step:

$$\hat{x}_{k+1}^{-} = \Phi_k \hat{x}_k \tag{3.5}$$

$$P_{k+1}^{-} = \Phi P_k \Phi^T + \Gamma Q \Gamma^T \tag{3.6}$$

Kalman Gain:

$$K_{k+1} = P_{k+1}^{-} H_{k+1}^{T} (H_{k+1} P_{k+1}^{-} H_{k+1}^{T} + R_{k+1})^{-1} \tag{3.7}$$

Measurement Update:

$$\hat{x}_{k+1} = \hat{x}_{k+1}^{-} + K_{k+1}(z_k - H_{k+1}\hat{x}_{k+1}^{-}) \tag{3.8}$$

$$P_{k+1} = (I - K_{k+1}H_{k+1})P_{k+1}^{-} \tag{3.9}$$

As a final note, if Φ and H are non linear, they are replaced with the Jacobian matrices $\frac{\partial \Phi}{\partial x}\big|_{x=\hat{x}_k}$ and $\frac{\partial H}{\partial x}\big|_{x=\hat{x}_k}$ which must be computed at each time step. Non-linear filters in this form are called extended Kalman filters (EKF).

3.2 Attitude Representations

With a basic understanding of Kalman filters established, the focus can be shifted toward the quantity that will be estimated by the Kalman filter: attitude. An object's attitude can be thought of as its 3D orientation in space, and it is usually defined by relating an object's local coordinate frame to a global reference frame. The local frame is often called the "body" frame, while the global reference is usually referred to as the "earth" or "inertial" frame. The relationship between the body frame and the earth frame can be expressed using many different methods and nomenclatures. Each of these methods offers a unique way to "parameterize" the attitude of an object. For a detailed discussion on this topic see the work of Shuster [29]. This section will provide a brief introduction to three common forms of attitude parameterization.

3.2.1 Euler Angles

Euler angles parameterize attitude by using a length 3 vector. This vector will be denoted by

$$\text{Euler Angles} = \begin{bmatrix} \psi \\ \theta \\ \phi \end{bmatrix} \tag{3.10}$$

Each element of this vector is a scalar value representing a rotation angle about one of the body axes and together, the angles ψ, θ, and ϕ, are known as Euler angles. These rotations must be applied consecutively, and as a result, the order must be explicitly defined beforehand. Standard convention dictates the following order:

1. First, rotate ϕ radians about the body z axis.
2. Next, rotate θ radians about the body's new y axis.
3. Finally, rotate ψ radians about the body's new x axis.

When the above convention is used, the angles ψ, θ, and ϕ are given the names yaw, pitch and roll respectively. This rotation can be visualized by Fig. 3.2.

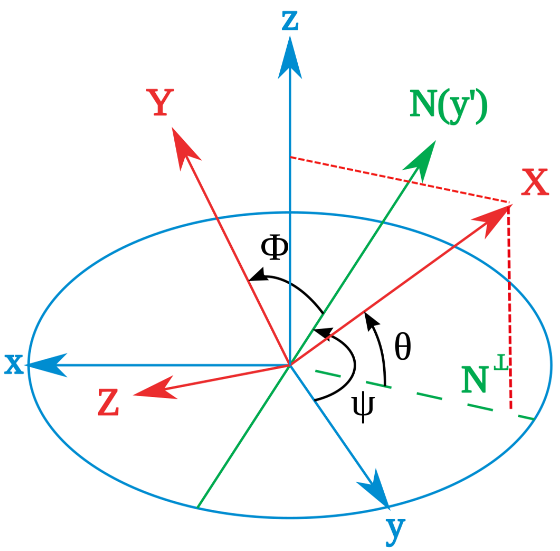


Figure 3.2: Euler Angle rotations. These angles parameterize attitude by using a length 3 vector. This method of attitude parameterization is intuitive, but has other limiting drawbacks.

A key advantage to the use of Euler angles is their intuitive nature. An engineer can very easily inspect the Euler angles for a given object and immediately glean an understanding of its orientation. A second advantage to the use of Euler angles is that they can represent an object's attitude with only three numbers. It will be seen later that other methods require more parameters.

The biggest disadvantage to the use of Euler angles is the existence of a mathematical singularity at a pitch angle of $\pi/2$. This singularity can cause algorithms using Euler angles to fail (a phenomenon known as gimbal lock [14]). A second and less catastrophic disadvantage to the use of Euler angles is the additional manipulation often required to use them in calculations. For example, to rotate an arbitrary vector in the earth frame into the body frame, the Euler angles must first be converted to a rotation matrix. This rotation matrix can then be multiplied by the earth-frame vector. Forming this rotation matrix, however, requires the computation of transcendental functions that are computationally expensive. Because of this, some engineers choose to forgo the use of Euler angles entirely, opting to use a method called the direction cosine matrix instead.

3.2.2 Direction Cosine Matrix

The direction cosine matrix (DCM) parameterizes attitude by the use of a 3×3 orthonormal rotation matrix. This matrix relates the basis vectors of the earth coordinate frame to the basis vectors of the body frame. Specifically, each (i, j) element of the DCM is the cosine of the angle between the i^{th} basis of the global frame to the j^{th} basis of the local frame. In the context of Euler angles, the DCM can be written as:

$$DCM = \begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{3.11}$$

As long as the DCM remains orthonormal, it is a valid rotation matrix and can be used directly to rotate arbitrary vectors from one coordinate frame to another. Moreover, propagating a DCM from one time step to another is a computationally simple integration. This is a distinct advantage over the use of Euler angles. Another advantage is that the DCM does not have any mathematical singularities. This makes it an appealing choice when avoiding gimbal lock is critical.

There are three key disadvantages to the use of the DCM. The first is that the DCM is not a numerically intuitive representation of attitude. It would be very difficult for an engineer to inspect a DCM and have an immediate intuition of the attitude of the object. The second disadvantage is that the DCM requires 9 values to represent attitude (the matrix is 3×3). This is inherently inefficient and can hinder the performance of attitude estimation algorithms when computing resources are limited. Finally, the DCM has a propensity to drift from orthonormality when the attitude estimate is propagated. When this happens, extra computation is required to return the DCM to an orthonormal matrix.

3.2.3 Quaternions

A quaternion is a mathematical construct that extends the concept of imaginary numbers to 4-dimensional space. In the context of attitude, an attitude quaternion, $\bar{q}$ is given by:

$$\bar{q} = \left[\begin{array}{c} q_0 \\ \hline q_1 \\ q_2 \\ q_3 \end{array} \right] \tag{3.12}$$

Where q_0 is a scalar, $[q_1, q_2, q_3]^T$ is a vector in imaginary space corresponding to the axis of rotation between body and earth coordinates, and $||\bar{q}|| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$ is identically 1. There are a number of unique mathematical properties that must be understood when using quaternions as a representation of attitude, and these will be discussed in Section 3.3.

There are three major advantages to using quaternions as a representation of attitude. The first is that quaternions contain no mathematical singularities and therefore do not suffer from gimbal lock. The second is that quaternions only need four values in order to represent an objects attitude. This is slightly worse than the Euler angle representation which requires three values but is a marked improvement over the DCM. The final advantage is that an attitude quaternion can easily be used to rotate any arbitrary vector form one reference frame to another. As will be seen, this is very beneficial when the attitude estimate needs to be used for further computation.

The only significant disadvantage to the use of quaternions as an attitude parameterization is the fact that they are not numerically intuitive. As with the DCM, it is nearly impossible to visually inspect the components of an attitude quaternion and derive an intuitive understanding of the corresponding 3D orientation in space. For the current application, this disadvantage is not considered detrimental. Therefore, attitude quaternions are the parameterization that will be used for the proposed attitude estimation algorithm.

3.3 Quaternion Arithmetic

As mentioned above, quaternions will be used as the parameterization of attitude in this chapter. Therefore, it is critical to understand the mathematical properties that govern their use. This section presents a few of these mathematical relationships, beginning with the construction of a quaternion. Specifically, if an object's body coordinates must be rotated by an angle θ about an axis $\vec{s}$, in order to rotate them into earth coordinates the attitude quaternion can be constructed by:

$$\bar{q} = \begin{bmatrix} cos\left(\frac{\theta}{2}\right) \\ s_x sin\left(\frac{\theta}{2}\right) \\ s_y sin\left(\frac{\theta}{2}\right) \\ s_z sin\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{3.13}$$

In general, any arbitrary quaternion has both an inverse, $\bar{q}^{-1}$, and a conjugate, $\bar{q}^{*}$. These quantities can by computed as:

$$\bar{q}^{*} = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} \tag{3.14}$$

$$\bar{q}^{-1} = \frac{\bar{q}^{*}}{||\bar{q}||} \tag{3.15}$$

However, because an attitude quaternion is defined to have a unit norm, its inverse is equivalent to its conjugate. Therefore,

$$\bar{q}^{-1} = \bar{q}^{*} = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} \tag{3.16}$$

The next key property of quaternions is the multiplication operator $\otimes$. This operator is used to rotate a quaternion $\bar{q}$ by the rotation given by quaternion $\bar{p}$. It is defined as:

$$\bar{p} \otimes \bar{q} = \begin{bmatrix} p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3 \\ p_0q_1 + p_1q_0 - p_2q_3 + p_3q_2 \\ p_0q_2 + p_1q_3 + p_2q_0 - p_3q_1 \\ p_0q_3 - p_1q_2 + p_2q_1 + p_3q_0 \end{bmatrix} \tag{3.17}$$

It is important to note that if $\bar{p}$ and $\bar{q}$ are both unit norm, the result of $\bar{p} \otimes \bar{q}$ will also be unit norm.

Finally, given a quaternion $\bar{q}_{a \to b}$ that represents the rotation from coordinate frame a to b, any arbitrary vector $\vec{x}_a$ in the Cartesian coordinates of frame a can be expressed in the Cartesian coordinates of frame b as follows:

$$\vec{x}_b = \bar{q}_{a \to b} \otimes \begin{bmatrix} 0 \\ \hline \vec{x}_a \end{bmatrix} \otimes \bar{q}^*_{a \to b} \tag{3.18}$$

3.4 Problem Geometry

With the Kalman filter equations and quaternion arithmetic equations defined, a sufficient technical foundation has been established to begin the development of an attitude estimation algorithm. The first step in the development of this algorithm is to formulate a clear definition of the problem geometry and notation that will be used.

The filter developed in this thesis uses information from an IMU mounted to the top of a users wrist to estimate the attitude of the wrist. Fig. 3.1 shows this setup. The axes of the IMU are consistent with standards typically used in the aerospace industry [14]. The x axis points toward the hand, the y axis points out the right side of the wrist, and the z axis points down. Some of the calculations in this chapter rely on a vector $\vec{d}$ which originates at the center of the elbow joint and terminates on the surface of the wrist where the IMU is mounted. This is shown in Fig. 3.3.

The filter was designed to operate in conjunction with the MPU6000, a 6 degree-of-freedom IMU equipped with a 3-axis gyroscope and a 3-axis accelerometer. Therefore, the system measurements (fed as inputs to the filter) are the raw outputs of the accelerometer and gyroscope. The accelerometer measurements have units of g's and are given by:

$$\vec{a}_m = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{3.19}$$

The gyroscope measurements have units of radians per second and are given by:

$$\vec{\omega}_m = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3.20}$$

Lastly, this chapter uses both quaternions and vectors in its analysis. To avoid confusion, all vector quantities will be denoted by the vector symbol, as in $\vec{d}$, and all quaternions will be denoted with a bar, as in $\bar{q}$. All vectors are assumed to be column vectors. Finally, any quantity that is directly measurable will be denoted with a subscript, as in $\vec{a}_m$.

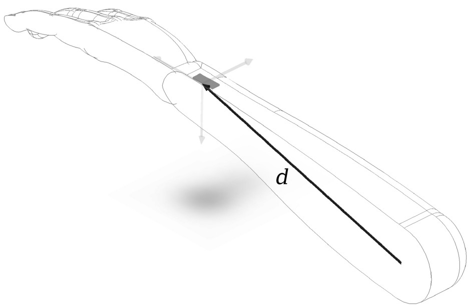


Figure 3.3: Lateral cross-section of a human forearm, wrist, and hand. The inertial measurement unit and its axes are mounted identically to Fig 3.1. The vector $\vec{d}$ points from the center of the elbow joint to the surface of the wrist where the IMU is mounted.

3.5 Design of a Two-Stage Multiplicative Kalman Filter

The filter developed in this chapter operates as two consecutive Kalman filters. This is illustrated by the block diagram in Fig. 3.4. The first stage is a standard Kalman filter which smooths the raw gyroscope measurements. These values are then passed to a second stage filter which propagates the current attitude estimate by forward integrating the smoothed rotation rates. The second stage filter then incorporates the accelerometer measurements to correct for error in the open loop integration. A convenient byproduct of the 2^{nd} stage filter is an indirect measurement of the change in gyroscope bias. These bias values are tracked by a first order forward integration and are used to correct the raw gyroscope measurements. The remainder of this section is devoted to a detailed mathematical description of the stage 1 and stage 2 filters.

It is important to remember that the filter is designed to operate with only accelerometer and gyroscope information. With only these two input sources, there are no aiding sensors to provide an accurate correction for the yaw estimate. Several methods exist to manage the accumulation in drift along the yaw axis. One such implementation is to model yaw as a zero-mean spring-damper system. This approach would cause the yaw to slowly return to zero, but would allow any transient movements detected by the gyroscopes to be observed in the attitude estimate. However, for the specific application in this thesis, the yaw axis is irrelevant. Therefore, the yaw estimate is simply reset to zero at every time step and discarded.

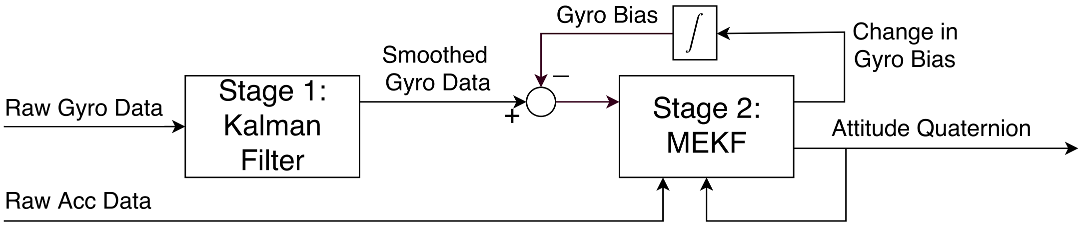


Figure 3.4: Block diagram of filter architecture. The filter is designed in two discrete stages. The first stage smooths the gyroscope measurements using an appropriate process model. The second stage incorporates the accelerometer measurements and returns the attitude estimate as a quaternion.

3.5.1 Stage 1: Kalman Filter

The stage 1 Kalman filter is a discrete-time filter that smooths the gyroscope measurements. It fits the format of a standard Kalman filter presented in Section 3.1.1 and contains an internal state of

$$\vec{x} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3.21}$$

Where $[\omega_x, \omega_y, \omega_z]$ are estimates of the body coordinate rotation rates. It has been shown [35] that a first-order Gauss-Markov process is a simple yet sufficient dynamic model for the rotation rates of human limbs. The continuous time process model for the rotation rate of a limb is given by

$$\frac{d\vec{x}}{dt} = -\frac{\vec{x}}{\tau} + \frac{d\vec{\mathrm{w}}}{\tau} \tag{3.22}$$

Where τ is a tunable constant that may or may not differ between axes, and $\vec{\mathrm{w}}$ is a zero-mean, white-noise Gaussian process. Based on (3.22), the equivalent discrete-time transition matrix, with a sampling rate of T_s, can be written as

$$\Phi_1 = e^{-\frac{T_s}{\tau}} \tag{3.23}$$

Where the subscript on the variable Φ_1 indicates that this is the discrete transition matrix for the stage 1 filter. In order to use the standard Kalman filter equations, it is also necessary to derive a process noise covariance matrix Q. This is done by first inspecting the continuous process model given in equation (3.22). Based on this equation, the variance of each state is governed by the following differential equation:

$$\dot{\sigma}^2 = -\frac{2}{\tau}\sigma^2 + \left(\frac{1}{\tau}\right)^2 \tag{3.24}$$

And the discrete form of (3.24) can be written as

$$\sigma_{k+1}^2 = e^{-\frac{2T_s}{\tau}}\sigma_k^2 + Q_1 \tag{3.25}$$

Where Q_1 is the process noise covariance matrix for the stage 1 filter. The value of Q_1 can be analytically determined by solving the differential equation given by (3.24) and comparing like-terms as follows:

$$\sigma^2 = Ce^{-\frac{2t}{\tau}} + \frac{1}{2\tau} \tag{3.26}$$

Where C is an unknown constant. It is clear by inspection that the first terms in (3.25) and (3.26) match. However, (3.26) has the extra term $\frac{1}{2\tau}$ which must correspond to Q_1 in (3.25). Therefore, the value of the process noise covariance matrix, Q_1 must be

$$Q_1 = \begin{bmatrix} \frac{1}{2\tau} & 0 & 0 \\ 0 & \frac{1}{2\tau} & 0 \\ 0 & 0 & \frac{1}{2\tau} \end{bmatrix} \tag{3.27}$$

With the above equations, the prediction step for the stage 1 filter is given by

$$\hat{\vec{x}}_{k+1}^{-} = \Phi_1 \hat{\vec{x}}_k \tag{3.28}$$

$$P_{k+1}^{-} = \Phi_1 P_k \Phi_1^T + Q_1 \tag{3.29}$$

Where P is the state covariance matrix. It should be noted that in this implementation, a first order approximation of the state transition matrix is used (either a matrix exponential or higher order approximations could also be used). Therefore,

$$\Phi_1 = \begin{bmatrix} 1-\frac{T_s}{\tau} & 0 & 0 \\ 0 & 1-\frac{T_s}{\tau} & 0 \\ 0 & 0 & 1-\frac{T_s}{\tau} \end{bmatrix} \tag{3.30}$$

Once the prediction step is complete, a measurement model is needed in order to construct the update step. As described in Section 3.4, the IMU is able to directly measure the body rotation rates. Therefore, the measurement vector, z, that will be used in the stage 1 filter is given by

$$\vec{z} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{3.31}$$

Because these measurements correspond identically to the states of the stage 1 filter, the measurement model H_1 is given by a 3×3 identity matrix. Finally, the measurement noise covariance matrix, R, is determined based on the variance of the specific sensors used. In the implementation of this filter, an Invensense MPU6000 IMU is used to collect data. With this sensor, the variance on the gyroscope axes was experimentally determined to be

$\sigma^2_{gryo} = 4.68 \times 10^{-5}$ rad$^2/s^2$. Therefore,

$$R_1 = \begin{bmatrix} \sigma^2_{gryo} & 0 & 0 \\ 0 & \sigma^2_{gryo} & 0 \\ 0 & 0 & \sigma^2_{gryo} \end{bmatrix} \tag{3.32}$$

With the above equations, the update step of the stage 1 filter is given by the following equations:

$$K_{k+1} = P^-_{k+1} H_1^T (H_1 P^-_{k+1} H_1^T + R_1)^{-1} \tag{3.33}$$

$$\hat{\vec{x}}_{k+1} = \hat{\vec{x}}^-_{k+1} + K_{k+1}(\vec{z}_k - H_1 \hat{\vec{x}}^-_{k+1}) \tag{3.34}$$

$$P_{k+1} = (I - K_{k+1} H_1) P^-_{k+1} \tag{3.35}$$

Where K is the Kalman gain.

The states of this Kalman filter (which are the smoothed rotation rates), as well as the raw accelerometer measurements, are then passed as inputs to the multiplicative Kalman filter. At this point, however, the rotation rates have not been corrected for the DC bias that corrupts MEMs gyroscopes. Before being passed to the stage 2 filter, the smoothed rotation rates must be corrected by subtracting the gyroscope bias. As is discussed further in Section 3.5.2.3, the gyroscope bias is calculated as a byproduct of the stage 2 filter and fed back to the stage 1 filter. Therefore, at the first time step of operation, the bias terms are set to zero. During all future time steps, the bias values are taken from the output of the stage 2 filter. To simplify notation,

$\hat{\omega}$ is defined as the smoothed and corrected gyroscope measurements.

$$\hat{\omega} = \begin{bmatrix} \hat{\omega}_x \\ \hat{\omega}_y \\ \hat{\omega}_z \end{bmatrix} = \begin{bmatrix} w_x - b_x \\ w_y - b_y \\ w_z - b_z \end{bmatrix} \tag{3.36}$$

where $[b_x, b_y, b_z]$ are the gyroscope bias terms.

3.5.2 Stage 2: Multiplicative Kalman Filter

The stage 2 filter uses the smoothed and unbiased estimates of rotation rate from the stage 1 filter in order to generate an attitude estimate. To do so, it leverages a multiplicative Kalman filter which is a slight variation on the standard Kalman filter described in Section 3.1.1. The major difference between a standard and a multiplicative Kalman filter lies within the algorithm used to compute the update step. In a standard filter, the update step uses an additive equation when computing the innovation. When using quaternions as a representation of attitude, however, Markley [26] showed that an additive update is mathematically incorrect. This work follows the suggestions of Markley, and uses a multiplicative method for computing the innovation at each time step. This section will begin by setting up the framework for the stage two filter. It will then define the prediction step as well as the multiplicative update step.

3.5.2.1 Filter Setup

The multiplicative Kalman filter devopled in this work is based on the works of [18, 26] but contains significant modifications. To begin the analysis, the filter states are defined as

$$\vec{x} = \begin{bmatrix} \tilde{e}_x \\ \tilde{e}_y \\ \tilde{e}_z \end{bmatrix} \tag{3.37}$$

Where $\tilde{e}$ is an attitude error vector. Conceptually, $\tilde{e}$ is an estimate of the error between the current attitude estimate and the true attitude. At the end of every time-step, the state vector is used to correct the current estimate. It is assumed that the error is completely corrected and there is no error at the beginning if each time step. Therefore, the *a priori* state prediction becomes

$$\vec{x}_{k+1}^{-} = 0 \tag{3.38}$$

However because the attitude quaternion is being maintained externally, the smoothed and adjusted rotation rates from the stage 1 Kalman filter must be used to manually propagate it forward. The state transition matrix (for both the vector $\vec{x}$ and the attitude quaternion) is given by

$$\Phi_2 = [\hat{\omega}_\times] = \begin{bmatrix} 0 & -\hat{\omega}_z & \hat{\omega}_y \\ \hat{\omega}_z & 0 & -\hat{\omega}_x \\ -\hat{\omega}_y & \hat{\omega}_x & 0 \end{bmatrix} \tag{3.39}$$

and the attitude quaternion is propagated forward with the equation

$$\bar{q}^-_{k+1} = \bar{q}_k + \frac{T_s}{2}\left(\begin{bmatrix} 0 \\ \hat{\bar{\omega}} \end{bmatrix} \otimes \bar{q}_k \right) \tag{3.40}$$

As explained by Hall et al. [18], this computation is a discrete approximation and has the potential to result in a quaternion with non-unity magnitude. Therefore $\bar{q}^-_{k+1}$ must be re-normalized before continuing.

$$\bar{q}^-_{k+1} = \frac{\bar{q}^-_{k+1}}{||\bar{q}^-_{k+1}||} \tag{3.41}$$

The standard Kalman filter equation applies to compute the *a priori* prediction for the covariance matrix. Therefore, P^-_{k+1} is given by

$$P^-_{k+1} = \Phi_2 P_k \Phi_2^T + Q_2 \tag{3.42}$$

Where Q_2 was manually tuned to

$$Q_2 = \begin{bmatrix} 0.014 & 0 & 0 \\ 0 & 0.014 & 0 \\ 0 & 0 & 0.014 \end{bmatrix} \tag{3.43}$$

This tuning was performed through trial and error within the simulation described in Section 3.6. The values above were selected because they worked relatively well, however further optimization could improve the results of the filter.

At this point, the second stage filter still fits the framework of the standard Kalman filter. The next step, incorporating the accelerometer measurements, will require deviations. The measurement update is performed by using the accelerometer readings to compute an attitude error vector and then rotating the current attitude estimate, $\bar{q}_{k+1}^{-}$, based on this error. The error vector is constructed by comparing the gravity vector measured by the accelerometers with the gravity vector determined by the current attitude estimate. Therefore, the accelerometer readings are critical in this step of the procedure.

In general, the accelerometers measure the inertial accelerations on the sensor minus gravity (see equation 2.2). If the limb being measured were completely at rest, the accelerometers would only record gravity and the measurements could be used without adjustment, as has been done in previous literature [18, 35]. However, in a dynamic environment, the presence of inertial accelerations makes it difficult to determine the direction of gravity based on accelerometer readings. This is mitigated in two ways: incorporating geometric knowledge of the human wrist, and dynamically varying the measurement noise covariance matrix R.

Incorporating wrist geometry

The geometry of the wrist, combined with the body rotation rates, can be used to predict the inertial accelerations on each accelerometer axis. The simplifying assumption is made that the human forearm is fixed at the elbow and that the IMU is mounted on the back of the wrist. Given this assumption, the Newton-Euler equations for rigid-body motion relate the inertial accelerations at the surface of the wrist to the rotation rates of the forearm as follows:

$$\hat{\vec{a}}_{inertial} = \frac{[\hat{\omega}_\times][\hat{\omega}_\times]d + [\dot{\hat{\omega}}_\times]d}{g} \tag{3.44}$$

Where $\vec{d}$ is the vector from the elbow to the point on the wrist where the IMU is mounted (as shown in Fig. 3.3), and $g = 9.8$ is the acceleration of gravity. The vector d can be measured exactly for a specific user, or approximate values can be used. The term $[\dot{\hat{\omega}}_\times]$ can be approximated using the first order Gauss-Markov model for the rotation rate of human limbs given by (3.22), or by computing the numerical derivative of $\hat{\omega}$ over a single time-step. The numerical derivative, however, is very susceptible to noise in the sensor measurements; therefore the Guass-Markov model is the preferred method.

The accelerometer readings can now be used more effectively to estimate the measured gravity vector, $\vec{g}_m$, as follows:

$$\vec{g}_m = \hat{\vec{a}}_{inertial} - \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{3.45}$$

Where a_x, a_y, and a_z are the raw accelerometer measurements. This adjustment is effective at creating a more accurate measurement of gravity, but is not exact. In reality, the forearm is not fixed at the elbow. As a result, $\vec{g}_m$ may still be influenced by inertial accelerations that do not fit the model. To mitigate the effect of such errors, the measurement noise covariance matrix R_2 can be dynamically varied.

Dynamic R Matrix

Theoretically, the adjusted gravity vector should have unit norm. However, noise and unmodeled inertial accelerations could cause the norm to be greater or less than one. The further the norm of $\vec{g}_m$ is from one, the less confident the filter should be in the validity of the measurement. This lack of confidence can be incorporated into the measurement noise covariance matrix as follows.

$$R_2 = \begin{bmatrix} \sigma_{acc}^2 & 0 & 0 \\ 0 & \sigma_{acc}^2 & 0 \\ 0 & 0 & \sigma_{acc}^2 \end{bmatrix} \left(1 + \left|1 - ||\vec{g}_m||\right|\right) \tag{3.46}$$

Where σ_{acc}^2 is the experimentally determined variance on the accelerometer readings. In the case of the MPU6000 IMU this value was found to be $\sigma_{acc}^2 = 4.15 \times 10^{-5}$ g^2. The matrix R_2 is used when computing the Kalman Gain. By increasing R_2, the Kalman gain will decrease, and the filter will reduce the weight placed on the accelerometer measurements.

3.5.2.2 Measurement Update

With the enhancements outlined above, the adjusted accelerometer readings can now be used in the multiplicative update step. The measurement update uses the vector $\vec{g}_m$ to construct an attitude error vector. The attitude error is then used to correct the current attitude estimate and to generate a measurement of the gyroscope bias. To construct the error vector, $\vec{g}_m$ must be compared to the gravity vector as determined by the current attitude estimate, denoted $\hat{g}$. This can be found by rotating the inertial z axis into body coordinates:

$$\hat{g} = \bar{q}_{k+1}^{-} \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \otimes (\bar{q}_{k+1}^{-})^{-1} \tag{3.47}$$

With $\vec{g}_m$ and $\hat{g}$ computed, the task becomes finding the quaternion that rotates the current attitude estimate such that $\vec{g}_m = \hat{g}$. This process was explained in detail in [18] and is implemented by first computing the axis and angle of this rotation. The axis and angle of the rotation, expressed in the body frame, can be found as follows.

$$E_{axis} = \frac{\vec{g}_m}{||\vec{g}_m||} \times \hat{g} \tag{3.48}$$

$$E_{angle} = \cos^{-1}\left(\frac{\vec{g}_m}{||\vec{g}_m||} \cdot \hat{g}\right) \tag{3.49}$$

Therefore the error vector $\tilde{e}$ is given by:

$$\tilde{e} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = E_{angle} E_{axis} \tag{3.50}$$

This vector, $\tilde{e}$, is the attitude error vector expressed in the Euler angles of the body coordinate frame. Recall that the state of the stage 2 filter has an identical definition: the error between the current estimate and the true attitude. Therefore, $\tilde{e}$ serves as the measurement vector $\vec{z}$ during the MEKF computation. The Kalman gain K can then be computed by the following (where H_2 is a 3×3 identity matrix):

$$K_{k+1} = P_{k+1}^{-} H_2^T (H_2 P_{k+1}^{-} H_2^T + R_2)^{-1} \tag{3.51}$$

The following measurement update is then used to compute the *a posteriori* state vector:

$$\hat{\vec{x}}_{k+1} = \hat{\vec{x}}_{k+1}^{-} + K_{k+1}(\tilde{e} - H_2 \hat{\vec{x}}_{k+1}^{-}) \tag{3.52}$$

However, by recalling that $\vec{x}_{k+1}^{-} = 0$, the above equation can be simplified to

$$\hat{\vec{x}}_{k+1} = K_{k+1} \tilde{e} \tag{3.53}$$

The *a posteriori* estimate of the state covariance matrix is then

$$P_{k+1} = (I - K_{k+1}H_2)P_{k+1}^{-} \tag{3.54}$$

Equation (3.13) can then be used directly with $\hat{\vec{x}}_{k+1}$ to construct the quaternion rotation, $\delta\bar{q}$, that will correct the attitude estimate. When $\bar{q}_{k+1}^{-}$ is rotated by $\delta\bar{q}$, the final attitude estimate becomes

$$\bar{q}_{k+1}^{+} = \delta\bar{q} \otimes \bar{q}_{k+1}^{-} \tag{3.55}$$

3.5.2.3 Gyroscope Bias Measurement

The above computations generate the final attitude estimate; however additional information can be gleaned from the attitude error vector. This error is reflected in the *a posteriori* state, $\hat{\vec{x}}_{k+1}$ and can provide an indirect measurement of the gyroscope bias.

In an ideal setting, void of noise or inertial acceleration, the attitude error vector, $\hat{\vec{x}}_{k+1}$, would only be non-zero as a result of gyroscope bias. Therefore, a measurement of the incremental change in gyroscope bias can be computed by dividing $\hat{\vec{x}}_{k+1}$ by T_s. Adding this incremental change in bias to the previous estimate with the following first order forward integration provides the measurement allows the filter to dynamically track the gyroscope bias.

$$\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}_{k+1} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}_{k} + \frac{\alpha \hat{\vec{x}}_{k+1}}{T_s} \tag{3.56}$$

Where α is a tunable constant. In this way, the stage 2 filter is able to output a dynamic measurement of the gyroscope bias. This bias is fed back to the output of the stage 1 filter and used to correct the rotation rates in equation 3.36.

3.6 Simulation Results

The two-stage filter described above is tested in a simulated environment. The gyroscope measurements are corrupted with 5.73 degrees per second DC bias (0.1 radians per second). All sensor measurements were corrupted with white, Gaussian noise representative of the MPU6000 IMU ($\sigma^2_{acc} = 4.15 \times 10^{-5}\,\mathrm{g}^2$ and $\sigma^2_{gryo} = 4.68 \times 10^{-5}\,\mathrm{rad}^2/s^2$). The IMU is simulated at rest for 5 seconds and then with a 1 Hertz sine wave on the yaw, pitch and roll axes simultaneously. The errors seen in the initial few seconds of Fig. 3.5 are due to the injected sensor bias. Once these biases are determined by the algorithm, the error is corrected.

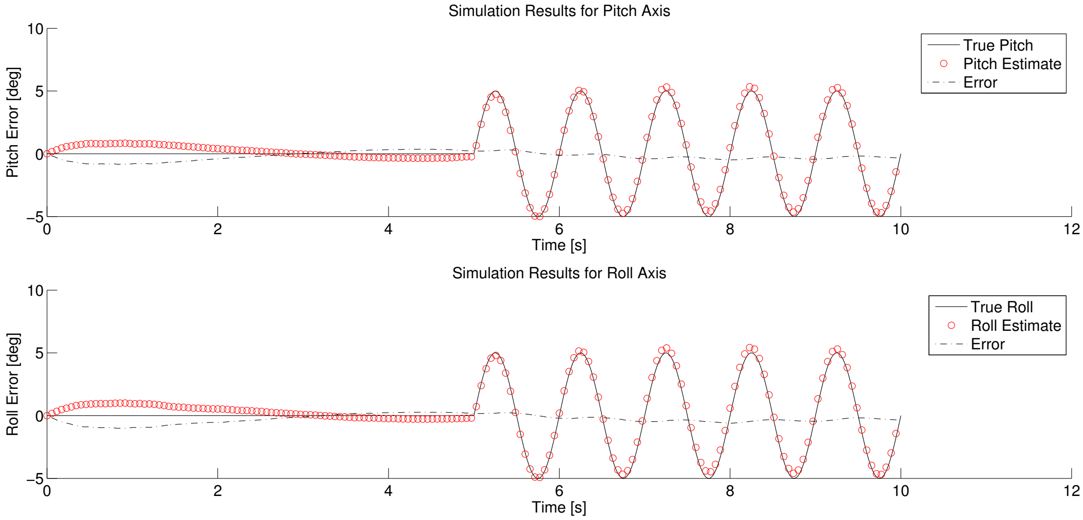


Figure 3.5: Simulation results of the two-stage Kalman filter. The errors seen in the initial few seconds are due to the injected sensor bias. Once these biases are determined by the algorithm, the error is corrected.

As can be seen in Fig. 3.5, the filter performs very well and can quickly adjust for the DC gyroscope bias. In both axes, the effects of the DC gyroscope bias are corrected after approximately 3 seconds. The RMS errors were 0.395° and 0.438° for the pitch and roll axes respectively, and with a total amplitude of 10°, these error values are sufficiently accurate. The results also indicate that the manually tuned value of Q_2 (equation 3.43) is reasonable. While further optimizations could likely be made, the simulation results are adequate to proceed with experimental validation.

3.7 Experimental Validation

To validate the algorithm experimentally, it is tested with a mechanical rig that can mimic Parkinsonian tremor. The test rig is designed to oscillate in a manner similar to the tremor caused by PD. A detailed description of this rig is presented in Section 4.2. In this test, it was set to oscillate at approximately 5Hz. A single IMU is placed on the rig and the software developed in Section 2.2.2 is used to record the motion data. The data is then passed through the two-stage Kalman filter. In order to validate the accuracy of the filter, a magnetic encoder is mounted to the oscillating portion of the test rig. This encoder provides the true angle of the rig as a reference measurement. Figures 3.6-3.7 show the performance of the algorithm during this test.

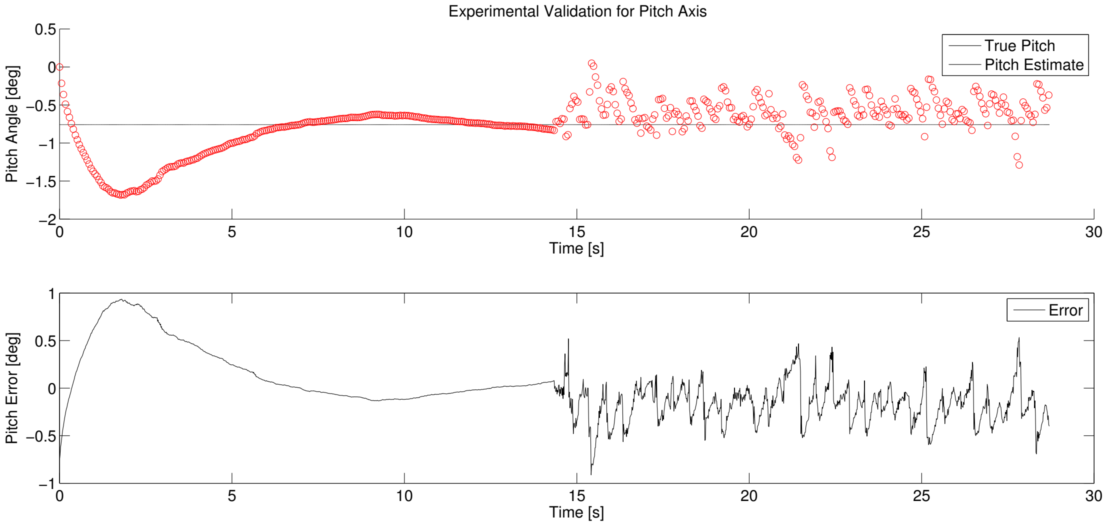


Figure 3.6: Experimental validation of the two-stage Kalman filter for the pitch axis. A mechanical test rig was used for this validation. The maximum error was 0.936° and the RMS error was 0.322°. These values represent high accuracy given the dynamic motion in the test rig.

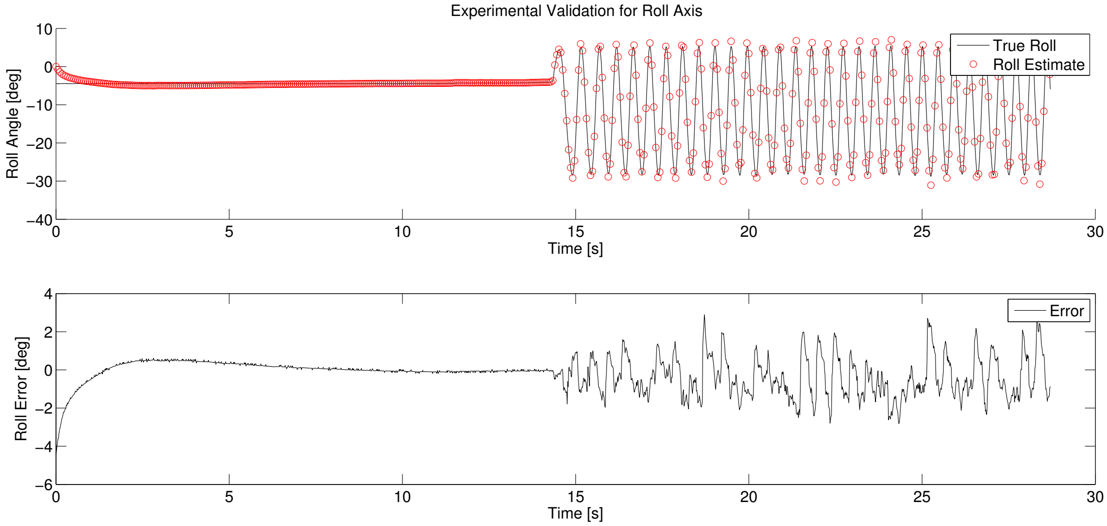


Figure 3.7: Experimental validation of the two-stage Kalman filter for the roll axis. A mechanical test rig was used for this validation. The maximum error was 3.336° and the RMS error was 0.911°. These values represent high accuracy given the dynamic motion in the test rig.

It is clear from Figures 3.6-3.7 that there is considerably more noise when the mechanical test rig is used than when the simulation is used. This is because the simulation only accounted for the Gaussian sensor noise. The use of the test rig also introduces many other sources of noise, such as vibrations and unmodeled dynamics. Even in the presence of these disturbances, however, the algorithm performed very well. The maximum error at any time instant on either axis is 3.336°.

During a 5Hz tremor the true motion is highly dynamic. As can be seen in Fig. 3.8 the rotation rates reach 260 degrees per second. Given this, the maximum error of 3.336° is considered quite low. The root mean square (RMS) errors are 0.321° and 0.911° for the pitch and roll axes respectively. For the purposes of this thesis, this is more than sufficient.

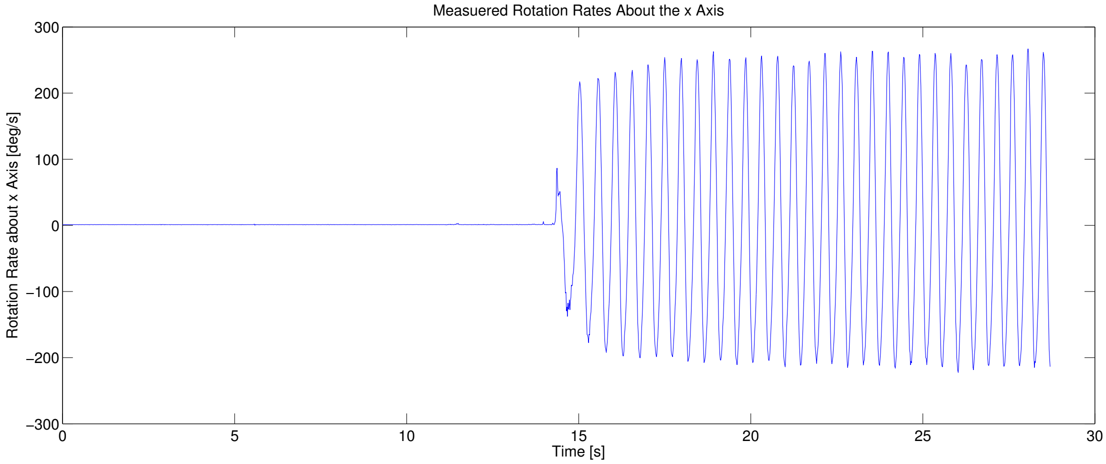


Figure 3.8: Measured rotation rates about the roll axis during a test on the mechanical test rig. Despite being highly dynamic, the attitude estimation algorithm must be able to track wrist attitude during this motion.

Another key insight can be made by analyzing the values of the Kalman gain matrices, denoted K_1 and K_2 for the first and second stage filters respectively. Both of these matrices are of dimension 3×3, yet an initial inspection indicated that the diagonal elements were six orders of magnitude larger than the off-diagonal elements. This observation indicates that the diagonal elements dominate the gain calculations. As such, the computational complexity of the filter could likely be reduced without a loss in accuracy by setting the off diagonal entries to zero. Figures 3.9 and 3.10 show how the diagonal elements of both gain matrices change with time.

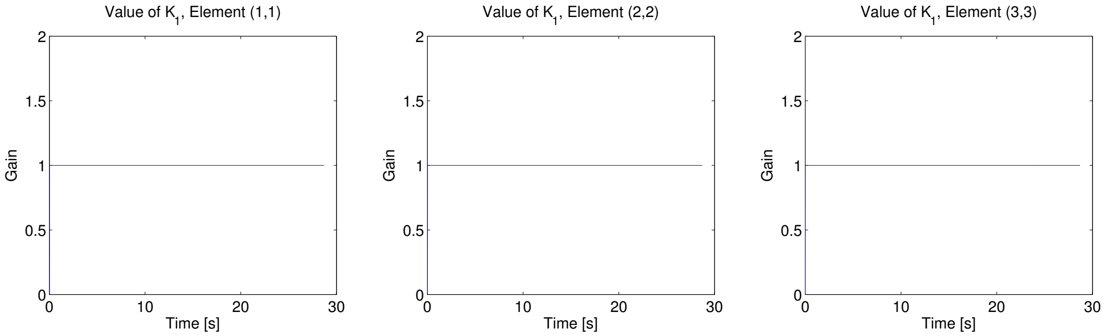


Figure 3.9: Steady state Kalman gains from the stage 1 filter. Only diagonal elements are plotted, as the off diagonals were 6 orders of magnitude smaller and deemed to have minimal impact on the filter. These gains reach steady state very quickly. Therefore, the computational complexity of a Kalman filter may not be necessary and could potentially be replaced with a static filter.

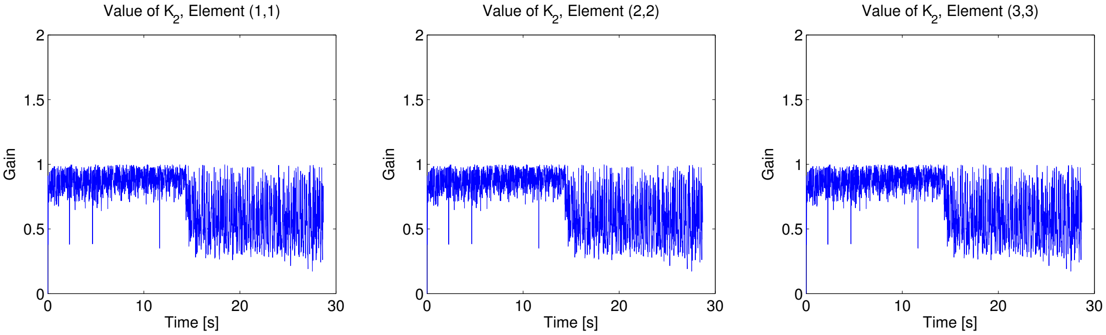


Figure 3.10: Steady state Kalman gains from the stage 2 filter. Only diagonal elements are plotted, as the off diagonals were 6 orders of magnitude smaller and deemed to have minimal impact on the filter. These gains do not reach steady state, indicating that a static filter would likely be insufficient.

As can be seen in Fig. 3.9, the stage 1 gain matrix converges to a constant value very quickly. Therefore, the computational overhead of a Kalman filter may not be necessary. Instead, a simple static observer could be constructed using the steady-state gains. In the case of the second stage filter, however, the gain matrix is not static, indicating that a static filter would likely be insufficient. Future work should investigate the use of a static stage 1 filter in conjunction with the current stage 2 filter.

3.8 Conclusions

This chapter presents a two-stage Kalman filter for the attitude estimation of a human wrist. It represents an overall improvement from previous work. The multiplicative nature of the update step used improves the accuracy and mathematical correctness of the filter. Leveraging wrist geometry also allows the filter to rely on the accelerometer measurements with more confidence, thus improving the estimate. Dynamic tracking of the gyroscope bias allows the filter to be used immediately after applying power to the IMU sensors, with no calibration required. This is an important benefit if the filter is to be used in conjunction with a clinical device.

Both simulations and experimental tests are conducted to validate the effectiveness of the filter. In both cases, results indicate that the filter is effective at tracking the attitude of a human wrist, even during highly dynamic oscillations. The experimental validation is particularly revealing. During this test, a mechanical test rig is configured to oscillate in a manner that mimics the effects of PD. The noise and vibrations during this test are large, but the filter is still able to achieve an RMS error less than 1 degree.

Chapter 4

Tremor Suppression

Parkinson's Disease can manifest in a number of ways, and with many different tremor profiles, the most common of which is pronation and supination in the wrist. This tremor mode is also the most difficult to stabilize, as evidenced by the lack of viable solutions (mechanical or pharmacological)

The few mechanical attempts to stabilize this unintended pronation and supination require the construction of a large and intrusive exoskeleton, such as the devices built by Rocon et al. [28] and Kiguchi et al. [20]. In order for these exoskeletons to apply a moment force about the wrist, they must be mounted along the entirety of the forearm and the upper arm. The efficacy of such systems appear promising, but their weight and intrusive design make them ill-suited for daily use.

Other researchers, such as Maneski et al. [23], Dosen et al. [13], and Poppendieck et al. [27], have attempted to counter upper limb tremor with a technology called functional electrical stimulation (FES). FES requires small electrodes to be placed on the surface of the forearm in order to stimulate the muscle groups antagonistic to those causing tremor. Systems employing FES technology have shown promise but have disadvantages such as the propensity to cause muscle fatigue. While this fatigue can be reduced or possibly eliminated by keeping the electrical stimulation below the motor threshold [13], FES remains unfeasible for other reasons.

Due to the inherent nature of FES, only superficial muscle groups (close to the surface of the skin) can be stimulated. In the forearm, the muscles responsible for flexion and extension are indeed superficial, but the muscle groups responsible for pronation and supination are deep. For this reason, FES can be applied in the correction of ET but is much less effective in the context of PD. Any attempt to correct a pronation/supination tremor via electrical stimulation would require intramuscular electrodes, an invasive and potentially dangerous prospect.

Intrusive systems such as unwieldy exoskeletons or intramuscular probes are undesirable. This chapter develops a proof-of-concept prototype for a potentially superior alternative. The device is a very early stage prototype, as yet unfit for human testing. This work, therefore, devotes significant effort to the construction of a mechanical testing rig used to evaluate the performance of the tremor suppression device.

4.1 Background

The technical content in this chapter utilizes concepts and methods from several unrelated fields of engineering. At a high level, the tremor suppression device leverages the use of a weighted reaction wheel in order to stabilize the unintended pronation and supination characteristic of Parkinsonian tremor. To understand the design and construction of the system, knowledge of DC motors and modern control algorithms is also required. The following background sections will provide all necessary context for the remainder of the chapter.

4.1.1 Reaction Wheels

A reaction wheel, or momentum wheel as it is sometimes called, is a spinning, weighted disk commonly used in the attitude control of orbiting satellites. The disk, powered by an on-board motor, rotates at a nominal rpm. Once in orbit, any torque applied by the motor will push against the inertia of the wheel and generate a reaction torque on the body of the satellite. Due to the conservation of angular momentum, this reaction torque will be equal in magnitude, but in the opposite direction of the original applied torque. Therefore, the satellite can control its attitude by either accelerating or decelerating the motor.

When designing a traditional reaction wheel system, it is important that the nominal speed of the wheel never saturate, or approach the top speed limit of the motor. If this happens, the system loses the ability to apply a torque in one direction and can no longer be stabilized. For this reason, engineers carefully design the moment of inertia of the wheel to minimize the likelihood of saturation. These optimization calculations are complex, and beyond the scope of this thesis, however interested readers are encouraged to review the work of Manual [24] for a deeper discussion on this topic.

The prototype described in this work will have a reaction wheel mounted as a bracelet around a user's wrist in order to apply a torque about the forearm. For the purposes of this work, the amount of torque the reaction wheel is required to generate to stabilize the system is the most critical design aspect.

In a mechanical system, torque is governed by the equation

$$\tau = I\ddot{\theta} \tag{4.1}$$

Where τ is the torque, I is the moment of inertia of the rotating body, and $\ddot{\theta}$ is the angular acceleration of the rotating body. In subsequent sections of this work, the maximum angular accelerations present in the system will be used with equation (4.1) to calculate the maximum torque needed from the system motors.

4.1.2 DC Motors

In any reaction wheel based system, an actuator is needed to apply a force to the wheel. DC motors were chosen in this work because of their low cost and straightforward operation. DC motors are simple devices that convert a direct current (DC) source into rotational motion. While the internal construction of these motors is not relevant to this work, the properties that govern their operation are crucial. This section briefly introduces the most pertinent equations regarding DC motors. For a more rigorous introduction to the use of DC motors, the reader is encouraged to review [7].

All DC motors convert current to rotational energy, or torque. The relationship between this current and the torque generated by the motor is given by the motor torque equation:

$$\tau = K_t i \tag{4.2}$$

Where K_t is called the motor torque constant and is unique to each motor. K_t is always positive and has units of Nm/A.

The speed of a motor is governed by the motor speed equation given below.

$$\omega = \frac{V - Ri}{K_e} \tag{4.3}$$

Where ω is the steady state motor speed, V is the voltage across the terminals of the motor, R is the electrical resistance of the motor, and K_e is the motor speed constant. K_e is always positive and has units of V/rad/s. Occasionally, a value noted by K_v is also used to describe a motor. When this is the case, $K_v \triangleq 1/K_e$. Although counter intuitive, an important property of the equations above is that K_e and K_t have the same numerical value, despite having different units.

The motor speed equation (4.3) also reveals some interesting motor properties. It shows that for a fixed voltage, the relationship between the speed of a motor and the torque it produces is linear. The maximum current flow through the motor (and therefore the maximum torque) occurs when the motor is at rest. Similarly, the maximum speed of the motor occurs when current and torque are at a minimum. This relationship is depicted in Fig. 4.1.

This relationship means that controlling the speed of a motor requires the ability to control the current passing through it. However, because the resistance across the leads of a motor is constant, the current through the motor is directly proportional to the applied voltage. Therefore, in an electromechanical system, the easiest way to control the speed of a DC motor is to vary the voltage applied across the positive and negative terminals. In practice, this is achieved using pulse-width-modulation (PWM), in which the voltage from the power supply is

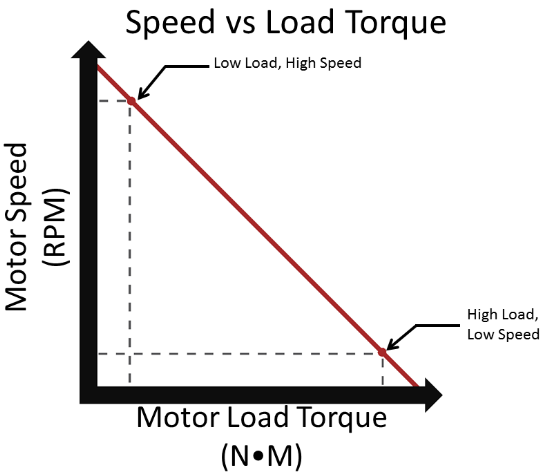


Figure 4.1: Relationship between speed and torque of a DC motor. This relationship is governed by equation 4.3. The slope of this line is governed by constants unique to each motor.

rapidly switched between on and off. By varying ratio between the “on” periods and the “off” periods, the effective voltage applied to the motor can be varied. This ratio, called the duty cycle, is illustrated by Fig. 4.2. The relationship between the effective motor voltage and the duty cycle is given by:

$$V_{eff} = V_{Suply} \frac{\%\ duty\ cycle}{100} \tag{4.4}$$

4.1.3 PID Controllers

The previous section describes how to use a PWM duty cycle to control the speed of a motor. This information is only useful, however, if the required motor speed is known. To complicate matters, the desired motor speed may not be a constant value. Instead, it could be dependent on additional factors within the system. These challenges are addressed by a field of

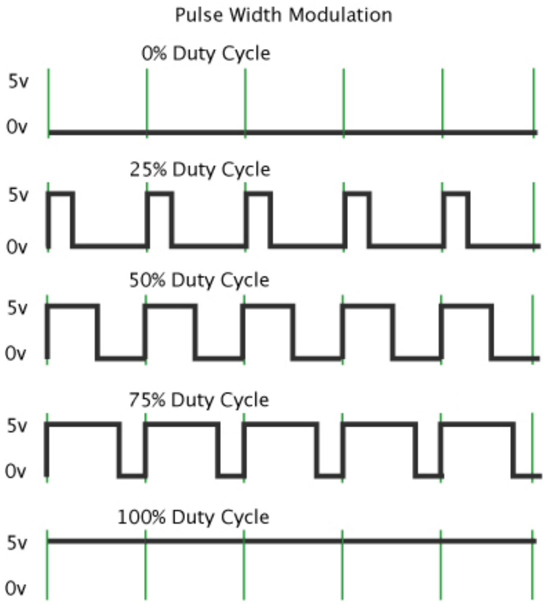


Figure 4.2: Illustration of various PWM duty cycles. Duty cycles are used to control the voltage across the terminals of a motor. A higher duty cycle causes a higher effective voltage and thus a larger torque output from the motor.

engineering called control theory.

The role of a control algorithm in a robotic system is to read data from sensors and compute the required signal for the system's actuators. In this work, the control signal is simply a PWM signal that is sent to the motors. There are many architectures that can underpin a control algorithm (e.g. linear quadratic regulators, optimal control, model predictive control, hybrid control, etc.) and the pros and cons of each method are thoroughly explored in existing literature [17, 30]. Among the architectures commonly used in modern engineering systems, the proportional-integral-derivative (PID) control scheme is one of the simplest to implement. It is also known to be impressively robust despite its simplicity.

A PID controller comprises a proportional (P), integral (I) and a derivative (D) component. These three sections operate independently and are summed to generate the final control signal. Each of these components will be described in detail. First, however, some basic terminology must be defined.

- **Set point:** The set point is the desired state of the system.
- **Error (e):** The error is the difference between the set point and the current state of the system.
- **Control Variable (CV):** The control variable is the output of the control algorithm. It is the signal that is sent to the motors or actuators. A larger value for the control variable causes a more aggressive response from the motors.

4.1.3.1 Proportional Controller

The proportional controller operates by multiplying the error by a constant value, called the proportional gain, and adding this to the control variable. The proportional gain, written as K_p, is a parameter that is tuned to adjust the performance of the algorithm. Larger values of K_p cause the system to react more quickly but can lead to overshoot, or in extreme cases, instability.

$$CV = K_p e \tag{4.5}$$

As the name implies, the proportional controller increases the control variable linearly with respect to error. Therefore, larger errors result in a larger control variable and a more aggressive response from the actuators in the system.

4.1.3.2 Integral Controller

The integral controller operates by integrating the error signal over time, multiplying this integral by a constant, and adding this to control variable. The constant is called the integral gain and is written as K_i. The integral controller is useful to reject a steady state error in a system. It is very common for mechanical systems to have a “dead band” region near the set point. In this region, the error is very small, and the control signal generated by the proportional controller alone is not sufficient to make the correction. In this case, the integral controller will slowly accumulate this error and generate a control signal that can close the gap. The integral controller is rarely used alone. It is usually coupled with the proportional controller as follows.

$$CV = K_p e + K_i \int e\,dt \tag{4.6}$$

Larger values of K_i cause the steady state error to correct more quickly but, as with the proportional controller, large values of K_i can make the system oscillate or become unstable.

4.1.3.3 Derivative Controller

The derivative controller operates by evaluating the rate of change of the error signal, multiplying this value by a constant, and adding the result to the control variable. The constant is called the derivative gain and is written as K_d. The derivative controller causes the aggressiveness of the control signal to decrease when the error signal is rapidly decreasing. As such, a properly tuned derivative gain can help reduce overshoot in a system. As with the integral controller, the derivative controller is usually used in conjunction with the proportional controller (PD control), and sometimes with both the proportional and the integral controller (PID control). The equation and a block diagram for a full PID controller are shown below.

$$CV = K_p e + K_d \frac{de}{dt} + K_i \int e\, dt \tag{4.7}$$

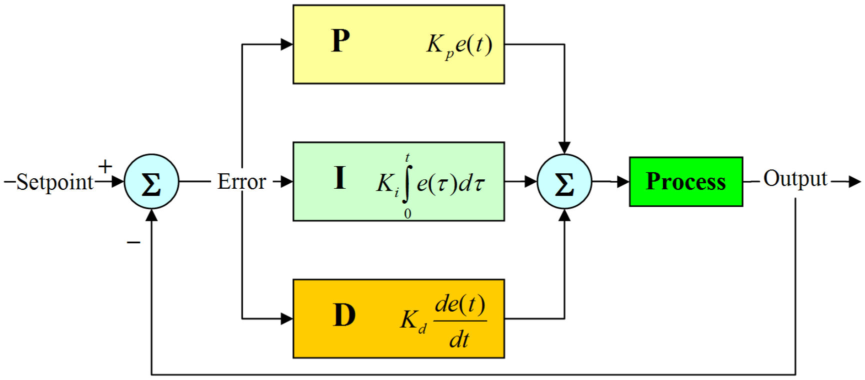


Figure 4.3: Sample block diagram for a simple PID controller. A PID controller is implemented with the tremor suppression device in order to dampen Parkinsonian tremor.

4.2 Mechanical Test Rig

The ultimate goal of this work is to develop a device that can dampen a Parkinsonian tremor. Before the device can be built, however, a system must be developed that can ensure reliable, safe testing of the prototype. Within the scope of this thesis, the final prototype will be built for proof-of-concept testing only. Because considerations for comfort or power consumption are outside the scope, testing the device on diagnosed patients or healthy volunteers is not feasible.

To provide a platform with which to test tremor suppression system, a mechanical test rig was developed that can effectively mimic Parkinsonian tremor. This section gives a detailed description of the design and construction of the test rig.

4.2.1 Design Requirements

At a high level, the mechanical test rig must mimic Parkinsonian tremor. To do so effectively, the rig should be able to produce a tremor with the same motion characteristics of PD and should enable the testing of tremor-suppressing prototypes and control algorithms in a risk-free setting. These goals are formalized into a set of specific design requirements as follows:

1. **Form Factor:** The most notorious and debilitating tremor mode found in patients with PD is the pronation and supination of the wrist. This tremor mode affects the entire forearm. Therefore, the rig must contain a segment which approximately resembles a human forearm in both size and shape.

2. **Tremor Characteristics:** Several studies suggest that PD typically presents as a 3-7Hz tremor. While less is known about tremor amplitude, discussions with doctors and analyses of online video indicate that the tremor can increase to about $\pm 30^\circ$. The oscillations in the affected forearm are approximately sinusoidal along the longitudinal axis of the forearm. Therefore, the rig must be able to generate approximately sinusoidal tremors at frequencies of 3Hz to 7Hz and amplitudes of 0° to 30°.

3. **Tremor Suppression:** The ultimate goal of the prototype device will be to suppress the tremor induced in the rig. Therefore, the mechanical rig cannot be completely rigid. The

portion of the rig that is oscillating must be decoupled in such a way that a reasonable force can be applied to stabilize it.

4. **Data Acquisition:** To know if the proposed tremor suppression device is effective, the rig must have a way to measure and record the motion of the tremor. This measurement should have a precision of 0.01° or better and must be able to sample at 70Hz or faster.
5. **Convenience:** In order to maximize the convenience of the rig, it must be small enough to fit comfortably on a lab bench. Also, it must be powered by a standard 12v power supply.
6. **Material/Manufacturing Limitations:** In order to minimize costs and facilitate the construction of a rapid prototype, the test rig is manufactured using an in-house laser cutter. This manufacturing method is very efficient but limits the design choices and materials that can be used. The laser cutter is capable of cutting through flat sheets of wood or acrylic. Therefore, the rig must be designed such that all structural components can be cut from a flat board and pieced together for final construction.

4.2.2 Physical Design

With the design requirements established, it is clear that the central component of the rig is the portion that simulates the motion of Parkinsonian tremor. Design requirement 1 dictates that this segment must reasonably resemble the size and shape of a human forearm. To satisfy this requirement, a 2in diameter wooden dowel is used. As shown in Fig. 4.4, this wooden dowel is supported by panels of medium-density fiber board (MDF) on each side. These

MDF panels fit together with alternating tabs and slots. The two main MDF panels support a 5/16in bolt protruding into the wooden dowel. These bolts are rigidly fixed to the wooden dowel but pass through bearings in the MDF, allowing them to rotate in place. Thus, the entire wooden dowel can rotate freely.

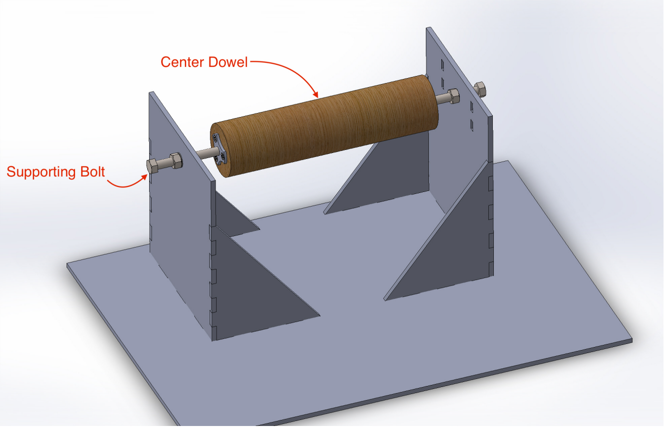


Figure 4.4: Center dowel of the mechanical test rig. The 2in diameter dowel is made from poplar and designed to rotate freely so that it can mimic the effects of Parkinson's Disease.

In order to replicate the specific characteristics of Parkinsonian tremor, a mechanism is developed that can drive the center dowel within the required amplitude and frequency ranges. This is achieved with a customized four-bar linkage which can be seen in Fig. 4.5. The linkage, classified as a crank-rocker mechanism, is added to one side of the center dowel. The rocker is fixed to the bolt supporting the center dowel. The crank, which can rotate 360° and is driven by a DC motor can be seen toward the right side of Fig. 4.5. Design requirement 2 mandates that the linkage be able to generate tremor amplitudes from 0° to 30°. To meet this design specification, the linkage is built with a number of different mounting configurations. This allows the lengths of the linkage segments to be quickly adjusted depending on the desired tremor amplitude. To compute the specific linkage dimensions needed, a more detailed analysis is presented.

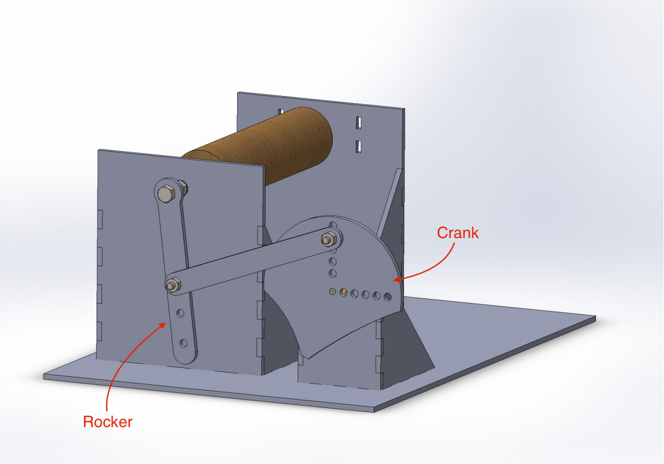


Figure 4.5: Crank-rocker mechanism used to simulate tremor. As the crank wheel rotates 360 degrees, the rocker arm oscillates back and forth through a limited range of angles.

A schematic of the four-bar linkage is presented in greater detail in Fig. 4.6. In this figure, the crank arm is labeled s, the rocker arm is labeled p, and the linkage between them is labeled l. The relationship between the lengths of p, l, and s controls the range of motion of the rocker arm. As shown in Fig. 4.5, the top point of the rocker arm is attached to the center dowel of the rig. Therefore, the challenge becomes designing the lengths of p, l, and s to achieve possible tremor amplitudes of 0° to 30°.

As can be seen by Figs. 4.7 and 4.8, the maximum angle that can be achieved by the rocker arm occurs when links s and l are collinear and not overlapping. The minimum angle that can be achieved by the rocker arm occurs when links s and l are collinear and overlapping. With basic trigonometry, θ_{max} and θ_{min} can be solved in terms of p, l, and s as follows.

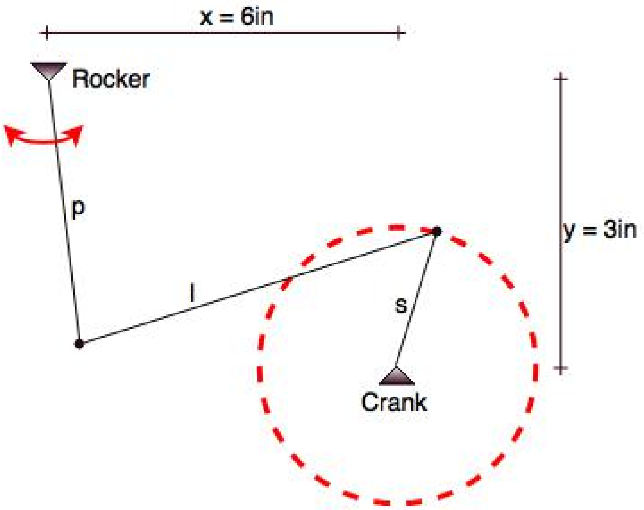


Figure 4.6: Graphical depiction of the crank-rocker mechanism used to simulate tremor. The crank arm, *s*, is connected to a motor and has a full range of motion. The rocker arm, *p*, is connected to the center dowel and has a limited range of motion.

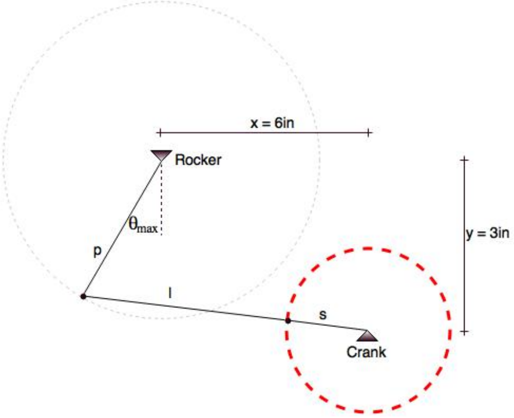


Figure 4.7: Graphical depiction of the crank-rocker mechanism at its maximum angle position.

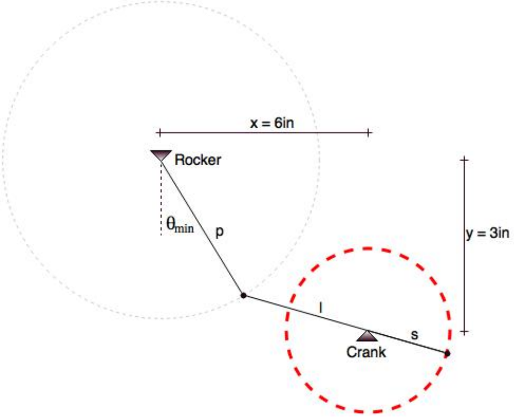


Figure 4.8: Graphical depiction of the crank-rocker mechanism at its minimum angle position.

$$\theta_{max} = \cos^{-1}\left(\frac{p^2+(x^2+y^2)-(l-s)^2}{2h\sqrt{x^2+y^2}}\right) - \tan^{-1}\left(\frac{x}{y}\right) \tag{4.8}$$

$$\theta_{min} = \frac{\pi}{2} - \cos^{-1}\left(\frac{p^2+(x^2+y^2)-(l-s)^2}{2h\sqrt{x^2+y^2}}\right) - \tan^{-1}\left(\frac{y}{x}\right) \tag{4.9}$$

Using the above equations, optimal values for p, l, and s are determined by trial and error. The middle link, l is fixed at $l = 6$in. With this value set, a range of values for p and s are selected as $p \in \{3\text{in}, 4\text{in}, 5\text{in}\}$ and $s = \{0.4\text{in}, 0.6\text{in}, 0.8\text{in}, 1.0\text{in}, 1.2\text{in}, 1.4\text{in}, 1.6\text{in}, 1.8\text{in}, 2\text{in}, 2.2\text{in}\}$. Tables 4.1 - 4.3 list the possible lengths of each link, as well as the angle range generated by each configuration. As can be seen by the tables, the different configurations enable tremor amplitudes of ±0.91° to ±45.8°. This successfully satisfies design requirement 2.

s [in]	**p [in]**	**l [in]**	θ_{max}	θ_{min}
2.2	3	6	45.77°	−45.77°
2	3	6	40.95°	−40.95°
1.8	3	6	36.35°	−36.35°
1.6	3	6	31.93°	−31.93°
1.4	3	6	27.65°	−27.65°
1.2	3	6	23.49°	−23.49°
1.0	3	6	19.43°	−19.43°
0.8	3	6	15.45°	−15.45°
0.6	3	6	11.53°	−11.53°
0.4	3	6	7.66°	−7.66°

Table 4.1: Possible values for θ_{min} and θ_{max} when the mechanical rig is set to $p = 3$in and $s = \{0.4\text{in}, 0.6\text{in}, 0.8\text{in}, 1.0\text{in}, 1.2\text{in}, 1.4\text{in}, 1.6\text{in}, 1.8\text{in}, 2\text{in}, 2.2\text{in}\}$. With this configuration, the rig can generate a maximum tremor of ±46° and a minimum tremor of ±8°.

s [in]	p [in]	l [in]	θ_{max}	θ_{min}
2.2	4	6	33.24°	-33.24°
2	4	6	29.77°	-29.77°
1.8	4	6	26.39°	-26.39°
1.6	4	6	23.10°	-23.10°
1.4	4	6	19.89°	-19.89°
1.2	4	6	16.74°	-16.74°
1.0	4	6	13.64°	-13.64°
0.8	4	6	10.60°	-10.60°
0.6	4	6	7.60°	-7.60°
0.4	4	6	4.64°	-4.64°

Table 4.2: Possible values for θ_{min} and θ_{max} when the mechanical rig is set to $p = 4\text{in}$ and $s = \{0.4\text{in}, 0.6\text{in}, 0.8\text{in}, 1.0\text{in}, 1.2\text{in}, 1.4\text{in}, 1.6\text{in}, 1.8\text{in}, 2\text{in}, 2.2\text{in}\}$. With this configuration, the rig can generate a maximum tremor of $\pm 33^\circ$ and a minimum tremor of $\pm 5^\circ$.

s [in]	p [in]	l [in]	θ_{max}	θ_{min}
2.2	5	6	24.21°	-24.21°
2	5	6	21.43°	-21.43°
1.8	5	6	18.72°	-18.72°
1.6	5	6	16.05°	-16.05°
1.4	5	6	13.43°	-13.43°
1.2	5	6	10.86°	-10.86°
1.0	5	6	8.32°	-8.32°
0.8	5	6	5.82°	-5.82°
0.6	5	6	3.35°	-3.35°
0.4	5	6	0.91°	-0.91°

Table 4.3: Possible values for θ_{min} and θ_{max} when the mechanical rig is set to $p = 5\text{in}$ and $s = \{0.4\text{in}, 0.6\text{in}, 0.8\text{in}, 1.0\text{in}, 1.2\text{in}, 1.4\text{in}, 1.6\text{in}, 1.8\text{in}, 2\text{in}, 2.2\text{in}\}$. With this configuration, the rig can generate a maximum tremor of $\pm 24^\circ$ and a minimum tremor of $\pm 1^\circ$.

The linkage designed above successfully meets the requirements for tremor amplitude but does not address the need to generate tremors at multiple frequencies. The frequency of the induced tremor is determined directly by the angular velocity of the crank arm. Recall that this crank arm is driven by a single DC motor. Design specification 5 requires that the system is powered by a 12v supply voltage. Therefore, a motor is selected that can achieve the desired angular velocity with a 12v supply voltage. The specific motor used is a brushed DC motor manufactured by CNDF. It is rated for 1000rpm (16.6Hz) at 12v. This is more than twice as fast as the required frequency but ensures that a frequency of 7Hz will be attainable even if the manufacturing tolerances of the motor cause a reduced maximum speed. To control the speed of the motor, a pulse-width-modulated (PWM) motor drive board is purchased from RioRand. This board is powered by a 12v supply and has a physical dial which is used to manually control the PWM duty cycle sent to the motor. By changing the duty cycle, frequencies of 0Hz to 16Hz are possible. This more than covers the desired range of tremor frequencies and satisfies the second design requirement.

Design requirement 3 stipulates that while shaking, the rig cannot be completely rigid. It must be pliable such that an external force is capable of stabilizing it. If the rig were built exactly as depicted in Fig. 4.5 the connection between the crank arm and the center dowel would be completely rigid. In order to remedy this, a torsional spring is added at the interface between the top of the rocker arm and the bolt that supports the center dowel. This torsional spring decouples the system while still transferring the force of the rocker arm to the center dowel.

Design requirement 4 describes the specification that the motion of the rig must be measurable. It is critical that these measurements be accurate and precise because they will be the only reliable way evaluate the performance of the tremor suppression device. As described above, the center dowel is contained such that it can only rotate about its longitudinal axis. Therefore, the only measurement that is required is the angle of the center dowel about its axis of rotation. This measurement can be attained by mounting an encoder to the end of one of the bolts supporting the center dowel. Several types of encoders exist (including optical and magnetic), but for this application, an AMS 5048A magnetic encoder is used. The AMS 5048A has 14 bits of precision, allowing measurements to be taken at a granularity far superior to the 0.01° required. The encoder is mounted to the bolt opposite the four-bar linkage. The final design of the mechanical test rig, complete with the encoder and the PWM motor drive board can be seen in Fig. 4.9.

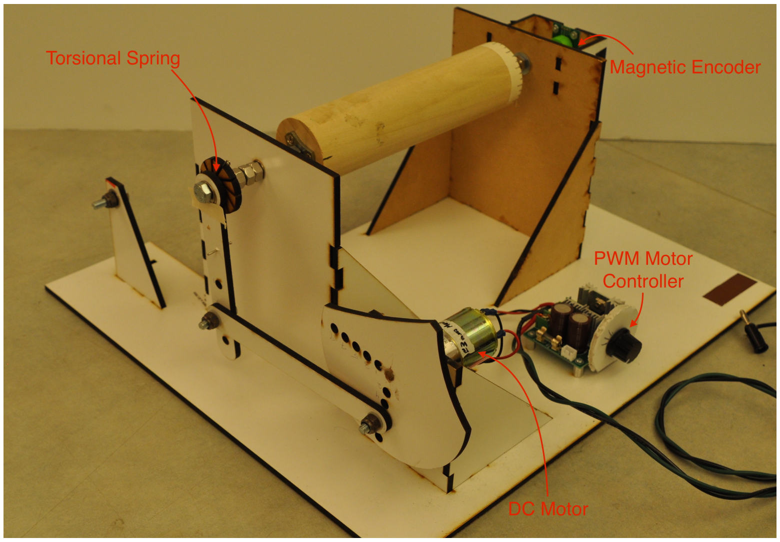


Figure 4.9: Mechanical test rig used to simulate Parkinsonian tremor. This rig can simulate a Parkinsonian tremor at multiple frequencies and amplitudes. It has a magnetic shaft encoder to record the absolute angle of the center dowel and a torsional spring to decouple the dowel from the drive motor.

4.3 Tremor Suppression Device

The tremor suppression device is intended to dampen a pronation/supination tremor caused by Parkinson's Disease. Due to the early stage of this research, testing the device on patients, or even healthy volunteers is beyond the current scope. Instead, the device will be tested with the mechanical test rig developed in Section 4.2.

The prototype in this work attempts to stabilize tremor by use of a weighted reaction wheel, similar to the stabilization of satellites discussed in Section 4.1.1. The device takes the form of a wristband with a weighted reaction wheel that can spin freely around the perimeter of the wrist. Motors mounted to the wristband will apply a torque to the reaction wheel, and the rotational inertia of the wheel will apply an equal and opposite torque to the wrist. Therefore, if the motors apply torque at the right time and intensity, the tremor can be stabilized. Although the device will not be tested with human users, care is taken to ensure that the design is adaptable to human use in the future. The formal design requirements for the tremor suppression device are described below.

4.3.1 Design Requirements

The tremor suppression device, at this phase, is only intended to prove merit in the use of reaction wheels for the stabilization of tremor. Therefore, certain design constraints that may seem important for a commercial device are ignored. Specifically, an analysis of power consumption and device comfort are omitted. The requirements at this stage of the project are as follows.

1. **Capable of Robust Tremor Attenuation:** This design requirement is the highest priority. At this stage, since the tremor suppression device cannot be tested on humans the prototype will be mounted to the mechanical test rig described in Section 4.2. The device will be considered a success if it can reduce the tremor amplitude in the rig by 50% or more. It must also be robust to variation in the amplitude and frequency of the tremor.

2. **Minimally Intrusive:** The tremor suppression device must be small and light enough that a future patient could wear it comfortably for an extended period. This means that the device must not require an elaborate exoskeleton that spans the entire arm. It must be small enough to eventually fit in a single, compact wristband.

3. **Reasonably Adaptable to Commercial Use:** This design requirement does not refer to the physical design of the product, but instead to the sensors and software used. The prototype device should be constructed with sensors and motors that are viable in a commercial product. Moreover, the software and control algorithm must not rely on any inputs that would not be available in practice. This means that no external tracking mechanism can be used to measure the tremor. Furthermore, all control and estimation algorithms must be efficient enough to run in real-time on a microcontroller.

4. **Material/Manufacturing Limitations:** The tremor suppression prototype will be primarily constructed using a CubePro 3D printer. The 3D printer allows for a rapid development and testing process. It also enables the manufacture of complicated geometries that would be difficult to create with a laser cutter or traditional machining tools. While any future commercial device would be manufactured with different processes, a 3D printer is an ideal choice for the initial prototype. Though, several tests of the 3D printer indicated poor performance when printing components with an overhang. Therefore, care is taken to avoid overhangs during the physical design process.

4.3.2 Reaction Wheel Design

The first step in constructing the tremor suppression device is the design of the reaction wheel system. Recall from section 4.1.1 that the most critical aspect of this design is an analysis of the torque requirements of the system which will define other critical parameters such as the number and size of the motors that will be used. Once finished, the reaction wheel system will influence the physical design of the prototype device described in Section 4.3.3.

4.3.2.1 Torque Requirements

To identify the proper size of the reaction wheel and motors for the system, it is necessary to determine the torque required to counteract the tremor. This can be found by analyzing the characteristics of the mechanical test rig. The center dowel of the rig is made of poplar and has the following properties.

$$\text{density} = \rho_{poplar} = 350 kg/m^3 \tag{4.10}$$

$$\text{radius} = r_{dowel} = 0.0254 m \tag{4.11}$$

$$\text{length} = l_{dowel} = 0.205 m \tag{4.12}$$

$$\text{mass} = m_{dowel} = \pi r^2 * l * \rho_{poplar} = 0.1454 kg \tag{4.13}$$

To account for the non-wood components that rotate with the center dowel (i.e., the bolts and nuts), a scaling factor of 1.5 is multiplied by this mass. Therefore, the approximate mass and moment of inertia, I, of the rotating portion of the rig are given as:

$$m_{rig} = m_{dowel}(1.5) = 0.2181kg \tag{4.14}$$

$$I_{rig} = \frac{m_{dowel}(r_{dowel}^2)}{2} = 7.0367 \times 10^{-5} kg\, m^2 \tag{4.15}$$

As mentioned in Chapter 1, PD typically presents as a 3-7Hz tremor from $\pm 0°$ to $\pm 30°$. For this analysis, it is assumed that the rig will be oscillating at a frequency of 5Hz with an amplitude of 15° ($A = 0.261$rad). This tremor represents the mean amplitude and frequency seen in a typical PD patient.

By making the simplifying assumption that the tremor is a pure sinusoid, the angle of the center dowel, θ_{rig}, can be written as:

$$\theta_{rig} = A\sin(\omega t) \tag{4.16}$$

$$\dot{\theta}_{rig} = -A\omega\cos(\omega t) \tag{4.17}$$

$$\ddot{\theta}_{rig} = -A\omega^2\sin(\omega t) \tag{4.18}$$

Newtonian physics gives the relationship between angular acceleration and torque as:

$$\tau = I\ddot{\theta} \tag{4.19}$$

Where τ is torque, and I is the moment of inertia of the rotating object. It is clear by inspection that $\ddot{\theta}_{rig}$ cannot exceed $A\omega^2$. Therefore the maximum torque required to stabilize the rig is:

$$\tau_{max} = I_{rig}A\omega^2 = (7.037 \times 10^{-5})(0.261)((5)(2\pi))^2 = 0.0182Nm \tag{4.20}$$

4.3.2.2 Motor Selection

Based on the above calculations the motors will theoretically need to output $0.0182Nm$ in order to stabilize the test rig. However, unmodeled factors such as friction, stiction, and electrical inefficiencies will require greater torque. Therefore, a design factor of 2 is placed on this torque requirement, making the desired torque equal to $0.0364Nm$.

To reduce the size of any single motor on the device, this torque requirement is divided across three motors. The specific motor used is the CL-0720-14 coreless motor, sold by Micro Motor Warehouse. The listed specifications of this motor can be seen in Table 4.4.

K_v	14000
Max Voltage	3.7V
Stall Current at 3.7V	1.8A

Table 4.4: Specifications of the CL-0720-14 coreless motor, sold by Micro Motor Warehouse. K_v is the motor speed constant and is equal to $1/K_t$. Three of these motors are used on the tremor suppression device to apply a torque to the weighted reaction wheel.

Based on these specifications, the motor torque constant, K_t, can be inferred to be:

$$K_t = \frac{1}{14000} \tag{4.21}$$

At the maximum current of $1.8A$, the motor torque equation can be used to calculate the stall torque produced by the motor as follows.

$$\tau = K_t i \tag{4.22}$$

$$\tau_{stall} = \frac{1}{14000} 1.8 = 1.286 \times 10^{-4} Nm \tag{4.23}$$

Where i is current, and τ is the output torque from the motor. As will be explained in Section 4.3.3, there is a gear ratio of 7 to 1 between the CL-0720-14 motors and the reaction wheel. Therefore, each motor contributes 9×10^{-4}Nm of torque. Combining three motors provides a total torque output of $.0027 Nm$. Although this falls well short of the desired torque output, it will be sufficient to show merit in the overall design concept.

4.3.2.3 Mass of Reaction Wheel

When designing a traditional reaction-wheel system, an appropriately sized reaction wheel must be used in order to prevent saturation. In this device, however, the reaction wheel will never have the opportunity to reach to full speed as it will rapidly switch between spinning in the clockwise and counter-clockwise direction. Thus while the mass of the wheel may impact the performance or efficiency of the system, it is not a critical aspect of the design.

4.3.3 Physical Design

Given the reaction wheel developed in Section 4.3.2, the physical design of the tremor suppression prototype can be crafted to meet the system needs. Since the device is only intended to be used with the mechanical test rig, not with human subjects, the metric used to evaluate the effectiveness of the device is its ability to dampen the tremor in the mechanical test rig. The first attribute of the physical design is to ensure that device can mount easily to the test rig.

To achieve this, the device is designed as a 2in diameter ring that fits around the center dowel of the test rig. The ring is left partially open so that it can slip onto the rig easily. As can be seen in Fig. 4.10, the opening has a slot on both ends where a strap can be used to tighten the ring around the center dowel. The three motors that will be used to drive the system fit tightly into the cylindrical mounting holes around the perimeter of the ring.

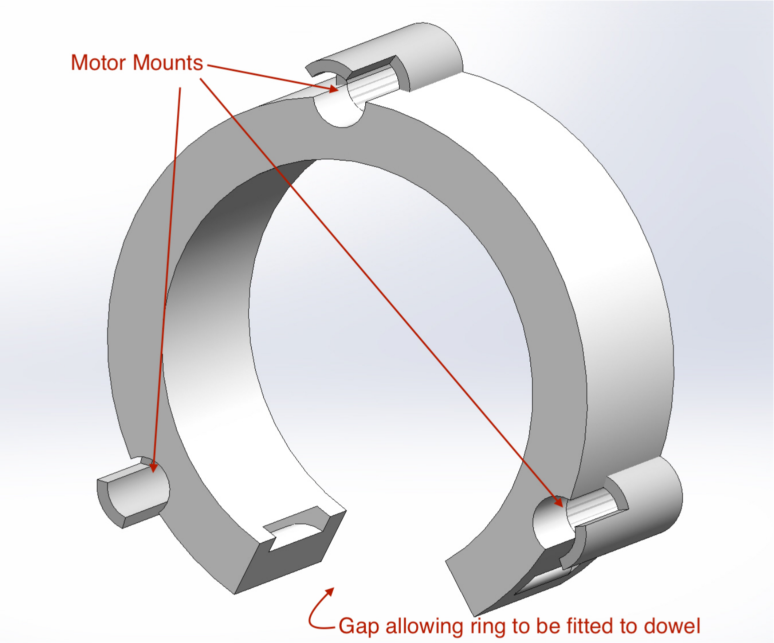


Figure 4.10: Ring used to house the motors of the tremor suppression device. The three motors fit tightly into the cylinders around the perimeter of the ring. This ring slips easily onto the mechanical test rig and is secured with a Velcro strap that passes through the slots on the bottom.

A pinion gear is mounted on the shaft of each motor, and the reaction wheel described in Section 4.3.2 encircles the three pinion gears as an internal gear. Each pinion gear has 11 teeth, and the reaction wheel has 77 teeth. This can be seen in Fig. 4.11 which depicts the complete 3D design for the tremor suppression prototype. In order to manufacture the device, a CubePro 3D printer was used. This 3D printer was able to create the bracelet, reaction wheel, and pinion gears with sufficient accuracy and precision. A picture of the 3D printed device is shown in Fig. 4.12. Although a rigorous analysis was not conducted for the optimal weight of the reaction wheel, it quickly became clear that the light ABS plastic from the 3D printer was not sufficient. To address this, two small steel nuts were glued to the perimeter of the reaction wheel. As discussed in Chapter 6, future work in this research would be to optimize the weight of the reaction wheel and test the use of more motors.

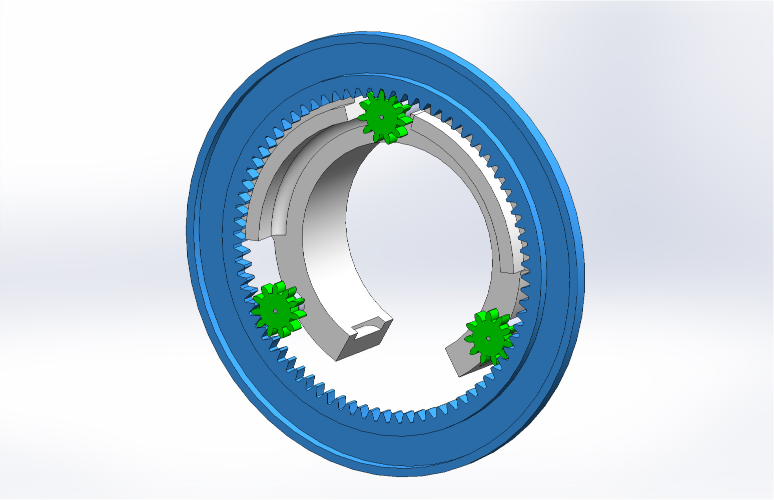


Figure 4.11: A complete rendering of the tremor suppression device. This device uses three motors, each with a small pinion gear (rendered in green). The motors push against the inertia of the reaction wheel (rendered in blue) to apply stabilizing torque to the mechanical test rig. All the components in the prototype are manufactured using a CubePro 3D printer.

Figure 4.12: Front and back of the tremor suppression device. This 3D-printed device features a weighted reaction wheel that is used to dampen Parkinsonian tremor. The strap at the bottom of the device allows for easy mounting to the mechanical test rig developed in Section 4.2.

4.4 Control Algorithm

With the tremor suppression prototype built and mounted to the test rig, the final step in the design becomes the development of a control algorithm that can dampen the tremor. Many complicated types of control algorithms exist that would likely perform quite well in this application. Examples include the use of model predictive control, optimal control, or a linear quadratic regulator (LQR). If the goal were to optimize the performance of the device, all options would need to be explored and thoroughly tested. However, within the scope of this thesis, the goal is simply to prove that the device is capable of attenuating tremor. A PID control architecture lends itself well to this objective. As discussed in Section 4.1.3, PID controllers are known for their straightforward implementation and relatively easy tuning.

In this controller, the proportional gain, K_p, operates on the absolute angle of the center dowel with respect to the horizontal axis. This angle is denoted by ϕ. The derivative gain, K_d, operates on the angular velocity of the center dowel, represented by $\dot{\phi}$. In order to simplify the control algorithm, it is assumed that the oscillations of the center dowel are centered perfectly at zero. Testing of the rig indicates that this is a reasonable assumption.

With the tremor centered perfectly at zero, there is no need for an integral component in the PID controller for this proof of concept. The rig will be shaking continuously, and there will be no steady state accumulation of error. Thus the integral gain, K_I, is held at zero and the algorithm becomes a simple PD controller.

The PD controller requires measurements of ϕ and $\dot{\phi}$ in order to operate. These values could be extracted from the magnetic encoder mounted to the shaft of the test rig, but this would not be representative of the target application. If the device were to be produced commercially, the angle of the wrist would need to be estimated, as it would not be possible to measure it directly. A realistic alternative would be to mount an IMU to the device and use this IMU to estimate the wrist attitude.

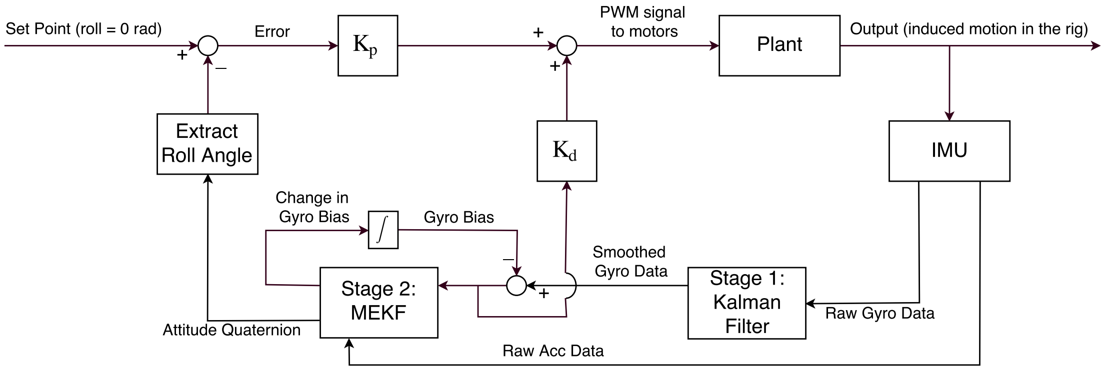


Figure 4.13: Block diagram for the PD controller and attitude estimator. The attitude estimator used here is identical to the two-stage Kalman filter developed in Chapter 3. The PD controller uses the outputs of this filter to generate a control signal for the motors.

To test the feasibility of this approach, an IMU is mounted to the center dowel of the rig. The attitude estimator developed in Chapter 3 is then used to estimate the attitude of the wrist. While oscillating, the rig only rotates along the roll axis of the IMU. Therefore, the roll estimate from the attitude estimator is the input ϕ into the PD controller. Gyroscopes in the IMU are able to measure the rotation rates of the center dowel directly, but using this raw measurement would make the system susceptible to noise and bias. Fortunately, the first stage of the filter developed in Chapter 3 smooths these raw gyroscope measurements. Therefore the input $\dot{\phi}$ for the PD controller is taken from the output of the stage 1 filter described previously. This entire architecture, including the attitude estimator, can be seen in Fig. 4.13. The output of the control algorithm is a PWM signal which is sent to the three motors in an attempt to stabilize the rig. Any residual motion in the rig is then measured by the IMU, and the feedback loop continues. In this way, the PD controller acts as a regulator attempting to suppress the pronation and supination of the wrist.

4.4.1 Tuning the PD Controller

Tuning a control algorithm is a process by which the various parameters in a system are optimized. Within the context of the PD control architecture described in Section 4.4, tuning the algorithm only requires the optimization of two parameters: K_d and K_p. Therefore, the goal of this section is to find values of K_d and K_p that will maximize the tremor attenuation.

Because there are only two values that can be adjusted to impact the performance of the algorithm, a grid search is used to find the best combination. This means that the controller is tuned by testing performance across a range of values for K_d and K_p. Each combination of values is tested directly on the mechanical rig, and the results of this grid search can be seen in Fig. 4.14. The z-axis in the plot shows the RMS value of the roll axis as measured by the magnetic encoder on the test rig.

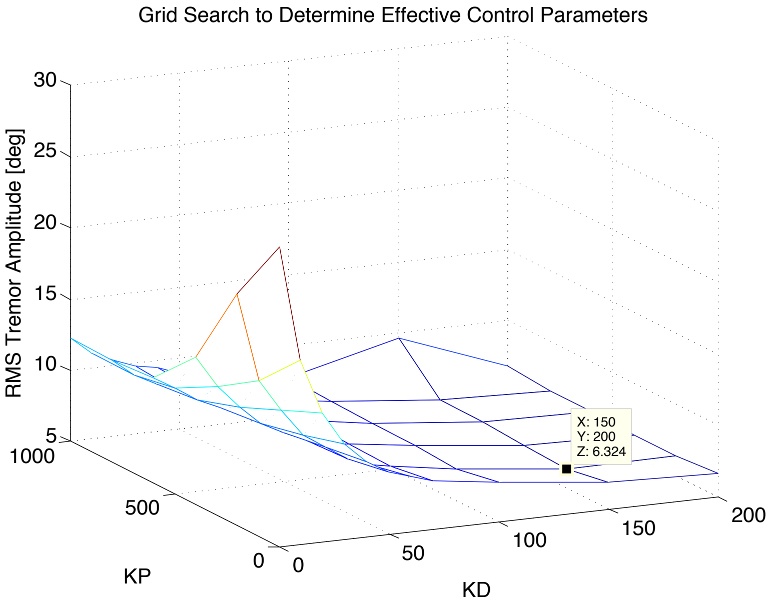


Figure 4.14: Grid search to determine effective control parameters. The PD controller is evaluated across a range of values for K_d and K_p. Each combination of values is tested directly on the mechanical rig. The minimum value is labeled.

When the control algorithm is completely turned off ($K_d = 0$ and $K_p = 0$) the RMS tremor amplitude is 26.0°. The lowest point in the graph occurs at values of $K_d = 150$ and $K_p = 200$. At this point, the RMS tremor amplitude is 6.32° - a 76% reduction. To better visualize the effectiveness of the tremor suppression device, the shaft angle of the rig is plotted against time in Fig. 4.15. The blue section of the plot shows the shaft angle of the oscillating rig while the control algorithm is turned off while the red section of the plot shows the effects of turning the controller on.

While this result is impressive, it only demonstrates that the tremor suppression is effective against one specific tremor. At this point, there a chance that the control algorithm would fail if applied to tremors of different frequency or amplitude. In the next section, the robustness of the system will be evaluated by testing it against multiple different tremors.

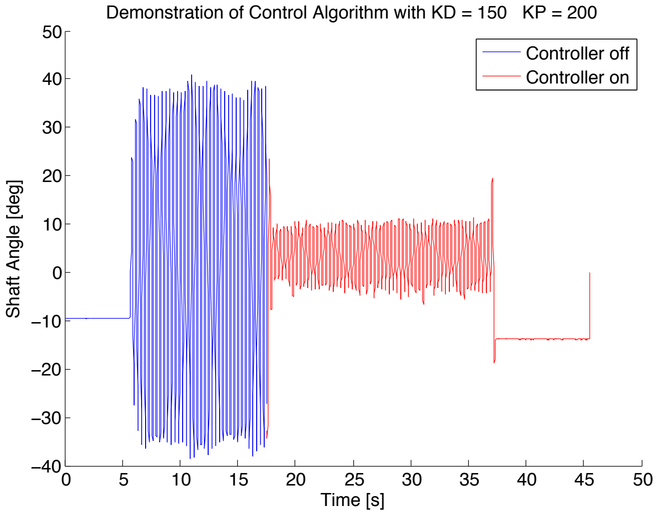


Figure 4.15: Shaft angle with tremor suppression device off and on. The optimal values of K_d and K_p are used. There is a 76% reduction in tremor amplitude when the device is operating. This meets the design requirement of a 50% attenuation factor. More tests are needed to demonstrate effectiveness across a range of tremor parameters; these are presented in Section 4.5.

4.5 Robustness Analysis

Analysis of the grid search conducted in Section 4.4.1 indicates that the tremor suppression device can dampen oscillatory motion in the wrist. The original design specification stated that a 50% reduction in tremor is sufficient. While this specification has been met, more tests are required to demonstrate effectiveness across a range of tremor parameters. To test the robustness of the device and the control algorithm the rig was adjusted to produce tremors of ±10, ±20 and ±30 degrees. At each amplitude, the frequency was tested at 3 different speeds, and the absolute angle of the center dowel was recorded with the magnetic encoder. Figures 4.16-4.18 show the results of these tests. In each trial, the control algorithm was turned off for the first few seconds and then turned on. In all trials, the optimal values of K_d and K_p (as determined in Section 4.4.1) are used.

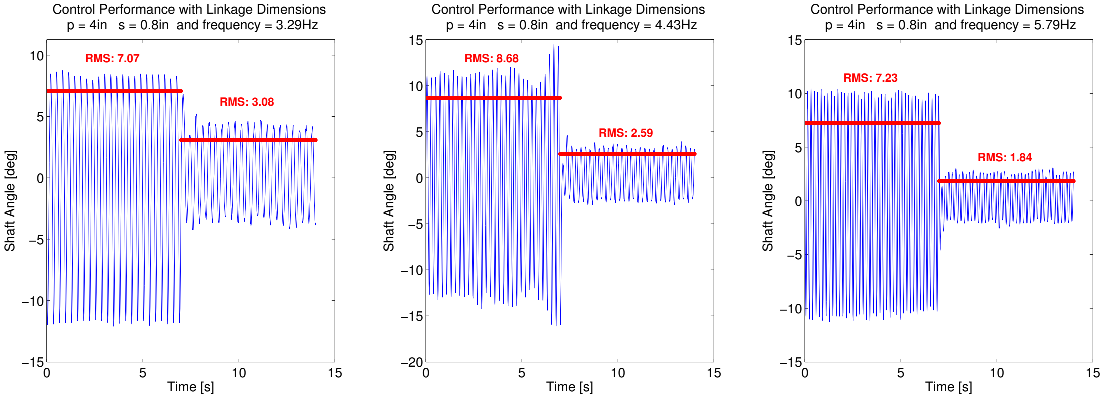


Figure 4.16: Results of robustness test for the tremor attenuation system with linkages in the mechanical test rig set to $p = 4.0$in, $s = 0.8$in. The individual plots show different tremor frequencies with this test configuration. From left to right, the plots depict a tremor attenuation of 56.52%, 70.12%, and 74.62%.

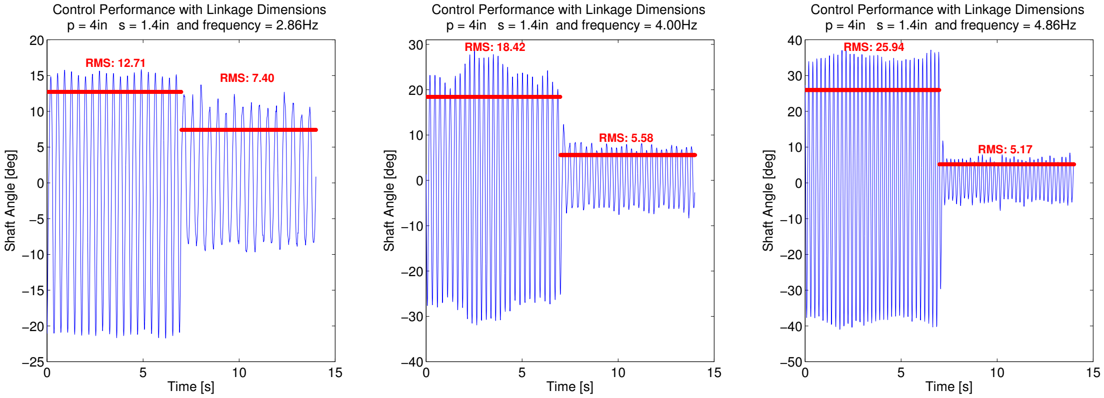


Figure 4.17: Results of robustness test for the tremor attenuation system with linkages in the mechanical test rig set to $p = 4.0$in, $s = 1.4$in. The individual plots show different tremor frequencies with this test configuration. From left to right, the plots depict a tremor attenuation of 41.79%, 69.69%, and 80.08%.

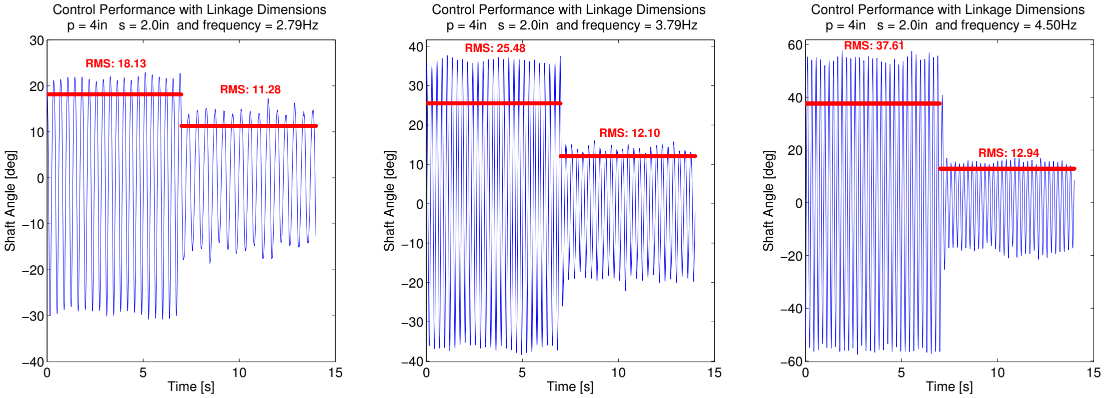


Figure 4.18: Results of robustness test for the tremor attenuation system with linkages in the mechanical test rig set to $p = 4.0$in, $s = 2.0$in. The individual plots show different tremor frequencies with this test configuration. From left to right, the plots depict a tremor attenuation of 37.75%, 52.50%, and 65.59%.

Figures 4.16-4.18 qualitatively indicate the device is fairly robust to tremor variations. In order to quantify the effectiveness of the system, the RMS value is calculated during the periods when the device is on and off for each trial. Then, the percent reduction in tremor is calculated. These values are tabulated in Table 4.5. On average, the device is able to reduce the tremor amplitude by 60.96%. This is satisfies the target set by the design specifications. The analysis also reveals that the device generally performs better when the tremor is at a higher frequency. Future work should consider adjusting the control algorithm to better handle lower frequency tremor.

Link p [in]	Link s [in]	Frequency [Hz]	% Attenuation
4.0	0.8	3.29	56.52
4.0	0.8	4.43	70.12
4.0	0.8	5.79	74.62
4.0	1.4	2.86	41.79
4.0	1.4	4.0	69.69
4.0	1.4	4.86	**80.08**
4.0	2.0	2.79	37.76
4.0	2.0	3.79	52.5
4.0	2.0	4.5	65.59
		average	**60.96**

Table 4.5: Tremor reduction across a number of different tremor amplitudes and frequencies. On average, the device is able to reduce the tremor amplitude by 60.96%, this satisfies the target set by the design specifications. The maximum attenuation was 80.08%, which occurred at the same settings used during the tuning of the control algorithm.

4.6 Conclusions

This chapter provides insight into the use of weighted reaction wheels for the stabilization of tremor. The prototype wristband is lighter, smaller, and more aesthetic than any existing exoskeletons. It also is sufficiently effective at attenuating tremor. The current design is far from optimized but still able to attenuate tremor by an average of 60.96%. This early-stage result alone would merit serious consideration toward designing a commercial device. In reality, however, many enhancements could be made to the control algorithm and the design of the reaction wheel. It is likely that further iterations of the device would be considerably more effective.

While the reaction wheel system is the primary contribution of this chapter, the mechanical test rig is equally relevant. The rig proved to be a useful and robust tool for testing the tremor suppression device. It allowed the system to be quantifiably evaluated in a safe and repeatable environment. Even if future researchers choose to pursue alternate methods of tremor suppression, the mechanical test rig can be used as a testing platform.

Chapter 5

Conclusions

Millions of people around the world suffer from Parkinson's Disease, Essential Tremor, or other tremor causing conditions. These diseases are chronic and can become debilitating as they progress. Despite decades of research, the medical standards for the diagnosis and treatment of these diseases remain severely inadequate. Diagnostic methods are qualitative rather than quantitative, and treatments are often accompanied by burdensome side effects. This thesis applies state of the art engineering practices toward the mission of improving medical care for chronic tremor patients.

The first hurdle facing a patient with chronic tremor is obtaining a diagnosis. Medical professionals currently base this diagnosis on qualitative observations made during a physical examination as well as a patient's history. The process is prone to error, and is especially imprecise during the critical early stages of disease onset. To improve the quality of care during this aspect of the patient experience, Chapter 2 of this work develops a prototype medical device that can accurately record the motion characteristics of upper-limb tremor. The device contains a number of embedded sensors mounted to a glove and wristband made from light-weight fabric.

The effectiveness of the device was tested with the help of 30 healthy volunteers. Participants were educated on the motion characteristics of PD and ET and asked to mimic the diseases to the best of their ability. A number of machine learning algorithms were then used to attempt automated diagnoses of the two diseases. The most successful of these algorithms achieves a 99.9% classification accuracy against an exhaustive cross-validation of the validation data. At the time of writing, a trial evaluating the clinical efficacy of the device is underway. If this result can be replicated in a clinical setting, it will represent a great improvement in diagnostic standards.

The third chapter of this thesis focuses on the development of a new algorithm for attitude estimation of the human wrist. Attitude estimation of aircraft and spacecraft is a common, and well-researched field. As such, several algorithms exist that have been optimized to perform very well in aerospace applications. The Kalman filter developed in this work builds off these prior efforts but makes several domain specific improvements. By leveraging knowledge about human anatomy and standard movement patterns of the forearm, the filter achieves reliable, accurate results even while tracking high-frequency tremor. During experimental validation, the algorithm achieved an RMS error of less than 1 degree even when the motion of the object being tracked exceeded 260 degrees per second. This algorithm is a core technical contribution of the thesis.

The final component of this work begins to address shortcomings in the treatment of Parkinsonian tremor. After a PD diagnosis is reached, patients are usually prescribed a dopaminergic drug to counter the symptoms. Even the most time-tested of these drugs carry side effects such as hallucinations, dizziness, and nausea. This work develops a prototype device that can aide, and potentially reduce the need for, dopaminergic drugs. The device is a self-contained wristband that uses a weighted reaction wheel to stabilize the forearm. When testing the device with a mechanical test rig, it was able to attenuate tremor by an average of 60.96%. Although the current prototype is not suitable for patient use, it is an effective proof-of-concept and shows promise in the development of a low profile mechanical exoskeleton for the stabilization of tremor.

When assembled, the three aspects of this work lay the groundwork for a great improvement in the quality of medical care for tremor patients. With additional research, the prototypes have the potential to positively impact millions of lives. While the current devices are limited prototypes, their success sheds light on the gap between current standards of care and the capabilities of modern engineering. Most importantly, however, they show that the challenges facing chronic tremor patients merit further inquiry from the engineering community. With time and effort, the outcomes for PD and ET patients could be dramatically improved.

Chapter 6

Future Work

Each chapter of this thesis has a natural continuation that should be further explored. In Chapter 2, a tremor classification device was developed. The prototype hardware is sufficient for preliminary clinical testing, but many improvements can be made. The portion of the device that fits around a user's upper arm includes a pre-made development board and microcontroller shield that is much larger than necessary for the application. A simple improvement to the device would be to develop a custom circuit board containing the microcontroller, data logger, and exposed pins for all external communication with the sensors. This custom board could be a fraction of the current size and might even eliminate the need for the upper-arm band. All components could potentially fit on the wristband and glove. Conducting thorough human factors studies could also lead to improvements in both the comfort and aesthetics of the device.

The algorithms used to perform the tremor classification also have room for further development. There are many classification algorithms that have already been developed and tested by the machine learning community. This thesis tested a few of these methods (both parametric and non-parametric). The results indicate that high classification accuracy is achievable with the data, but a more exhaustive exploration of various classification strategies could yet lead to improved results.

In Chapter 3, an algorithm was developed to perform attitude estimation of the human wrist. This algorithm was tested using a simulation in Matlab, as well as with a mechanical test rig. The next step in this research would be to test the algorithm on a human wrist. To accomplish this, a volunteer would need to wear the sensor system developed in Chapter 2 in order to collect data. The user would also need an external reference to provide the true attitude of his or her wrist, which could be accomplished with a camera based tracking system (such as a Vicon). Conducting an experiment in this manner would further validate the effectiveness of the attitude estimation algorithm. Another avenue of future research could be to apply the algorithm to other limbs. The aspects of the filter that make it specific to the human wrist can be easily modified to accommodate other geometries, allowing the filter to be used for a wider range of applications.

Chapter 4 explores the potential of a tremor suppression device. The logical next step in this research is to refine the physical design so the device could be comfortably worn by a healthy volunteer. Doing so would allow the device to be tested in a more realistic setting. If successful, such a test would allow the device to be evaluated for its clinical effectiveness on diagnosed patients.

Another area of exploration within this application is the analysis and development of more complex control algorithms. The control architecture used in this work was only intended as a proof of concept. More advanced algorithms would likely produce better results and should be investigated. The weight of the reaction wheel, as well as the size and quantity of the motors will also have an impact on the effectiveness of the control algorithm and are parameters that should be explored further. Lastly, many of the assumptions made while designing the devices and algorithms are based on insights from clinicians. With actual data from clinical trials, these designs should be revisited and further refined.

Appendix A

Update on Clinical Trial

During the writing of this thesis, a clinical trial was initiated to evaluate the efficacy of the diagnostic system developed in Chapter 2. The trial officially commenced less than a week before the submission of this work. This appendix presents a brief description of the clinical data collected to date. The primary objective of the trial is to collect data that will aid in refining the diagnostic algorithm developed in Chapter 2.

Patients were recruited who had been clinically diagnosed with PD or ET. After obtaining informed consent, the study coordinators assisted each participant in putting the device on the hand/arm with the most pronounced tremor. Patients were then asked the following questions, and responses were recorded.

1. "On a scale from zero to 10 how much discomfort does the device cause? Zero means no discomfort whatsoever, 10 means maximum discomfort."
2. "What about the device is causing you discomfort?" [This question will only be asked if the patient answered more than zero discomfort].

Next, each patient was asked to perform the following actions for 20 seconds.

1. Letting the arm sit at rest while supported by a table.

2. Holding the arm in an outstretched position in front of the patient.
3. Performing a tap test (tapping index finger and thumb together)
4. Performing a finger-to-nose test (repeatedly moving the finger from an outstretched position to the tip of the nose)

At the time of submission, three patients had participated in the study. The clinical diagnoses of these patients are listed in Table A.1. Each of the patients performed all 4 actions outlined above. As described in Chapter 2, the sensor system has 4 IMUs, and each of these sensors records 6 measurements every time it is sampled. Plotting all of the raw data would result in an impractical amount of graphs. Instead, a preliminary visualization of the data can be obtained by plotting 1 sensor of the system, during 1 action of the trial, for each patient. These plots can be seen in Figs. A.1-A.6. In these figures, the IMU on the back of the patient's wrist is plotted during action 1 (letting the arm sit at rest while fully supported).

These plots reveal some interesting preliminary trends. First, all three patients suffered from tremors at approximately 10Hz, regardless of their diagnosis. Based on existing literature [11], this frequency is higher than expected for the PD patients, but is within the expected range for ET patients.

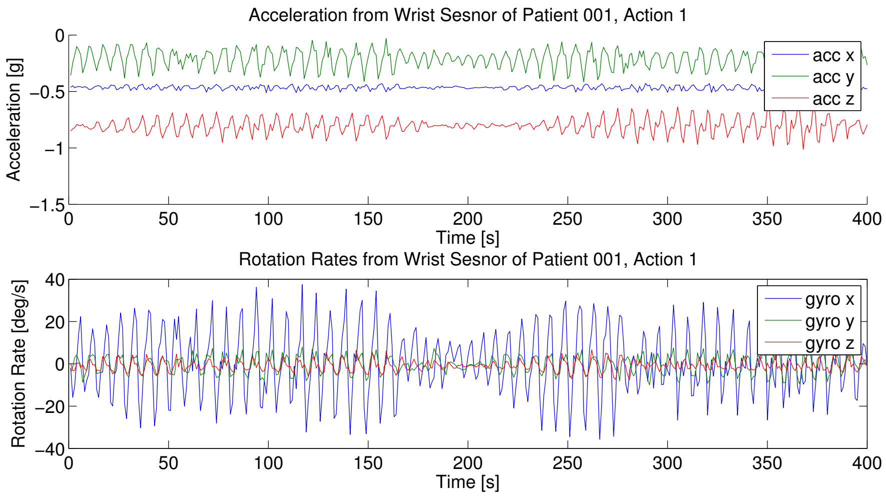
Another potentially meaningful trend can be seen by noting which measurements exhibit the largest magnitudes in the FFT plot. In the data from both PD patients, the x-axis gyroscope measurements recorded tremors of substantially larger amplitude than any other axes (seen in Figs. A.2 and A.6). In the case of the ET patient however, the x-axis gyroscope measurements and the y-axis gyroscope measurements recorded tremors of similar amplitudes.

This insight could lead to an improved diagnostic algorithm if the trend is supported by future study participants.

While the analysis is currently limited by the small sample size, these observations show that additional data may reveal exciting trends. It should also be noted that two of the three participants reported ”0” discomfort while wearing the device, while the final participant reported a discomfort level of ”1”. Ultimately, this preliminary inspection of the raw data shows that the sensing device is on a promising trajectory.

	Clinical Diagnosis
Patient 001	Parkinson’s Disease
Patient 002	Essential Tremor
Patient 003	Parkinson’s Disease

Table A.1: Clinical diagnoses of the patients who had participated in the trial at the time of writing. These patients are the first of many who will eventually participate in the study.


Acceleration from Wrist Sesnor of Patient 001, Action 1

Rotation Rates from Wrist Sesnor of Patient 001, Action 1

Figure A.1: Accelerometer and gyroscope data from Patient 001. This data was collected while the patient’s arm was at rest, and fully supported by a table. The x-axis gyroscope measurements recorded tremor of substantially larger amplitude than any other axes. This is likely due to the pronation and supination that typically presents with PD.

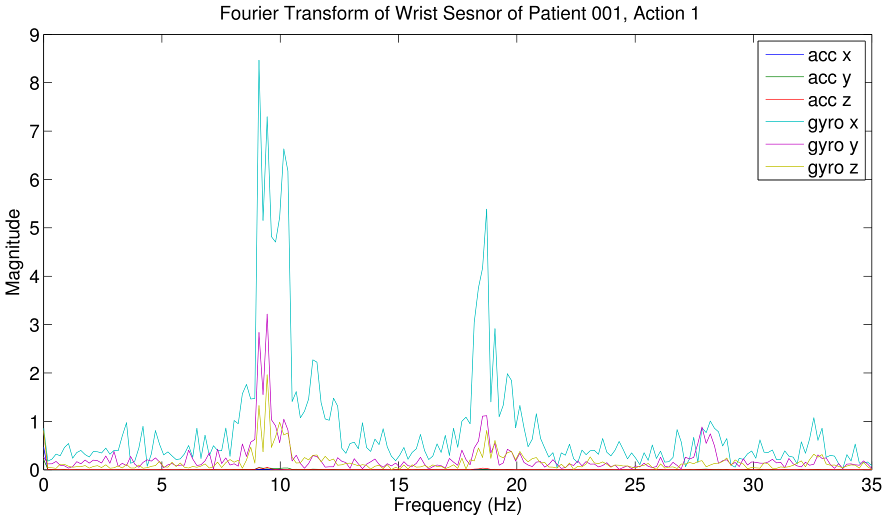


Figure A.2: FFT of the IMU data from Patient 001. This data was collected while the patient's arm was at rest, and fully supported by a table. The x-axis gyroscope measurements recorded tremor of substantially larger amplitude than any other axes. This is likely due to the pronation and supination that typically presents with PD.

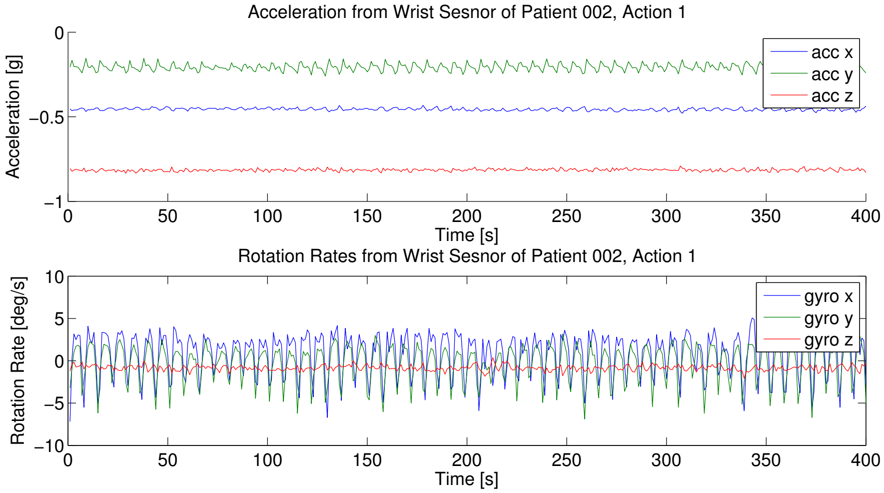


Figure A.3: Accelerometer and gyroscope data from Patient 002. This data was collected while the patient's arm was at rest, and fully supported by a table. The x-axis gyroscope and the y-axis gyroscope recorded tremor of approximately equal amplitude. This is different than the observations made for the current PD patients, and may lead to a diagnostically relevant trend.

Figure A.4: FFT of the IMU data from Patient 002. This data was collected while the patient's arm was at rest, and fully supported by a table. The x-axis gyroscope and the y-axis gyroscope recorded tremor of approximately equal amplitude. This is different than the observations made for the current PD patients, and may lead to a diagnostically relevant trend.

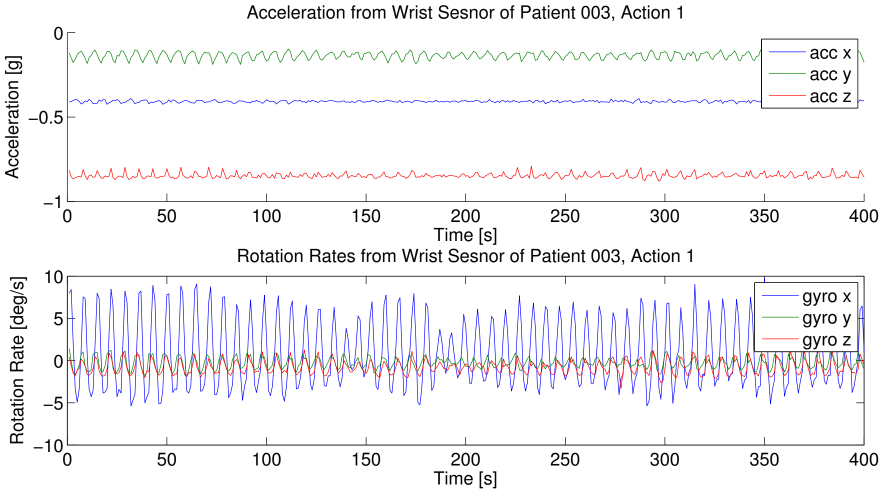


Figure A.5: Accelerometer and gyroscope data from Patient 003. This data was collected while the patient's arm was at rest, and fully supported by a table. The x-axis gyroscope measurements recorded tremor of substantially larger amplitude than any other axes. This is likely due to the pronation and supination that typically presents with PD.

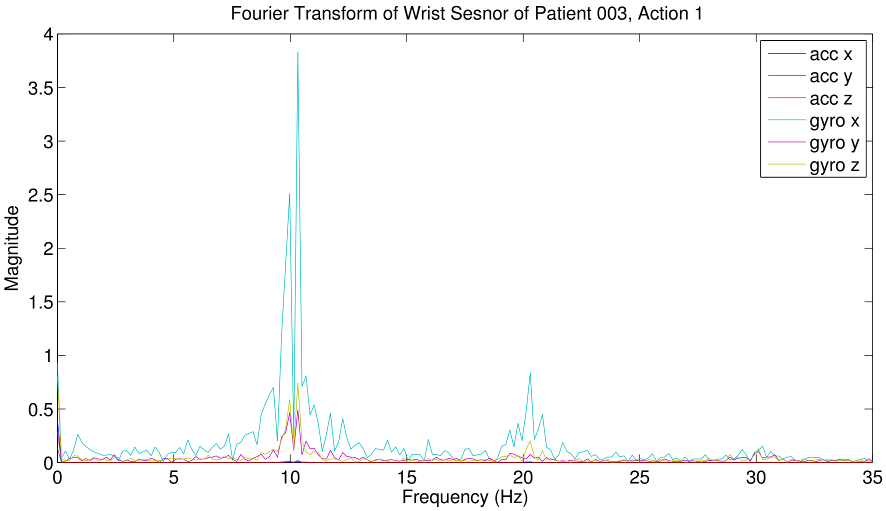


Figure A.6: FFT of the IMU data from Patient 003. This data was collected while the patient's arm was at rest, and fully supported by a table. The x-axis gyroscope measurements recorded tremor of substantially larger amplitude than any other axes. This is likely due to the pronation and supination that typically presents with PD.

Bibliography

[1] 2017. URL: http://www.pdf.org/parkinson_statistics.

[2] Anwar Ahmed and Patrick Sweeney. *Tremors*. 2014. URL: http://www.clevelandclinicmeded.com/medicalpubs/diseasemanagement/neurology/tremors/.

[3] Ethem Alpaydin. *Introduction to Machine Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, 2009. URL: https://www.amazon.com/Introduction-Machine-Learning-Adaptive-Computation-ebook/dp/B008H5Q5OU?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B008H5Q5OU.

[4] Sylvain Arlot and Alain Celisse. "A survey of cross-validation procedures for model selection". In: *Statistics Surveys* 4 (2010), pp. 40–79. ISSN: 1935-7516. DOI: 10.1214/09-SS054. arXiv: 0907.4728.

[5] Paolo Bonato et al. "Data mining techniques to detect motor fluctuations in Parkinson's disease." In: *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference* 7 (2004), pp. 4766–9. ISSN: 1557-170X. DOI: 10.1109/IEMBS.2004.1404319. URL: http://www.ncbi.nlm.nih.gov/pubmed/17271375.

[6] Brian D Bue and James M Killian. "Classification and Diagnosis of Myopathy from EMG Signals *". In: May (2013).

[7] J Carryer. *Introduction to mechatronic design*. Upper Saddle River: Prentice Hall, 2011. ISBN: 0131433563.

[8] R Contreras et al. "Tremors Quantification in Parkinson Patients Using Smartwatches". In: (). DOI: `10.1109/ETCM.2016.7750866`.

[9] John L Crassidis and F Landis Markley. "Unscented filtering for spacecraft attitude estimation". In: *Journal of guidance, control, and dynamics* 26.4 (2003), pp. 536–542.

[10] John L. Crassidis, F. Landis Markley, and Yang Cheng. "Survey of Nonlinear Attitude Estimation Methods". In: *Journal of Guidance, Control, and Dynamics* 30.1 (2007), pp. 12–28. ISSN: 0731-5090. DOI: `10.2514/1.22452`.

[11] P. Crawford and E.E. E Zimmerman. "Differentiation and Diagnosis of Tremor". In: *American family physician* 83.6 (2011), pp. 697–702. ISSN: 0002-838X. DOI: `10.1016/S0002-838X(11)60093-2`. URL: `http://ovidsp.ovid.com/ovidweb.cgi?T=JS{\&}CSC=Y{\&}NEWS=N{\&}PAGE=fulltext{\&}D=emed10{\&}AN=2011286180`.

[12] Gelb DJ, Oliver E, and Gilman S. "Diagnostic criteria for parkinson disease". In: *Archives of Neurology* 56.1 (1999), pp. 33–39. DOI: `10.1001/archneur.56.1.33`. eprint: `/data/journals/neur/6394/nsa7701.pdf`. URL: `+http://dx.doi.org/10.1001/archneur.56.1.33`.

[13] Strahinja Dosen et al. "Online Tremor Suppression Using Electromyography and Low Level Electrical Stimulation." In: *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*

4320.c (2014), pp. 1–11. ISSN: 1558-0210. DOI: 10.1109/TNSRE.2014.2328296. URL: http://www.ncbi.nlm.nih.gov/pubmed/25051555.

[14] Gabriel Hugh Elkaim and Demoz Gebre-egziabher. “Attitude Estimation for”. In: *Multisensor Attitude Estimation*. Ed. by Fourati Hassen et al. CRC Press, 2016. Chap. 19, pp. 331–350. ISBN: 978-1-4987-4571-0. URL: https://www.amazon.com/Multisensor-Attitude-Estimation-Fundamental-Applications-ebook/dp/B01N3B5ZNB?SubscriptionId=0JYN1NVW651KCA56C102{\&}tag=techkie-20{\&}linkCode=xm2{\&}camp=2025{\&}creative=165953{\&}creativeASIN=B01N3B5ZNB.

[15] T. Flash et al. “Kinematic analysis of upper limb trajectories in Parkinson’s disease”. In: *Experimental Neurology* 118.2 (1992), pp. 215–226. ISSN: 00144886. DOI: 10.1016/0014-4886(92)90038-R.

[16] Weaver FM et al. “Bilateral deep brain stimulation vs best medical therapy for patients with advanced parkinson disease: A randomized controlled trial”. In: *JAMA* 301.1 (2009), pp. 63–73. DOI: 10.1001/jama.2008.929. eprint: /data/journals/jama/4446/joc80124_63_73.pdf. URL: +http://dx.doi.org/10.1001/jama.2008.929.

[17] Gene F. Franklin, J. Da Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems (7th Edition)*. Pearson, 2014. ISBN: 0133496597. URL: https://www.amazon.com/Feedback-Control-Dynamic-Systems-7th/dp/0133496597?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0133496597.

[18] James Hall, Nathan Knoebel, and Timothy McLain. "Quaternion attitude estimation for miniature air vehicles using a multiplicative extended kalman filter". In: *Record - IEEE PLANS, Position Location and Navigation Symposium* (2008), pp. 1230–1237. DOI: `10.1109/PLANS.2008.4570043`.

[19] Gali Heimer et al. "Dopamine replacement therapy does not restore the full spectrum of normal pallidal activity in the 1-methyl-4-phenyl-1,2,3,6-tetra-hydropyridine primate model of Parkinsonism". In: *Journal of Neuroscience* 26.31 (2006), pp. 8101–8114. ISSN: 1529-2401. DOI: `10.1523/JNEUROSCI.5140-05.2006`. URL: `http://www.ncbi.nlm.nih.gov/pubmed/16885224`.

[20] Kazuo Kiguchi and Yoshiaki Hayashi. "Upper-limb tremor suppression with a 7DOF exoskeleton power-assist robot". In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (2013), pp. 6679–6682. ISSN: 1557170X. DOI: `10.1109/EMBC.2013.6611088`.

[21] Baoli Li and Liping Han. "Distance Weighted Cosine Similarity Measure for Text Classification". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2013: 14th International Conference, IDEAL 2013, Hefei, China, October 20-23, 2013. Proceedings*. Ed. by Hujun Yin et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 611–618. ISBN: 978-3-642-41278-3. DOI: `10.1007/978-3-642-41278-3_74`. URL: `https://doi.org/10.1007/978-3-642-41278-3_74`.

[22] Paweł Maciejasz et al. "A survey on robotic devices for upper limb rehabilitation." In: *Journal of neuroengineering and rehabilitation* 11.1 (2014), p. 3. ISSN: 1743-0003. DOI:

`10.1186/1743-0003-11-3`. URL: `http://www.jneuroengrehab.com/content/11/1/3`.

[23] Lana Popovic Maneski et al. “Electrical stimulation for the suppression of pathological tremor”. In: *Medical and Biological Engineering and Computing* 49.10 (2011), pp. 1187–1193. ISSN: 01400118. DOI: `10.1007/s11517-011-0803-6`.

[24] Emfforce O P S Manual. “Reaction Wheel Design”. In: (2003), pp. 1–11.

[25] Timo Von Marcard. “Design and implementation of an attitude estimation system to control orthopedic components”. In: *Thesis* (2010).

[26] F. Landis Markley. “Multiplicative vs . Additive Filtering for Spacecraft Attitude Determination Quaternion estimation”. In: *Journal of Guidance, Control, and Dynamics* 26.2 (2003), pp. 311–317. URL: `https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20040037784.pdf`.

[27] Wigand Poppendieck et al. “Multi-channel EMG recording and muscle stimulation electrodes for diagnosis and treatment of tremor”. In: *2014 IEEE 19th International Functional Electrical Stimulation Society Annual Conference, IFESS 2014 - Conference Proceedings* (2014), pp. 14–17. DOI: `10.1109/IFESS.2014.7036739`.

[28] E. Rocon et al. “Design and validation of a rehabilitation robotic exoskeleton for tremor assessment and suppression”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15.1 (2007), pp. 367–378. ISSN: 15344320. DOI: `10.1109/TNSRE.2007.903917`.

[29] Malcolm D Shuster. “A survey of attitude representations”. In: *Navigation* 8.9 (), pp. 439–517.

[30] Robert F. Stengel. *Optimal Control and Estimation (Dover Books on Mathematics)*. Dover Publications, 2012. ISBN: 0486682005. URL: `https://www.amazon.com/Optimal-Control-Estimation-Dover-Mathematics-ebook/dp/B00A3M0ZNW?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B00A3M0ZNW`.

[31] Bing Sun, Yang Wang, and Jacob Banda. “Gait characteristic analysis and identification based on the iPhone’s accelerometer and gyrometer”. In: *Sensors (Switzerland)* 14.9 (2014), pp. 17037–17054. ISSN: 14248220. DOI: `10.3390/s140917037`.

[32] Xiao-Hong Wang et al. “Target Selection Recommendations Based on Impact of Deep Brain Stimulation Surgeries on Nonmotor Symptoms of Parkinson’s Disease.” In: *Chinese medical journal* 128.24 (2015), pp. 3371–80. ISSN: 0366-6999. DOI: `10.4103/0366-6999.171464`. URL: `http://www.ncbi.nlm.nih.gov/pubmed/26668154{\%}5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4797515`.

[33] F. Widjaja et al. “Towards a sensing system for quantification of pathological tremor”. In: *2007 International Conference on Intelligent and Advanced Systems, ICIAS 2007* (2007), pp. 986–991. DOI: `10.1109/ICIAS.2007.4658533`.

[34] Ferdinan Widjaja et al. “Filtering of intended motion for real-time tremor compensation in human upper limb using surface electromyography”. In: *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology So-*

ciety: Engineering the Future of Biomedicine, EMBC 2009 (2009), pp. 2996–2999. ISSN: 1557-170X. DOI: `10.1109/IEMBS.2009.5332523`.

[35] Xiaoping Yun and Eric R Bachmann. “Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking”. In: *IEEE Transactionson Robotics* 22.6 (2006), pp. 1216–1227.

[36] Rong Zhu and Zhaoying Zhou. “A real-time articulated human motion tracking using triaxis inertial/magnetic sensors package”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 12.2 (2004), pp. 295–302. ISSN: 15344320. DOI: `10.1109/TNSRE.2004.827825`.