6-22-2015

Sensor Fusion for Effective Hand Motion Detection

Fatemeh Abyarjoo

Florida International University, fabya001@fiu.edu

DOI: 10.25148/etd.FIDC000084

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>

Part of the [Signal Processing Commons](#)

Recommended Citation

Abyarjoo, Fatemeh, "Sensor Fusion for Effective Hand Motion Detection" (2015). *FIU Electronic Theses and Dissertations*. 2215.
<https://digitalcommons.fiu.edu/etd/2215>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

SENSOR FUSION FOR EFFECTIVE HAND MOTION DETECTION

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Fatemeh Abyarjoo

2015

To: Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Fatemeh Abyarjoo, and entitled Sensor Fusion for Effective Hand Motion Detection, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Jean H. Andrean

Malek Adjouadi

Ismail Guvenc

Chenzhong Li

Armando Barreto, Major Professor

Date of Defense: May 11, 2015

The dissertation of Fatemeh Abyarjoo is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Dean Lakshmi N. Reddi
University Graduate School

Florida International University, 2015

© Copyright 2015 by Fatemeh Abyarjoo

All rights reserved.

DEDICATION

To my parents, sisters, grandparents, Sofia and Edi.

ACKNOWLEDGMENTS

I would like to express the most sincere gratitude for my major Professor, Dr. Armando Barreto, for his guidance, patience and support in navigating the research process. Without his knowledge and guidance completion of this endeavor would never have been possible.

I also would like to recognize my committee members Drs. Malek Adjouadi, Jean Andrian and Ismail Guvenc and Chenzhong Li for their guidance and expertise. I am specially grateful to Dr. Naphtali Rishe and Dr.Keng Yen for their constant support. I need to thank Dr. Masoud Sadjadi for challenging my work and providing professional guidances.

This dissertation was supported by National Science Foundation grants HRD-0833093, and CNS-0959985, Florida International University Graduate School Dissertation Year Fellowship and National Science Foundation PIRE program. I would like to thank NSF-FIU I/UCRC CAKE Center staff, FIU graduate school staff and PIRE program staff specially Dr. Alvarez and Vasilka Chergarova.

I would like to thank my parents, grandparents and my sisters Somayeh and Parisa. I especially have to thank my little sister Bahareh Abyarjoo who, even though was busy with her own dissertation, helped me with all graphical preparation for this dissertation.

Over the last few years, I have been fortunate to be supported by official/ unofficial DSP Lab members including Francisco Ortega, Jonathan Cofino, Peng Reng, Jian Huang, Luz Adriana, Daisy Lu, Nonnarit O-Larnnithipong, Sudarat Tangnimitchok and Panuwat Janwattanapong . Their friendship, encouragements, expertise and guidances facilitated challenges that I encountered during my PhD program. I need to thank FIU Electrical Department staff, specially Pat Brammer, Oscar Silveira, Ana Saenz, Catherine Olnick and Mais Kayyali.

I would like to express my gratitude to all my beloved friends for their encouragement and supports. I should name Lida, Shaghayegh, Setareh, Maryam, Sharon and Kristin and their families, Farzaneh, Fereshteh khanom and agha Mazy. I have to also thank my special friends in 208 gang including Elaheh, Tara, Mahbubeh, Maryam, Naeemeh and Tayebah for their encouragements and for being supportive unconditional friends.

ABSTRACT OF THE DISSERTATION
SENSOR FUSION FOR EFFECTIVE HAND MOTION DETECTION

by

Fatemeh Abyarjoo

Florida International University, 2015

Miami, Florida

Professor Armando Barreto, Major Professor

Human hands have a critical role in many human activities. There are many applications in the field of human computer interaction where it is necessary to monitor human hand motions. Different technologies have been employed to track the human hand motion. However, many of those approaches did not work well in real life situations. This research focused on tracking the human hand motion so that it can be used regardless of location or environmental lightening. MEMS (Micro-Electro-Mechanical sensors) inertial and magnetic sensors were used in this study for tracking the palm and the thumb motions.

In this dissertation a sensor fusion algorithm was proposed to track the three-dimensional rotational motion of the human hand. The proposed sensor fusion algorithm was a quaternion-based Kalman filter. Two optimization methods (Gradient Descent and Newton-Gauss) were utilized to estimate the corresponding quaternion vector using the observation vector in the Kalman filtering process.

Two sensor units were attached on a glove on the palm and thumb sections to record the motion related data. As the articulation of the remaining four fingers (index, middle, ring and pinky) to the palm are more restricted than the thumb, the applied approach for the thumb can be applicable for other fingers.

It is challenging to compute the rotational motion using MESMS sensors with acceptable accuracy because the MEMS sensors are prone to different types of sys-

tematic and stochastic errors. To calculate reliable results from these sensors, it was necessary to compensate for all types of error. The calibration process was performed to compensate for systematic errors. The sensors stochastic errors were compensated in the sensor fusion process.

To evaluate the system performance, three experiments were carried out. Statistical analysis was performed on the experimental data. Statistical analysis showed that orientation provided by the sensor fusion algorithm is accurate and reliable. It was also observed that both optimization techniques performed similarly, although the Newton-Gauss approach converged faster. Additional results are found in this dissertation.

TABLE OF CONTENTS

CHAPTER	PAGE
1. INTRODUCTION	1
1.1 Problem Statement	1
1.2 Objective of the Research	1
1.3 Motivation	2
1.4 Significance of this Research	2
1.5 Literature Review	3
1.5.1 Wearable Sensors Based on Inertial/Magnetic Measurements	4
1.5.2 Hand Motion Detection Technologies	10
1.6 Structure of the Dissertation	12
2. THREE DIMENSIONAL ROTATION	13
2.1 Introduction	13
2.2 Three Dimensional Rotation	13
2.3 Coordinate Systems	14
2.3.1 Inertial Frame	15
2.3.2 Body-fixed Frame	15
2.4 Euler Angles	15
2.4.1 Rotation Matrices	16
2.4.2 Gimbal Lock	18
2.5 Quaternions	19
2.5.1 What Are Quaternions	20
2.5.2 Quaternion Rotation	25
3. MEMS INERTIAL AND MAGNETIC SENSORS	27
3.1 Introduction	27
3.2 Inertial sensors	27
3.2.1 Accelerometers	28
3.2.2 Gyroscopes	30
3.3 MEMS Inertial Sensor Errors	31
3.3.1 Angle Random Walk	32
3.3.2 Rate Random Walk	33
3.3.3 Flicker Noise	33
3.3.4 Quantization Noise	33
3.3.5 Sinusoidal Noise	33
3.3.6 Bias Error	34
3.3.7 Scale Factor Error	34
3.3.8 Scale Factor Sign Asymmetry Error	35
3.3.9 Misalignment (Cross-coupling) Error	35
3.3.10 Non-linearity Error	36
3.3.11 Dead Zone Error	36

3.3.12	Temperature Effect	36
3.4	Magnetometers	37
3.5	MEMS Magnetometer Errors	38
4.	SENSOR FUSION ALGORITHMS	40
4.1	Introduction	40
4.2	Sensor Fusion	40
4.3	Least Squares Estimator	41
4.4	Kalman Filter	43
4.5	Discrete Kalman Filter	44
4.6	Quaternion-based Kalman Filter	50
4.6.1	Prediction Step	51
4.6.2	Correction Step	53
4.6.3	Observation Vector Using Gradient Descent Optimization	55
4.6.4	Observation Vector Determination Using Gauss-Newton Method	58
5.	FILTER IMPLEMENTATION	61
5.1	Introduction	61
5.2	Human Hand Movements	61
5.2.1	Human Wrist Anatomy and Movements	61
5.2.2	Human thumb Anatomy and Movements	65
5.3	System Description	68
5.4	Data Acquisition	70
5.5	Calibration Process	72
5.6	Test of Gauss-Newton Algorithm	84
5.7	Test of Gradient Descent Algorithm	90
5.8	Kalman Filter Implementation	94
6.	EXPERIMENTAL RESULTS	107
6.1	Introduction	107
6.2	Pool of Experimental Subjects	107
6.3	Experiment Procedure	108
6.4	Experiment 1: Performance Test and Accuracy Comparison	109
6.4.1	Quaternion-based Kalman filter with Gradient Descent optimization method-Results	110
6.4.2	Quantitative Analysis for Approach Using Gradient Descent	110
6.4.3	Quaternion-based Kalman filter with Gauss-Newton optimization method- Results	117
6.4.4	Quantitative Analysis for the Approach Using Gauss-Newton Opti- mization	118
6.4.5	Statistic Comparison for Gauss-Newton and Gradient Descent	124
6.5	Experiment 2: Movement Classification	127
6.5.1	Data Analysis for Experiment 2	131

6.6	Experiment 3: Thumb Motion Repetition	132
6.6.1	Data Analysis for Experiment 3	132
6.7	Questionnaires	132
6.8	Hand Motion Visualization	135
7.	CONCLUSION AND FUTURE WORK	148
7.1	Conclusions	148
7.2	Future Work	149
	BIBLIOGRAPHY	157
	VITA	167

LIST OF FIGURES

FIGURE	PAGE
2.1 Three Dimensional Reference Frame	14
2.2 Roll	14
2.3 Pitch	14
2.4 Yaw	14
2.5 Inertial and Body-fixed Reference Frames	15
2.6 Three Gimbals	19
2.7 Gimbal Lock Effect	20
3.1 Basic Accelerometer	29
3.2 Accelerometer Response to the External Forces	30
3.3 Bias Error	34
3.4 Scale Factor Error	35
3.5 Scale Factor Sign Asymmetry Error	36
3.6 Axis Misalignment	37
3.7 Non-linearity Error	38
3.8 Dead Zone Error	39
3.9 Geographic and Magnetic Poles	39
4.1 Combining the Sensors Output	41
4.2 Kalman Filter Cycle	44
4.3 Discrete Kalman Filter Flow Chart	49
4.4 Hand and Inertial Reference Frames	50
4.5 Discrete Kalman Filter Block Diagram	54
5.1 Hand Pronation Motion	62
5.2 Hand Supination Motion	63
5.3 Hand Extension Motion	63

5.4	Hand Flexion Motion	64
5.5	Hand Radial Motion	64
5.6	Hand Ulnar Motion	65
5.7	Human Thumb Bones	66
5.8	Thumb Flexion Motion	66
5.9	Thumb Extension Motion	67
5.10	Thumb Abduct Motion	67
5.11	Thumb Adduct Motion	68
5.12	Sensor Unit	69
5.13	Raw Sensor Data Example	72
5.14	Calibrated Gyroscope Data	73
5.15	Rotational Angles calculated by Gyroscope Data	74
5.16	Calibrated Accelerometer Data	75
5.17	Rotational Angles Using Accelerometer Data	76
5.18	Calibrated Magnetometer Data	77
5.19	Earth Magntic Vector	78
5.20	Unaffected Magnetometer	79
5.21	Hard Iron Effect on Magnetometer	79
5.22	Soft Iron Effect on Magnetometer	80
5.23	Soft Iron Effect on Magnetometer	81
5.24	X axis with respect to Y axis Uncompensated Magnetometer Data . . .	81
5.25	Soft and Hard Iron Effect on Magnetometer Data	82
5.26	Soft and Hard Iron Error Compensated 2D Magnetometer Data	82
5.27	Sphere-Fitted Calibrated Magnetometer Data	83
5.28	Three Axis Calibrated Magnetometer Data	83
5.29	Newton-Gauss Algorithm Convergence	88

5.30	Iteration Convergence Number for Different Noises	89
5.31	Gradient descent Algorithm Convergence	93
5.32	Iteration Convergence Number for Different Noises	94
5.33	Trace of Error Covariance Matrix	99
5.34	Smoothing the Filter Output	101
5.35	Kalman Filter Output- Trial 1	101
5.36	Kalman Filter Output- Trial 2	102
5.37	Kalman Filter Output- Trial 3	102
5.38	Kalman Filter Output- Trial 4	103
5.39	Kalman Filter Output- Trial 5	103
5.40	Kalman Filter Output- Trial 6	104
5.41	Gauss-Newton and Gradient Descent Comparison (q1)	104
5.42	Gauss-Newton and Gradient Descent Comparison (q2)	105
5.43	Gauss-Newton and Gradient Descent Comparison (q3)	105
5.44	Gauss-Newton and Gradient Descent Comparison (q4)	106
6.1	Volunteer Training with the Instrumented Glove	108
6.2	Experiment 1	109
6.3	Histogram for 20 Degree Roll Rotation Using Gradient Descent Method	113
6.4	Histogram for 20 Degree Pitch Rotation Using Gradient Descent Method	114
6.5	Histogram for 20 Degree Yaw Rotation Using Gradient Descent Method	114
6.6	QQ plot for Roll Data Using Gradient Descent Method	115
6.7	QQ plot for Pitch Data Using Gradient Descent Method	116
6.8	QQ plot for Yaw Data Using Gradient Descent Method	116
6.9	Histogram for 20 Degree Roll Rotation Using Gauss-Newton Method . .	120
6.10	Histogram for 20 Degree Pitch Rotation Using Gauss-Newton Method .	121
6.11	Histogram for 20 Degree Yaw Rotation Using Gauss-Newton Method . .	121

6.12	QQ plot for Roll Data Using Gauss-Newton Method	122
6.13	QQ plot for Pitch Data Using Gauss-Newton Method	122
6.14	QQ plot for Yaw Data Using Gauss-Newton Method	123
6.15	Box Plot for Comparison between Optimization Methods	124
6.16	QQ plot for Experimental Data Using Gauss-Newton Method	125
6.17	QQ plot for Experimental Data Using Gauss-Newton Method	126
6.18	Histogram (Answers to Question 1)	136
6.19	Histogram (Answers to Question 2)	136
6.20	Histogram (Answers to Question 3)	136
6.21	Histogram (Answers to Question 4)	137
6.22	Histogram (Answers to Question 5)	137
6.23	Hand Motion Visualization	138
6.24	Motions Snapshots	147

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

There is a wide range of applications where it is necessary to monitor the human body motion. Humans have limbs with the ability to rotate with one, two or three degrees of freedom. This dissertation deals with three-dimensional motion detection. In specific, how to detect the human hand motion in three dimensions. The human palm can rotate with three degrees of freedom and this research explores how to monitor its rotational movements. Furthermore, it explores whether it is possible to detect the thumb position with respect to the palm.

1.2 Objective of the Research

The objective of this research is the development and implementation of a sensor fusion algorithm to detect three dimensional human hand movement using Micro-Electro-Mechanical sensors (MEMS). The research aims to determine the possibility of tracking three-dimensional human hand motion using MEMS inertial (gyroscope and accelerometer) and magnetic sensors. This study seeks to understand what are the shortcomings and advantages of each sensor. This research explores how to utilize the advantages of each sensor to compensate the other sensor's shortcomings to achieve accurate rotational measurements.

The research is focused on tracking the orientation of the palm and the hand. As the articulations of the remaining 4 fingers (index, middle, ring and pinky) to the palm are more restricted than the thumb, a successful approach for thumb tracking should also be applicable for the remaining fingers.

1.3 Motivation

This work has been motivated by the need for information about the human body movement, position or gesture in a wide spectrum of applications. Three dimensional user interfaces (3DUI), human computer interaction (HCI), augmented reality, health care, cyber security, sport accessories, sign language, robotic and military applications are examples of these potential fields. Each of this applications include different examples for using human body motion detection. For instance, in health the care domain, there are limb detection requirements for stroke rehabilitation, remote health care monitoring, emergency occurrence detection, etc.

1.4 Significance of this Research

Hand movements have a critical role in many human activities and different technologies have been employed to track the hand movement. However, many of the previous approaches did not work well in real life situations. This research aims to find a solution to trace the movements in real life situations so that it can be used regardless of location or environmental situations. MEMS inertial sensors with miniature size and low energy consumption can be considered a very promising approach. However, it is critical to compute the rotational motion with acceptable accuracy, while MEMS sensors are prone to different types of errors which make it challenging to use them for motion detection. This study focuses on finding a solution to make MEMS sensors applicable for this application.

1.5 Literature Review

There is increasing interest in research on human body movement detection. Different technologies have been employed to track the movement and orientation of the human body [9] such as spatial scan (uses optical devices to scan the working volume), outside-in systems (uses video cameras at the reference to record the movement of the object), video metric (uses several cameras attached on the object) and beam scanning. Vision-based techniques suffer from some limitations such as portability, light and shadow interferences and requirement of line-of-sight, that make these techniques unsatisfactory for real time applications.

Recently Micro-electromechanical systems (MEMS) inertial/magnetic sensors have emerged. These sensors exhibit favorable properties such as [1] light weight, compact size and low energy consumption, and have the potential to resolve many limitations present in the vision-based techniques. These sensors can simply be attached on the human limbs to record the data associated with their motion. There are various studies that employed wearable sensors for different purposes.

Previous works related to the topic of this dissertation will be presented in following two section. In the first section, some of the studies that developed wearable devices for body motion tracking using inertial sensors will be introduced. The second section describes the efforts that concentrated on human hand motion detection.

1.5.1 Wearable Sensors Based on Inertial/Magnetic Measurements

As it was mentioned, several researchers have explored the use of MEMS inertial/magnetic sensors to monitor the movement of different human limbs or gestures produced by those motions. Some of these attempts focus on a specific limb such as the head, an arm, a leg or even the tongue. There are other studies that target the recognition or classification of whole body gestures. In the following, some of these attempts are categorized based on their area of application.

Sports Accessories

A number of studies have explored the use of inertial sensors for sport accessories. Traditionally, the performance of elite athletes is analyzed in the laboratories equipped with instrument to control the environmental conditions. Although those laboratory instruments determine the performance of the undertaken activity, the athlete cannot use most of them in a real training field [43]. Emerging MEMS sensors provide the possibility to build sport accessories which are portable. Sensor-based accessories can enhance the quality of training by providing feedback about the athlete's motion. Accelerometers and gyroscopes are used to detect the right movement characteristics and physical parameters of the athlete's movements. These sensors are also used to calculate the physical activity undertaken by athletes. The following paragraphs outline some of the efforts directed to the creation of sensor-based sport accessories.

King [47] described an IMU-based system that measures the dynamics of a golf club. This system is used for golf swing training to create a consistent and controlled swing. Earlier the swing training was performed using vision based techniques with

expensive equipment and sensitive set up that required stringent light control. In other study [30] inertial sensors were placed on both the golf club and the player's hand to provide more accurate information about the golf swing.

Alvarez et al. [4] attached an accelerometer and a gyroscope on a shoe to estimate the stride length. Harding [36] described using inertial sensors to classify aerial acrobatic maneuvers in half-pipe snowboarding.

A ski jumping motion detection system was developed [75] using an accelerometer, a magnetic sensor and a video camera. The purpose of this system is to measure the inrun descending speed, slant of the body of the jumper while flying and jumper's posture at the jumping point.

Auvient [6] developed an accelerometer-based system to gather information about characteristics of an athlete's stride. The system output would be information about the athlete's running style that can be used by trainers to coach the athlete.

Home Rehabilitation

The rehabilitation process is an expensive and often bothersome experience for the patients. Furthermore due to limited number of personnels and limited therapy resources in the rehabilitation centers, patients do not receive enough treatment or they need to wait long times to receive treatment. A home-based rehabilitation system would facilitate the rehabilitation process and decrease the cost of the process. Providing the home rehabilitation for stroke patients has been one of the major goals of home rehabilitation programs. More than 75 percentage of stork patients need rehabilitation treatments after being discharged from the hospital [86] . The stroke rehabilitation process aims to help patients to recover motor function abilities.

There are some efforts for implementing home rehabilitation using inertial sensors in combination with other sensors and technologies. Tao et al [81] implemented a home rehabilitation system for stroke patients combining visual and inertial sensors to track the human upper limbs. Zhou [98] places a tri-axis accelerometer, gyroscope and magnetometer close to the elbow and wrist joints to measure the upper limb motion. In [97] an arm movement tracking system was described that can detect the arm motion in three dimensions.

Health Monitoring

Some studies have focused on developing systems for supervision and assistance to ensure human wellbeing and safety. Such systems provide solutions for various needs such as fall detection, tremor or seizure detection, measuring human vital signs and assisted living services for people with disabilities and elderly individuals. Wearable inertial sensors can be utilized by users who are prone to life threatening occurrences, in order to detect the emergency situation and active an alarm message to their caregivers.

Zwartjes et al. [100] developed an ambulatory monitoring system that analyzes the occurrence and severity of tremor, bradykinesia and hypokinesia.

Salarian et al. [72] attached gyroscopes on the patients forearms to quantify the tremor and bradykinesia in patients with Parkinson's disease. Hoff [39] used four pairs of accelerometer sensors on the leg, arm and trunk of patients with Parkinsons disease to identify accelerometric characteristics of Levodopa-induced dyskinesias.

Powell et al. [64] used wireless inertial sensors for tremor assessment .

Jallon [42] has developed an epilepsy seizure detection system based on tri-axis accelerometer sensors. Hidden Markov models were used for signal processing algorithm.

In [11] inertial and magnetic sensors were placed on patients with epilepsy to quantify their movements during motor seizures. A classification system was designed for analysis of epileptic motor manifestations. Ryan and Venkatesan [15] attached an accelerometer on the human wrist to detect seizure movements based on a threshold-based algorithm.

Systems for monitoring the position of a patient's head were introduced in [22] and [19], using accelerometers. These systems intend to help the patient keep the head at the proper position after special ophthalmological surgery. Keeping the head at the proper position is vital for recovery. There are several studies exploring fall detection using wearable devices. Different approaches have been tested such as measuring accelerations of parts of the body, combination of body acceleration and physiological signs such as heart rate and blood pressure and infrared detectors [57]. Lai et al. [52] used several tri-axis accelerometers for joint sensing of injured body parts at the time of a fall accident. Sensors are placed at the different parts of body and the system aims to differentiate between behavioral events and falling. Wu and Xue [90] investigated making a fall detector that is able to sense the impending fall using inertial sensors. To distinguish between fall incidents and intentional activities a threshold detection algorithm was used. The threshold-based algorithm also was used in [13] and [45] to detect the fall occurrence.

Assisted Living

It is expected that in a near future robots will be used to assist elderly individuals, patients and people with disabilities. Some studies explored using wearable sensors to command a robot or other devices for assisted living purposes. Here a few of such studies are mentioned.

Huo developed an invasive system using a small tongue-mounted permanent magnet and an array of magnetic field sensors for people with severe disabilities to detect the intentional tongue movements. This system classifies the intentional tongue movements and translates them to user-defined computer commands. This system was designed to assist people with extreme disabilities such as tetraplegia [41]. Zhu et al. [99] presented an assisted living system helping disabled people and the elderly for human intention recognition to command a robot. The data from the inertial sensor placed in the finger of subject is used to recognize the gesture through a Hierarchical Hidden Markov Model.

Raya et al. [67] developed a head worn device for people with cognitive and physical impairments to be used as mouse. A tri-axis accelerometer, a tri-axis gyroscope and a tri-axis magnetometer are attached to the helmet. The system helps people with disability, who cannot use standard pointing devices, to interact with the computer using this head worn mouse.

Activity Monitoring

Activity monitoring aims to recognize the body posture, for example to sitting walking, standing and lying, or to quantify the movement intensity. Moore et al [51] used the cell phone accelerometer to identify the physical activities that user performs such as walking, jogging, climbing, sitting and standing. Giansanti et al. [31] used an inertial measurement unit to explore the trunk kinematics during the sit-to-stand transition. To recognize the beginning and end of the movement a threshold based method was used.

Bussmann et al. [16] placed accelerometers on the arms, trunk and thighs to determine the body posture in post measurement analysis. Bachman [7] used three-axis accelerometers, gyroscopes and magnetometers with a quaternion-based comple-

mentary filter to detect the posture of an articulated body in real time. Fleury [28] used a three-axis magnetometer to detect the changes of direction of a walking individual in an ideal situation, without any magnetic disturbance. Barralon et al. [10] attached three accelerometers to the chest to classify the walking and postural transitions (sit to stand and back to sit).

Human Computer Interaction for Gaming and Augmented Reality

Utilizing body worn sensors to capture the player's motion in the video gaming is becoming popular to enhance the game attraction. Some researchers have developed video games integrated with body worn sensors and optical tracking. The player's motion is used as input to navigate the game.

Hao [88] developed a multi-screen cyber-physical video game using an accelerometer and a compass to capture the player's movements. In [8] feasibility of automatic recognition of hand gestures using wearable motion sensors for a 3D educational parking game was explored. Different gesture classes were defined including circular motions, periodic motion, not periodic motions and pointing with the hand. It is stated that this system showed sufficient accuracy for real time gesture recognition. Geitmayr and Drummong [68] developed a hand held edge tracking system for outdoor augmented reality using combination of optical tracking, gyroscope, accelerometer and magnetic sensors. The extended Kalman filtering approach was used for the sensor fusion. Shark et al. [54] described a virtual reality game using inertial and ultrasonic tracking systems. Two players can play table tennis and the game tracks the position and orientation of the player's head and racket. [91] presents a hybrid tracking system including gyroscope and camera for augmented reality tracking that enables six degree of freedom pose tracking. There are also vision-magnetic hybrid systems to create augmented reality environments. [38] [5].

1.5.2 Hand Motion Detection Technologies

This section describes the studies that addressed human hand motion detection specifically. Different types of sensors and technologies were employed in order to monitor the human hand motion and gestures. These endeavors can be categorized on the basis of the approach used as follows:

- Vision-based methods
- Acoustic tracking sensors
- Strain gauge sensor gloves
- Flex sensor gloves
- Infrared transmitter-receiver sensor gloves
- Gloves based on magnetic induction coils
- Electromyogram (EMG) signals
- Set of permanent magnets and magnetometer
- Inertial sensors

Some of these studies are outlined in follow paragraphs.

In [82] flex sensors were attached on a cotton glove for finger position sensing.

Bui and Nguyan [14] developed a glove with six accelerometers, one on the each finger and one the back of the hand, to recognize posture in two dimensions. The system was designed to recognize posture in Vietnamese Sign Language. A fuzzy classification algorithm was implemented to classify the gestures.

Saggio [71] developed a data glove with bending and force sensing resistors to associate sounds to specific movements of fingers joints. The system was composed of three units including sensor unit, mapping unit and synthesizer unit. The system

maps n sensor outputs to the m synthesis parameters using a Feedforward Back-propagation Neural Network.

Tognetti [85] presents a comfortable and light glove based on a fabric built with Electrically Conductive Elastomer (CE) materials. When this type of material deforms by hand movement, it shows piezoresistive properties. The study aimed to map the sensor values into hand kinematic configuration.

Fahn [27] developed a data glove with magnetic generator coils and sensor modules to track the fingertip position.

In [21] a permanent magnet is attached to the back of the thumb and a pair of magnetometers were placed on the back of the fingers. Moving the thumb would affect the magnetometer output which is used to calculate the thumb position. This system can be used for 3D computer interaction.

Saponas [74] developed a muscle computer interaction system. The system used the forearm electromyograph (EMG) signals to classify the finger gesture using a support vector machine (SVM).

Kim [46] presented a data glove equipped with three tri-axis accelerometers placed on the thumb, the middle finger and the back of the hand. This glove was used in combination with a 3D digital hand model for hand movement tracking. A gesture recognition algorithm was implemented to classify three hand gestures including scissor, rock and paper. The results showed that the classifier can recognize the gestures with 100 percent accuracy.

In [37] five dual-axis MEMS accelerometers were used to recognize the 26 hand shapes of the American Sign Language Alphabet. The system classified twenty one out of twenty six letters with 100 percent accuracy. Xu et al. [96] described a human hand recognition system combining tri-axis accelerometer and electromyogram signals. The system is supposed to be used as a human computer interaction system.

Many vision-based studies have explored hand motion detection and gesture recognition [60] [70] [50] [61] [3] [62] [26].

1.6 Structure of the Dissertation

The rest of this dissertation is organized according to following structure.

Chapter two is about three dimensional motion. Mathematical expression of rotation in three dimension is explained in this chapter. Furthermore, concepts of quaternion, gimbal lock and Euler angels will be explained.

Chapter three describes MEMS inertial and magnetic sensors. Each of accelerometer, gyroscope and magnetometer sensors will be introduced. In addition, various types of potential errors for these sensors will be discussed.

Chapter four describes the sensor fusion approach which were used in this study to combine the sensors data and compute the rotational results. Quaternion-based Kalman filter approach will be explained in details. Two optimization methods, which are utilized to calculate the quaternion orientation from magnetometer and accelerometer sensors, will be described.

Chapter five explains the details about implementing the sensor fusion algorithm. It details the hardware components as well as codes.

Chapter six reports the experiments design and and results of experiments. Finally, the conclusion for this research is presented in chapter seven. It is described how tis study may potentially be furthered in the future.

CHAPTER 2

THREE DIMENSIONAL ROTATION

2.1 Introduction

To monitor the human hand motion it is necessary to detect the three-dimensional hand rotation. This chapter provides the background information about the concepts of three dimensional rotation and quaternions for this dissertation.

2.2 Three Dimensional Rotation

The three dimensional space (R^3) can be expressed using a Cartesian system defined by three mutually perpendicular axes passing through the origin point. These axes are called X-, Y- and Z-axis. These axes are placed based on the right hand rule such that pointing with right index finger positive X-axis, and with the right middle finger (folded) to the positive Y-axis will causes the thumb to point toward the positive Z-axis. Each point in such space is presented by three elements as

$$A = (x_1, y_1, z_1).$$

Each point can be presented as a vector from the origin to that point. Figure 2.1 depicts a point in the three dimensional space.

An object can rotate around each of these orthogonal axes. The rotation about X-axis is called Roll, rotation about Y-axis is called Pitch and Yaw refers to rotation about Z-axis. These rotations are shown in figures 2.2, 2.3 and 2.4.

Figure 2.1: Three Dimensional Reference Frame

Figure 2.2: Roll

Figure 2.3: Pitch

Figure 2.4: Yaw

2.3 Coordinate Systems

To describe the three dimensional rotation two reference frames are used. These reference frames are the inertial and the body-fixed frames. Each of these frames is an orthogonal, right handed Cartesian frame. In the following these frames are described.

2.3.1 Inertial Frame

The origin of this frame is located at the center of the Earth. The X-axis passes through the equator and the Z-axis passes through the Earth's axis. The Y-axis is orthogonal to the other two axes.

2.3.2 Body-fixed Frame

The origin of the body-fixed frame is located at the center of gravity of the rotating object and the Z-axis is perpendicular to the object. The x- and Y- axes are orthogonal in a horizontal plane. When the object rotates, this frame rotates with respect to the inertial frame. If the inertial sensors are attached to the rotation object using this reference frame is required. Figure 2.5 depicts the inertial and body-fixed frames for a rotation object.

Figure 2.5: Inertial and Body-fixed Reference Frames

2.4 Euler Angles

Euler angles are used to represent the rotation in three-dimensional Euclidean space. The Euler theorem states that [49] “Any two independent orthogonal coordinate frames can be related by a sequence of rotations (not more than three) about

the coordinate axes, where no successive rotations may be about the same axes". Based on Euler rotational theorem, any rotation can be express with three rotational angles. These rotational angles are called Euler angles and defined as:

ϕ represents rotation about x -axis

θ represents rotation about y -axis

ψ represents rotation about z -axis

2.4.1 Rotation Matrices

Euler angles present the orientation of a an object,with respect to a known coordinate system, using combination of rotations about different axes. The rotation around each axis can be expressed mathematically with rotation matrices. The rotation matrices are defied in equations 2.1, 2.2 and 2.3 Rotation matrix about x -axis:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad (2.1)$$

Rotation matrix about y -axis:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.2)$$

Rotation matrix about z -axis:

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Multiplication of a rotation matrix with a vector rotates the vector. Assuming V is a vector in $\mathbf{R}^3$, if V rotates about a single axis a new vector would be created. This new vector can be calculated as equation 2.4:

$$V' = R_n(\theta)V \quad (2.4)$$

where θ is the angle of rotation and n is the axis of rotation. Any rotation can be performed using a sequence of three rotations about the coordinate axes[63]. There are twelve different rotation sequences with three Euclidean axes. These twelve sequences are:

$$\begin{aligned} &XYZ \ YZX \ ZXY \\ &XZY \ YXZ \ ZYX \\ &XZX \ YXY \ ZYZ \\ &XYX \ YZY \ ZXZ \end{aligned}$$

Any rotation with three degrees of freedom can be expressed using one of the above sequences. To understand how each sequence rotates the objects, we explain one of the sequences here. The sequence XYZ means that initially rotation happens around the X-axis, then there is a rotation about the new Y-axis and is followed by a rotation about the new Z-axis. Notice that with three axes fifteen more sequences were possible with consecutive rotation around the same axis. Since consecutive rotation around the same axis reduces the degrees of freedom, these possible sequences are ignored. To apply a sequence of rotation like XYZ, the related rotation matrices should be multiplied as

$$R_{XYZ} = R_X(\phi)R_Y(\theta)R_Z(\psi) =$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta) \cos(\psi) & \cos(\theta) \sin(\psi) & -\sin(\theta) \\ \sin(\phi) \sin(\theta) \cos(\psi) - \cos(\phi) \sin(\psi) & \sin(\phi) \sin(\theta) \sin(\psi) + \cos(\phi) \cos(\psi) & \sin(\phi) \cos(\theta) \\ \cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi) & \cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi) & \cos(\phi) \cos(\theta) \end{bmatrix} \quad (2.5)$$

In a general form, the rotation matrix for a sequence of rotations is expressed as equation 2.6:

$$R_{abc} = R_a(\theta)R_b(\theta)R_c(\theta) \quad (2.6)$$

Where each of a , b and c can be X , Y or Z . Matrix R_{abc} is also called the Direct Cosine Matrix(DCM) which is expressed in terms of Euler angles.

2.4.2 Gimbal Lock

A ring that can rotate freely around an axis is called gimbal. To represent the three-dimensional rotation, three gimbals are used in a way that each of the gimbals can rotate about one of the X -, Y - or Z -axis. Each gimbal can rotate freely about its respective axis. These gimbals are placed together in a way that when the outer gimbal rotates all the other gimbals rotate with it. Rotating the intermediate gimbals does not affect the outer gimbal but the innermost gimbal follows its

rotation. Finally the innermost gimbal rotates without affecting the other gimbals. The combination of these three gimbals is shown in figure 2.6. The gimbal lock

Figure 2.6: Three Gimbals

phenomenon happens as a result of a series of rotations that align the two axes of the three gimbals into the same direction. In other words, it is possible that because of a sequence of rotation, two of the axes are driven into a parallel direction, then these two axes rotate in the same plane, which causes loss of one degree of freedom. This phenomenon is shown in figure (2.7). To avoid the gimbal lock problem the three dimensional rotation can be expressed using quaternions. The next section, describes how to use quaternions to represent three dimensional rotation.

2.5 Quaternions

Quaternions can be used to express the rotation. Advantage of using quaternions over Euler angles, for rotation is that the singularity problem can be avoided. Additionally, operation with quaternions do not need trigonometric functions, which

Figure 2.7: Gimbal Lock Effect

simplifies the process of filter implementation for real-time applications.

In this section the concept of quaternions and the quaternion mathematics are describe and eventually it is explained how a quaternion is used to represent the three-dimensional rotation.

2.5.1 What Are Quaternions

In 1843, quaternions were invented by Hamilton. Quaternions are hyper complex numbers of rank four. In quaternions, three imaginary parts are defined, presented by i, j and k elements. Equation 2.7 to 2.10 show relations for these basic elements:

$$i * i = i^2 = -1 \quad (2.7)$$

$$j * j = j^2 = -1 \quad (2.8)$$

$$k * k = k^2 = -1 \quad (2.9)$$

$$i * j * k = -1 \quad (2.10)$$

Quaternions can be expressed in different notations. A quaternion can be expressed as linear combination of four elements as equation 2.11:

$$q = (q_0, q_1, q_2, q_3) = q_0 + iq_1 + jq_2 + kq_3 \quad (2.11)$$

Where q_0 is the real part which is called scalar, and the rest are the imaginary parts of the quaternion, which Hamilton called them vector. q_0, q_1, q_2 and q_3 are real numbers. A quaternion may also be represented as a scalar and a vector as equation 2.12:

$$q = (q_0, \mathbf{q}) \quad (2.12)$$

The vector part, $\mathbf{q}$, is a vector in $\mathbf{R}_3$:

$$\mathbf{q} = iq_1 + jq_2 + kq_3 \quad (2.13)$$

The i, j and k elements are used to represent the standard orthogonal three dimensional basis as the following vectors:

$$i = (1, 0, 0) \quad (2.14)$$

$$j = (0, 1, 0) \quad (2.15)$$

$$k = (0, 0, 1) \quad (2.16)$$

Quaternions Equality

Two quaternions are equal if and only if their corresponding components are equal. Considering two p and q quaternions as:

$$q = q_0 + iq_1 + jq_2 + kq_3$$

$$p = p_0 + ip_1 + jp_2 + kp_3$$

Then q is equal to p if and only if:

$$q_0 = p_0$$

$$q_1 = p_1$$

$$q_2 = p_2$$

$$q_3 = p_3$$

Quaternions Addition

The sum of two quaternions follows the same rule for addition of complex numbers.

This rule says that the corresponding components should be added as equation 2.17:

$$q + p = (q_0 + p_0) + i(q_1 + p_1) + j(q_2 + p_2) + k(q_3 + p_3) \quad (2.17)$$

Quaternions Multiplication

The multiplication rule to obtain the product a scalar by a quaternion is simple.

Each component of quaternion should be multiplied by the scalar. If p is a quaternion and a is a scalar, then the product of p and the a is given by equation 2.18:

$$ap = ap_0 + iap_1 + jap_2 + kap_3 \quad (2.18)$$

However, the multiplication of two quaternions is not that straightforward. It's rule is similar to polynomial multiplication, considering the multiplicative properties of i , j and k as equations 2.19 to 2.22:

$$i^2 = j^2 = k^2 = ijk = -1 \quad (2.19)$$

$$ij = k = -ji \quad (2.20)$$

$$jk = i = -kj \quad (2.21)$$

$$ki = j = -ik \quad (2.22)$$

The above products are not commutative, therefore the quaternions multiplication can not be commutative either. Now, using the above fundamental rules, the product of two quaternions is defined as equation 2.5.1. The symbol $\otimes$ is used for quaternions multiplication.

$$\begin{aligned} p \otimes q &= (p_0 + ip_1 + jp_2 + kp_3) \otimes (q_0 + iq_1 + jq_2 + kq_3) \\ &= p_0q_0 + ip_0q_1 + jp_0q_2 + kp_0q_3 \\ &\quad + ip_1q_0 + i^2p_1q_1 + ijp_1q_2 + ikp_1q_3 \\ &\quad + jp_2q_0 + jip_2q_1 + j^2p_2q_2 + jkp_2q_3 \\ &\quad + kp_3q_0 + kip_3q_1 + kjp_3q_2 + k^2p_3q_3 \end{aligned}$$

Considering the fundamental rules, the product of two quaternions would be expressed as:

$$\begin{aligned} p \otimes q &= (p_0 + ip_1 + jp_2 + kp_3) \otimes (q_0 + iq_1 + jq_2 + kq_3) = \\ &= p_0q_0 + ip_0q_1 + jp_0q_2 + kp_0q_3 + ip_1q_0 - p_1q_1 + kp_1q_2 - jp_1q_3 \\ &\quad + jp_2q_0 - kp_2q_1 - p_2q_2 + ip_2q_3 + kp_3q_0 + jp_3q_1 - ip_3q_2 - p_3q_3 \end{aligned} \quad (2.23)$$

The 2.23 equation can be rewritten as equation 2.24:

$$\begin{aligned} p \otimes q &= p_0q_0 - (p_1q_1 + p_2q_2 + p_3q_3) + p_0(iq_1 + jq_2 + kq_3) + q_0(ip_1 + jp_2 + kp_3) \\ &\quad + i(p_2q_3 - p_3q_2) + j(p_3q_1 - p_1q_3) + k(p_1q_2 - p_2q_1) \end{aligned} \quad (2.24)$$

Quaternions Conjugate

The conjugate operation for the given quaternion, q , is defined as equation 2.25:

$$\begin{aligned} p &= p_0 + ip_1 + jp_2 + kp_3 \\ p^* &= p_0 - ip_1 - jp_2 - kp_3 \end{aligned} \quad (2.25)$$

p^* denotes the conjugate of p .

Quaternions Norm

The norm of a quaternion p , denoted as $|p|$ or $\mathbf{N}(p)$, is a scalar and is defined as equation 2.26

$$|p| = \sqrt{p^*p} = p_0^2 + p_1^2 + p_2^2 + p_3^2 \quad (2.26)$$

When the norm of a quaternion is equal to 1, the quaternion is called unit or normalized quaternion. The product of unit quaternions and inverse of unit quaternion are also unit quaternions. A unit quaternion can be written as equation 2.27

$$q = [q_0, \mathbf{q}] \quad (2.27)$$

Where,

$$q_0 = \cos\left(\frac{\theta}{2}\right)$$

and

$$\mathbf{q} = \hat{u} \sin\left(\frac{\theta}{2}\right)$$

u is a three dimensional vector with length 1 and θ is the rotational angle. Therefore the unit quaternion can be expressed as equation 2.28:

$$q = \cos\left(\frac{\theta}{2}\right) + i \sin\left(\frac{\theta}{2}\right) + j \sin\left(\frac{\theta}{2}\right) + k \sin\left(\frac{\theta}{2}\right) \quad (2.28)$$

And the conjugate of quaternion can be written as equation 2.29:

$$q = \cos\left(\frac{\theta}{2}\right) - i \sin\left(\frac{\theta}{2}\right) - j \sin\left(\frac{\theta}{2}\right) - k \sin\left(\frac{\theta}{2}\right) \quad (2.29)$$

Quaternion Inverse

The inverse of a quaternion p is denoted as p^{-1} and, based on the definition of inverse, we should have

$$p^{-1}p = pp^{-1} = 1$$

Using the norm and conjugate tools, the above statement can be expressed as equation 2.5.1

$$p^{-1}pp^* = p^*pp^{-1} = p^*$$

In the previous section we noticed $pp^* = \|N(p)\|^2$, so inverse formula can be written as equation 2.30:

$$p^{-1} = \frac{p^*}{\|N(p)\|^2} = \frac{p^*}{|p|^2} \quad (2.30)$$

If the quaternion is a unit quaternion, then the norm is equal to 1, therefore the inverse quaternion would be equal to quaternion conjugate, equation 2.31.

$$p^{-1} = p^* \quad (2.31)$$

2.5.2 Quaternion Rotation

We previously explained how a three dimensional rotation can be expressed using rotation matrices. It was shown there that a vector can be rotated by multiplying it by a rotation matrix. Rotations also can be accomplished using quaternions. A three dimensional vector v can be rotated around axis $\hat{u}$ by an angle of θ , using unit quaternion $q = \cos(\frac{\theta}{2}) + \hat{u} \sin(\frac{\theta}{2})$. The rotated vector v' is described as equation 2.32:

$$v' = q \otimes v \otimes q^* \quad (2.32)$$

Where, q^* is the conjugate of quaternion q and v' is the rotated version of vector v about axis of q . Axis q can be each of X-, Y- or Z-axes. The quaternion for rotation about each of these axis is expressed as equations 2.33, 2.34 and 2.35.

$$q_x = \cos(\frac{\phi}{2}) + i \sin(\frac{\phi}{2}) + j \sin(\frac{0}{2}) + k \sin(\frac{0}{2}) = \cos(\frac{\phi}{2}) + i \sin(\frac{\phi}{2}) \quad (2.33)$$

$$q_y = \cos(\frac{\theta}{2}) + i \sin(\frac{0}{2}) + j \sin(\frac{\theta}{2}) + k \sin(\frac{0}{2}) = \cos(\frac{\theta}{2}) + j \sin(\frac{\theta}{2}) \quad (2.34)$$

$$q_z = \cos(\frac{\psi}{2}) + i \sin(\frac{0}{2}) + j \sin(\frac{0}{2}) + k \sin(\frac{\psi}{2}) = \cos(\frac{\psi}{2}) + k \sin(\frac{\psi}{2}) \quad (2.35)$$

Equation 2.32 can be defined in a general form as equation 2.36:

$$v' = R_Q v \quad (2.36)$$

Where R_Q is quaternion rotation matrix. The general form of this rotation matrix is expressed as equation 2.37

$$R_Q = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_0q_3 + 2q_1q_2 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & -2q_0q_1 + 2q_2q_3 \\ -2q_0q_2 + 2q_1q_3 & 2q_0q_1 + 2q_2q_3 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.37)$$

As it was explained earlier, there are twelve forms of rotation sequences with three degrees of freedom. To express the quaternion rotation of each sequence, we should multiply the corresponding quaternion representations of each of the axes involved.

CHAPTER 3

MEMS INERTIAL AND MAGNETIC SENSORS

3.1 Introduction

Inertial sensors consist of accelerometer and gyroscope sensors. A unit with gyroscopes and accelerometers is also called an "Inertial measurement Unit" (IMU). These sensors can be utilized for motion detection applications. Inertial sensors are also used in combination with magnetometer sensors to achieve better measurements.

Advances in MEMS (Micro Electro Mechanical Systems) technology have revolutionized the inertial sensors industry. MEMS sensors have significant advantages over non-MEMS sensors in terms of size, energy consumption, reliability and cost. The Emergence of miniaturized MEMS inertial sensors has provided the possibility of adopting such sensors in the expanded range of consumer electronic products.

In this chapter the basic principles of accelerometer, magnetometer and gyroscope sensors are described and their error sources are introduced.

3.2 Inertial sensors

Inertial sensors function based on inertial forces. These sensors capture the external motion forces acting on them. These forces consist of acceleration and angular rotation. The measured forces are used to calculate the acceleration and angular rotation of the body. Knowing the acceleration forces acting upon the body, it should be possible (in theory) to determine the translational motion of the body. Accelerometers sensors respond to the forces associated with acceleration and gyroscopes sensors are sensitive to the angular rotations.

Inertial sensors have various applications in different areas. These sensors are applied for navigations, video gaming interface, automotive, motion detection for health monitoring, monitoring of machinery vibration, mobile phones, vibration detection of buildings, sport training, etc. [56] [12]

The combination of accelerometers and gyroscopes is called "inertial measurement unit" (IMU). Sometimes the IMU has magnetometers in addition to accelerometers and gyroscopes. The IMU is commonly used to measure the inertial forces in three dimensions. In the following, accelerometer, gyroscope sensors are described.

3.2.1 Accelerometers

Accelerometers detect the acceleration due to external forces acting on objects. The output of an accelerometer is combination of the acceleration due to the Earth's gravity and the linear acceleration due to motion. Each sensor responds to the acceleration along a specific axis. In the past, accelerometers could sense the acceleration in just one axis, but recently tri-axis accelerometers are available in the market. To measure the acceleration in three dimensions, with one-axis accelerometer, three accelerometers were used. These sensors were positioned so that their sensitive axes were oriented East-West, North-South and vertically. The tri-axis accelerometer has three orthogonal axes that can sense the acceleration forces in three dimensions.

The acceleration associated to external forces acting on an object can be explained by Newton's second law of motion: "A force F acting on a body of mass m causes the body to accelerate with respect to the inertial space" [84]. The equation 3.1 is the mathematical representation of Newton's second law of motion.

$$F = ma \tag{3.1}$$

Where a is the acceleration and m represents the mass.

The force (F) can be caused by both motion of the body and the Earth's gravity (g).

A basic form of accelerometer consists of a small mass that is connected to a case through a spring. The mass is also called proof or seismic mass. Figure 3.1 shows the basic form of the accelerometer. When an external force is applied to

Figure 3.1: Basic Accelerometer

the case, because of the mass inertia, it shows resistance against the movement. Consequently, it creates tension or extension in the springs, and the mass position with respect to the case will change. The mass displacement makes the output of the sensor vary. This variation is used to measure the acceleration. [84]

Figure 3.2 depicts the accelerometer when a force is applied to it along its sensitive axis.

There are also MEMS accelerometers that are built based on other techniques such as capacitive-based displacement accelerometers, piezoelectric and piezoresistive-based accelerometers and resonant element accelerometers. [59]

Figure 3.2: Accelerometer Response to the External Forces

3.2.2 Gyroscopes

Gyroscope sensors are devices to measure the angular motion of an object. The sensor output is proportional to its rotation. Some gyroscopes measure the angular velocity (rate gyroscopes) while other gyroscopes measure the orientation (angle gyroscopes).

Gyroscopes have been made based on different technologies. These sensors can be categorized into three major groups including mechanical gyroscopes, optical gyroscopes, and MEMS gyroscopes.

Mechanical gyroscopes typically consist of a spinning wheel mounted in two gimbals. The wheel can rotate in three axis. It works based on the effect of conservation of angular momentum.

There are two types of optical gyroscopes including fiber optic gyroscopes(FOG) and ring laser gyroscopes (RLG) [87]. FOGs use a coil of optical fiber and two light beams to measure rotation. Two light beams, in opposite directions, are entered into the coil and travel at constant speed. When the sensor is rotating, the light

traveling in the direction of rotation traverses a longer path compared to the light traveling in the opposite direction. This is so called "Sagnac effect". By measuring the path length, the angular rate of rotation can be computed. Ring laser gyroscopes consist of a closed loop tube with three arms. At each corner of tube is mirror is placed and the tube is filled with helium-neon gas. These sensors also work based on the "Sagnac effect".

MEMS gyroscopes work based on the Coriolis effect. If a mass m is moving with velocity v in a reference frame rotating at angular velocity w , then a force F will work upon the mass as equation 3.2:

$$F = -2m(w \times v) \quad (3.2)$$

These sensors measure the Coriolis acceleration acting on a vibrating element which is used as proof mass[76]. The vibrating elements are available in the form of a tuning fork or a vibrating wheel. All MEMS gyroscopes are rate gyroscopes.

MEMS gyroscopes have many advantages compared to mechanical and optical gyroscopes. MEMS gyroscopes have low power consumption, small size, low weight, high reliability, low maintenance cost, fast start up and lower price.

3.3 MEMS Inertial Sensor Errors

Inertial sensors suffer from several sources of errors. The error origin can be an environmental disturbance (such as temperature, magnetic field or air pressure), measurement equipment or random noises [23].

The inertial sensor errors can be classified into two major categories, including systematic (deterministic) errors and random (stochastic) errors. Both types of errors affect the accuracy of inertial sensor measurements. Therefore, to enhance

the performance of the inertial sensors it is necessary to detect and compensate for such errors [95].

Systematic errors can be estimated by reference measurement. The calibration process is used to estimate and compensate systematic errors. Scale factor, constant bias, misalignment, nonlinearity, and sign asymmetry are categorized as systematic errors. Unfortunately, even after the calibration process the data extracted from the sensors are contaminated with other types of errors.

Random noises consist of high frequency and low frequency components. To remove the high frequency noises different denoising techniques have been used, such as Wavelet, low pass filters and neural networks. The low frequency noises are modeled using random processes. Different random processes are utilized to model the inertial sensor's low frequency noises such as random walk, constant random and Gauss-Markov random processes[24].

To extract accurate and meaningful data from the sensor output, it is necessary to compensate all types of errors; otherwise the results would not be reliable. In the following, we introduce some types of common inertial sensor errors.

3.3.1 Angle Random Walk

Angle Random Walk (ARW) is the white noise added on the sensor output [92]. This noise usually results from the power supplies or semiconductor devices. The power spectral density of angle random walk is as equation 3.3:

$$s(f) = Q^2 \tag{3.3}$$

Where Q is the angle random walk coefficient.

3.3.2 Rate Random Walk

Rate random walk (RRW) is a random noise with unknown origin [25]. The power spectral density of this noise is expressed as equation 3.4:

$$s(f) = \frac{k^2}{2\pi} \frac{1}{f^2} \quad (3.4)$$

Where k is the rate random walk coefficient.

3.3.3 Flicker Noise

Flicker noise is a non-stationary noise whose power spectra is proportional to f^{-1} . The power spectrum shows that most of the power of this noise appears in low frequencies.

3.3.4 Quantization Noise

Quantization noise is the result of the digitalization process. A finite number of bits is used for storing the signal values, so information is lost and the digital signal is slightly different compared to original analog signal.

3.3.5 Sinusoidal Noise

The sensor's output exhibits an additive sinusoidal noise component. The reason is because these sensors work around a resonant frequency. Therefore, a pseudo-deterministic sinusoidal noise is seen in the output signal.

3.3.6 Bias Error

Bias error is a nonzero output signal which appears in the sensor output when input is zero. The bias offset causes the sensor output to offset from the true data by a constant value. This error is not dependent of external forces applied to the sensor. Bias error can be divided into three categories, including a static part (or bias offset), a random part (or drift) and a temperature dependent part[58]. The static bias and temperature bias can be compensated in the calibration process. Static bias can be measured by averaging the sensor output for a zero input signal. To compensate for this bias, it should be subtracted from the output data. The drift bias has a random nature and cannot be fixed in the calibration process. It should be treated as a stochastic error. Bias error is presented in Figure 3.3.

Figure 3.3: Bias Error

3.3.7 Scale Factor Error

The scale factor error is linear deviation of the input-output gradient from unity. Similar to the bias error, the scale factor error can be divided into a static part, a

drift part and a temperature dependent part. This effect is depicted in Figure 3.4. This error may arise from aging or manufacturing tolerance [34].

To compensate for the scale factor error the data should be multiplied with a constant factor [95].

Figure 3.4: Scale Factor Error

3.3.8 Scale Factor Sign Asymmetry Error

The scale factor sign asymmetry effect is the spontaneous change of at least one point in the measurement curve. This error is shown in Figure 3.5.

3.3.9 Misalignment (Cross-coupling) Error

This error is a systematic error and comes from the misalignment of the sensitive axes of the sensor with respect to the axes of the body frame [35]. This error appears because of manufacturing imperfection and can be compensated through calibration process. Figure 3.6 shows the sensor's X, Y and Z axis misalignment with respect to the body frame.

Figure 3.5: Scale Factor Sign Asymmetry Error

3.3.10 Non-linearity Error

The scale factor phenomenon does not always happen in a linear form, sometimes it appears in the form of second or higher order function [53]. This effect happens because of material properties and geometric shape of the sensor [95]. This effect is shown in Figure 3.7.

3.3.11 Dead Zone Error

The dead zone effect is an apparent discontinuity in the output data. The sensor does not sense the applied forces in the interval in which the discontinuity happens. This interval is called a "dead zone". This effect is usually produced due to mechanical stiction. Figure 3.8 depicts the dead zone effect.

3.3.12 Temperature Effect

Changes in the environment temperature affect the output of MEMS sensors. A temperature sensor is usually added to the inertial measurement units to be used

Figure 3.6: Axis Misalignment

for compensating the temperature effect.

3.4 Magnetometers

For centuries, navigators have been acquainted with the Earth's magnetic field effect. The earliest evidence about compass navigation is ascribed to the Chinese and dated 250 years B.C. [18]. Traditional magnetometers (compass) were simple instruments that pointed approximately to the magnetic north. A small magnetized aligns itself to be parallel with the horizontal component of the Earth magnetic field.

The Earth magnetic field is produced by the Earth and also interacting fields from the Sun, called the solar wind. The solar wind is the stream of high energy charged particles, which are released outward from the Sun continuously. To explain the source of Earth's magnetism, many hypotheses have been asserted. But only the "Dynamo Effect" mechanism is now considered plausible [18]. The Earth is composed of four layers, including Outer Crust, Mantle, Outer Core and Inner Core. The temperature at the Mantle layer, at the boundary with the Outer Core, is over 4000°C , which is hot enough to liquefy the outer core. The outer core consists of

Figure 3.7: Non-linearity Error

molten iron. In this theory, the Earth’s magnetism is attributed to the convection currents of molten iron in this layer.

The north pole of the Earth’s magnet is in the side of geographical south and vice versa. The geographical north and south axis, which is defined by the Earth’s rotational axis, does not coincide with to the axis of the Earth’s magnet [17]. The best approximation of discrepancy between magnetic pole and geographical pole is 11.5° . It is so called magnetic declination. Figure 3.9 presents the location of geographic and magnetic poles.

Three-axis MEMS magnetometers provide more information than traditional magnetometer. They measure the magnitude and direction of two horizontal and one vertical components of the magnetic field.

3.5 MEMS Magnetometer Errors

MEMS accelerometers are subject to bias error, scale factor error, misalignment errors and stochastic noise [44] [69]. Furthermore, the local magnetic fields can disturb the sensor output. Such disturbance is categorized into groups as [29]:

Figure 3.8: Dead Zone Error

Figure 3.9: Geographic and Magnetic Poles

- Soft iron disturbance: Soft iron disturbance is caused by the ferromagnetic objects in the vicinity of the magnetometer. These objects can distort the direction of the magnetic field.
- Hard iron disturbance: Hard iron effect is the result of the presence of any permanent magnetic field in the surrounding of the sensor. The permanent magnetic field causes a constant bias on the sensor output.

CHAPTER 4

SENSOR FUSION ALGORITHMS

4.1 Introduction

As it was mentioned in Chapter Three, both inertial and magnetic sensors are prone to different types of errors. Sensor fusion algorithms are used to compensate for different types of errors and to yield reliable results. The algorithms combine different types of sensor data and compensate for each sensor limitations using data from other sensors.

In this research human hand motion tracking was performed by implementing different types of sensor fusion algorithms. The purpose of this chapter is to explain the algorithms which were used for this hand motion detection project.

4.2 Sensor Fusion

The sensor unit which was used for this project consists of MEMS gyroscope, accelerometer and magnetometer sensors. Although the rotational motion can be computed by integrating gyroscope data, the results drift over time. The explanation for this phenomenon is that the integration accumulates the noise over time and turns noise into the drift, which yields unacceptable results. Roll and pitch rotations can be calculated using accelerometer data, but the results would be noisy. Furthermore, in order to measure rotation around the Z-axis (yaw), the other sensors need to be incorporated with the accelerometer. The Magnetometer sensor can be used in cooperation with inertial sensors to measure the yaw angle.

Neither the accelerometer nor the gyroscope provides accurate rotation measurements alone. This is the reason, for implementing a sensor fusion algorithm in order

to compensate for the weakness of each sensor by utilizing other sensors, as shown in Figure 4.1.

Figure 4.1: Combining the Sensors Output

To achieve the best possible hand rotation tracking from inertial and magnetic sensors, the Kalman filter approach was chosen. In this experiment the human hand motion tracking was pursued by implementing different algorithms. Since the Euler-based Kalman filter algorithm suffers from a singularity problem, the quaternion-based kalman filter was chosen to eliminate the singularity problem. Quaternion-based Kalman filter approach was implemented and two optimization techniques, Gauss-Newton and Gradient Descent, were used to track the human hand motion.

4.3 Least Squares Estimator

The “Least Squares Estimation” is a recursive algorithm to design linear adaptive filters. One of its variations known as RLS (Recursive least squares). The least squares approach is a method for estimating optimal data from noisy data[33]. The algorithm computes the filter coefficients that minimize the cost function which is the sum of weighted error squares. The RLS algorithm defines error as the discrepancy

between the desired signal and the actual signal. This method first was published by Andien-Marie Legendre, but it is mostly attributed to Carl Friedrich Gauss. Gauss describe this method in the following way: " the most probable value of the unknown quantities will be that in which the sum of the squares of the differences between the actually observed and the computed values multiplied by numbers that measure the degree of precision is a minimum" [77].

The Kalman filter is a least square error estimator. An estimator is a statistic that uses the observed data to calculate an estimate of a given quantity.

A system can be described in a matrix form as equation 4.1 [78],

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_l \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2n} \\ h_{31} & h_{32} & h_{33} & \cdots & h_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{l1} & h_{l2} & h_{l3} & \cdots & h_{ln} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_l \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_l \end{bmatrix} \quad (4.1)$$

The equation 3.4 can be written as equation 4.2.

$$Z_k = Hx_k + v_k \quad (4.2)$$

where Z is the observed dependent signal, x represent the original signal which carries the information and v is the measurement error.

Gauss assumed the signal x and the observed data to be linearly related. The overall objective is to estimate $\hat{x}$, in a way that minimizes the estimated measurement error $(H\hat{x} - Z)$. This difference between measured data and calculated data is called the residual. In the least-squares method, the best estimation is defined as the $\hat{x}$ that minimizes the the sum of the squares of the residuals, equation 4.3:

$$\varepsilon^2(\hat{x}) = (H\hat{x} - Z)^2 = \sum_{i=1}^m \left[\sum_{j=1}^n h_{ij} \hat{x}_j - z_i \right]^2 \quad (4.3)$$

To minimize the error we require to find the value of $\hat{x}$ such that:

$$\frac{\partial \varepsilon^2}{\partial \hat{x}_k} = 0 \quad (4.4)$$

or

$$2 \sum_{i=1}^m h_i k \left[\sum_{j=1}^n h_{ij} \hat{x}_j - z_i \right] = 2 \sum_{i=0}^m h_{ij} H \hat{x} - z_i = 0 \quad (4.5)$$

The equation 4.5 can be written as equation 4.6

$$2H^T[H\hat{x} - z] = 0 \quad (4.6)$$

Using equation 4.6, we can write the equation 4.7 which is called the normal form of the equation for the linear least squares problem [34].

$$H^T H \hat{x} = H^T z \quad (4.7)$$

The equation 4.7 has following solution,

$$\hat{x} = (H^H)^{-1} H^T z \quad (4.8)$$

4.4 Kalman Filter

The Kalman Filter was named after Rudolf Emil Kalman. He was born on May 1930 in Hungary and emigrated to the United States during world war II [34]. He completed his bachelor's and master's degrees in electrical engineering at Massachusetts Institute of Technology and received his PhD at Columbia university in 1957. His famous paper describing a recursive estimator was published in 1960. The estimator is called the Kalman filter and also is known as linear-quadratic estimation.

The Kalman filter has been applied in a wide range of applications and research areas, including economic modeling, process control, earthquake prediction, surveying and biomedical.[32] It has been used notably in the area of navigation, estimation

and tracking. The Kalman filter is a recursive algorithm which receives consecutive noisy data samples over time and estimates the variables in a way that minimizes the mean square error. The Kalman filter algorithm consists of two steps, including the Prediction (or Time Update) step and the Correction (or Measurement Update) step. In the prediction step, the Kalman filter predicts the state of the system and calculates the error covariance for the next step.

In the correction step, the filter incorporates received measurements to correct its prediction. In general, the prediction step is responsible for forecasting the the state of the system ahead in time and the correction step adjusts the prediction by applying the real measurement at the time.

The Kalman filter estimates the state of a system at a time (t) by using the state of the system at time ($t-1$). The Prediction-Correction cycle is depicted in Figure 4.2.

Figure 4.2: Kalman Filter Cycle

4.5 Discrete Kalman Filter

The discrete Kalman filter, which is applicable for linear systems, is described in this section. The discrete Kalman filter is much more frequently used than continuous Kalman filter. Even in applications in which the system's model is continuous,

because the measurement is mostly performed in a discrete manner, the discrete Kalman filter is often used.

In order to apply the Kalman filter, we need the "system model". The "system model" means to express the system in a mathematical form. Some times it is difficult to find the mathematical model for the system. To understand the "system model", you need to have knowledge about the state space model and the linear systems. The system model for the discrete time linear systems can be describe by the state transition as equation 4.9:

$$x_{k+1} = Ax_k + w_k, n \quad k \geq 0 \quad (4.9)$$

Where,

x_k is the state vector at time k, which is an $(n \times 1)$ column vector,

A is the state transition matrix, which is an $(n \times n)$ matrix,

w_k represents the zero-mean state noise vector, which is $(n \times 1)$ column vector,

The Kalman filter, minimizes the effect of the noise w on the signal x .The state variable, x_k , describes a physical quantity of the system like velocity, position, etc. The matrix A has the constant elements. The Kalman filter formulation assumes that the measurement is linearly related to the states as indicated in equation 4.10:

$$Z_k = Hx_k + v_k \quad (4.10)$$

where,

z_k is measurement of x at time k, which is an $(m \times 1)$ column vector,

H represents the observation matrix, which is an $(m \times n)$ matrix,

v_k is the zero-mean measurement noise, which is an $(m \times 1)$ column vector,

The matrix H includes constant elements and represents how each state variable is related to the measurements. The Kalman filter was developed under the following

assumptions:

- The initial state, x_0 is uncorrelated to both the system and measurement noise with known mean and covariance which are shown in equations 4.11 and 4.12:

$$\mu_0 = E[x_0] \quad (4.11)$$

$$P_0 = E[(x_0 - \mu_0)(x_0 - \mu_0)^T] \quad (4.12)$$

- The process noise, w_k , and measurement noise, v_k , are zero-mean, uncorrelated with known covariance matrices, equations 4.13 to 4.17.

$$E[w_k] = 0 \quad (4.13)$$

$$E[w_k w_j^T] = \begin{cases} Q_k & \text{if } k = j, \\ 0 & \text{if } k \neq j \end{cases} \quad (4.14)$$

$$E[v_k] = 0 \quad (4.15)$$

$$E[v_k v_j^T] = \begin{cases} R_k & \text{if } k = j, \\ 0 & \text{if } k \neq j \end{cases} \quad (4.16)$$

$$E[w_k v_j^T] = 0 \quad \text{for all } k \text{ and } j \quad (4.17)$$

In a Kalman filter, it is assumed that the noise is Gaussian and has the normal distribution with zero mean. So, the covariance is the only thing that needs to be calculated. Q and R are the covariance matrices of v_k and w_k respectively. These matrices are diagonal matrices which means all the non-diagonal elements have zero

value. The covariance matrix is a matrix whose elements are the variances of variables. For example, if the measurement noise vector, v_k , has m elements $v_1, v_2, \dots, v_m$ and $\sigma_1^2, \sigma_2^2, \dots, \sigma_m^2$ are the variances of each noise components respectively, then covariance matrix R is calculated as equation 4.18:

$$R = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_m^2 \end{bmatrix} \quad (4.18)$$

As it was mentioned earlier, the Prediction or Time Update is the first step of filter. The Time Update is responsible to predict the state of the system and error covariance ahead in time. The state of the system is a vector that contains as elements the variables of interest. The results of this step are called "priori estimates". The equations 4.19 and 4.20 are the two prediction questions:

$$\hat{x}_k^- = A.\hat{x}_{k-1} + B.u_{k-1} \quad (4.19)$$

$\hat{x}_k^-$ is the prediction of the system for the time k , while the system is at time $k - 1$.

$$P_k^- = A.P_{k-1}.A^T + Q \quad (4.20)$$

At this step, the filter projects the error covariance estimates, P_k^- , from the time step $k - 1$ to the time step k .

At the Correction step, the filter utilizes the result from the prediction step in combination with measurements received at time k to update its first prediction. The results of the Measurement Update step are called a "posteriori" estimates. At the beginning of the Measurement Update step the filter gain, which is called

Kalman gain, K_k , is calculated as equation 4.21:

$$K_k = P_k^- . H^T . (H . P_k^- . H^T + R)^{-1} \quad (4.21)$$

The Kalman gain is the variable weighting factor that is updated in each time step. It is used to update the state of the system, combining the new measurement, z_k , with the state prediction, $\hat{x}_k^-$, as equation 4.22:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (4.22)$$

From the above formula, it can be seen that the corrected state of the system is the predicted state modified by a correction component. The correction component is correlated to the disparity between the predicted state and the measured state. The term $(z_k - H\hat{x}_k^-)$ is known as the measurement residual and indicates the discrepancy between the measurement of system at time k and the prediction of the system for time k. The H is the matrix that relates the state of the system to the measurement.

The final step of the Measurement Update is the error covariance update. The error covariance reflects the degree of precision of the estimator. Larger error covariance indicates a larger estimator error and a small error covariance implies a small error in the estimation process. The error covariance is updated as equation 4.23:

$$P_k = (I - K_k H) P_k^- \quad (4.23)$$

As stated earlier, in the Prediction step, the error covariance was predicted. Then in the correction step, the predicted error covariance was utilized to compute the Kalman gain. The recursive flow chart of the discrete Kalman filter is shown in Figure 4.3.

Figure 4.3: Discrete Kalman Filter Flow Chart

4.6 Quaternion-based Kalman Filter

As it was described in Chapter 2, three dimensional rotation can be represented using quaternions. In our case, there are two frames including the Earth frame and the hand-fixed frame. If we called the Earth frame, frame A and the hand frame, frame B, the orientation of hand frame relative to Earth frame can be calculated through a rotation around vector a in Earth frame. These two reference frames are depicted in Figure 4.4.

Figure 4.4: Hand and Inertial Reference Frames

The quaternion describing the orientation of frame B with respect to frame A is defined as equation 4.24:

$${}^B_A\hat{q} = [q_0, q_1, q_2, q_3] = \left[\cos \frac{\theta}{2}, -a_x \sin \frac{\theta}{2}, -a_y \sin \frac{\theta}{2}, -a_z \sin \frac{\theta}{2} \right] \quad (4.24)$$

Where θ is the rotation angle around a vector defined in frame A. a_x , a_y and a_z are components of a unit vector r in the Earth frame. Interestingly, the orientation of Earth frame(A) relative to sensor frame(B) is the conjugate of the orientation of the sensor frame relative to the inertial frame. The conjugate is defined as equation 4.25:

$${}^B_A\hat{q} = {}^A_B\hat{q} = [q_0, -q_1, -q_2, -q_3] \quad (4.25)$$

Another property of quaternions, used for indicating rotation, is expressing a compound quaternion as a quaternion product. For instance, for the orientations ${}^B_A\hat{q}$, the compound orientation is defined as:

$${}^B_A\hat{q} = {}^B_C\hat{q} \otimes {}^C_A\hat{q} \quad (4.26)$$

If v is a vector in frame A, this vector can be represented in coordinate frame B by rotating the vector about a quaternion q as:

$$v_{rotated} = {}^B_A\hat{q} \otimes v \otimes {}^A_B\hat{q}^* \quad (4.27)$$

For the equation 4.27 vector v is written in quaternion form so that it contains zero for the first element:

$$v = [0, v_x, v_y, v_z] \quad (4.28)$$

The rotation matrix is written as equation 4.29 :

$$R_Q = \begin{bmatrix} 2q_0^2 - 1 + q_1^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2q_0^2 - 1 + q_2^2 & 2(q_3q_2 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 2q_0^2 - 1 + q_3^2 \end{bmatrix} \quad (4.29)$$

4.6.1 Prediction Step

The tri-axis gyroscope data is used as an input for the prediction step. The gyroscope records the angular rate about x-, y- and z-axis of the sensor frame. These angular

rotation rates will be symbolized with w_x , w_y and w_z notations. To represent the angular rate in quaternion format, the following vector is defined [80].

$$w_{Sensor} = \begin{bmatrix} 0 & w_x & w_y & w_z \end{bmatrix} \quad (4.30)$$

The objective is to calculate the angular change in reference to the Earth frame using the angular rate in sensor frame representation. The equation describing the rate of change of orientation in Earth frame relative to the change in sensor frame can be written as equation 4.31 :

$${}^S_E \dot{\hat{q}} = \left(\frac{1}{2}\right) {}^S_E \hat{q} \otimes w_{Sensor} \quad (4.31)$$

Where, ${}^S_E \dot{\hat{q}}$ is the rate of rotational change of the sensor frame relative to the Earth frame, and ${}^S_E \hat{q}$ is the normalized previous estimate of orientation.

The rate of change of orientation in the sensor frame relative to the Earth frame at time t is calculated as equation 4.32 :

$${}^S_E \dot{\hat{q}}_t = \left(\frac{1}{2}\right) {}^S_E \hat{q}_{t-1} \otimes w_{Sensor_t} \quad (4.32)$$

By integrating the rotational variations, ${}^S_E \dot{\hat{q}}_t$, over time the orientation of the sensor frame relative to Earth frame at time t is calculated as:

$${}^S_E \hat{q}_t = {}^S_E \hat{q}_{t-1} + {}^S_E \dot{\hat{q}}_t \Delta t \quad (4.33)$$

Where Δt is the sampling period and the state of system is ${}^S_E \hat{q}$.

Having ${}^S_E \hat{q}$ as the state of the system, the equation 4.33 can be rewritten as:

$$x_t = x_{t-1} + {}^S_E \dot{\hat{q}}_t \Delta t = A.x_{t-1} \quad (4.34)$$

State Transition Matrix

A is the state transition matrix and is defined as equation 4.35 :

$$A = \begin{bmatrix} 1 & -\frac{1}{2} \cdot \Delta t \cdot w_x & -\frac{1}{2} \cdot \Delta t \cdot w_y & -\frac{1}{2} \cdot \Delta t \cdot w_z \\ \frac{1}{2} \cdot \Delta t \cdot w_x & 1 & \frac{1}{2} \cdot \Delta t \cdot w_z & -\frac{1}{2} \cdot \Delta t \cdot w_y \\ \frac{1}{2} \cdot \Delta t \cdot w_y & -\frac{1}{2} \cdot \Delta t \cdot w_z & 1 & \frac{1}{2} \cdot \Delta t \cdot w_x \\ \frac{1}{2} \cdot \Delta t \cdot w_z & \frac{1}{2} \cdot \Delta t \cdot w_y & -\frac{1}{2} \cdot \Delta t \cdot w_x & 1 \end{bmatrix} \quad (4.35)$$

Process Noise Covariance Matrix

To describe the process noise , the covariance matrix of the input vector should be calculated. The input vector is:

$$X = \begin{bmatrix} 0 \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad (4.36)$$

For the vector X , the covariance matrix which is a 4×4 diagonal matrix is calculated as:

$$Q = E((X - E(X))(X - E(X))^T) \quad (4.37)$$

4.6.2 Correction Step

The sensor unit records nine-dimensional data from the gyroscope, the magnetometer and the accelerometer. If the nine-dimensional data is used as an observation vector, the measurement equation would be nonlinear and a heavy computational process would be needed. Therefore, it was suggested [93] that just data recorded by the accelerometer and the magnetometer be used as observation data to determine the orientation. The quaternion vector which corresponds to the orientation of

the sensor frame relative to the Earth frame is calculated using accelerometer and magnetometer data.

The magnetometer records the direction and magnitude of Earth's magnetic field and the accelerometer measures the gravity direction in the sensor reference frame. The orientation of the sensor frame relative to the Earth frame can be extracted using the recorded field's direction in the sensor frame. However for any sensor measurement, a unique sensor's orientation can not be obtained. It is necessary to compute a unique true quaternion orientation to be used as the observation vector in the Kalman filter. To obtain such quaternion, the use of optimization algorithms [55] was suggested. In this research two optimization algorithms: Gradient Descent optimization and Gauss-Newton optimization methods, were chosen to calculate the quaternion vector from accelerometer and magnetometer data. Figure 4.5 depicts the block diagram for the quaternion-based Kalman filter using an optimization algorithm to calculate the observation vector.

Figure 4.5: Discrete Kalman Filter Block Diagram

4.6.3 Observation Vector Using Gradient Descent Optimization

Gradient Descent is an optimization algorithm to find a local minimum of functions. This algorithm starts with an initial value of the solution and iteratively moves to values that minimize the cost function.

In our case, the process seeks an optimized solution for the orientation of the sensor relative to Earth frame, ${}^S_E\hat{\mathbf{q}}$, using a predefined direction of Earth field ${}^E\hat{\mathbf{d}}$, and the field direction measured in the sensor frame ${}^S\hat{\mathbf{S}}$, as:

$$\min f({}^S_E\hat{\mathbf{q}}, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{S}}) \quad (4.38)$$

Where the objective function is described as equation 4.39 :

$$f({}^S_E\hat{\mathbf{q}}, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{S}}) = {}^S_E\hat{\mathbf{q}}^* \otimes {}^E\hat{\mathbf{d}} \otimes {}^S_E\hat{\mathbf{q}} - {}^S\hat{\mathbf{S}} \quad (4.39)$$

The vectors ${}^E\hat{\mathbf{d}}$ and ${}^S\hat{\mathbf{S}}$ and ${}^S_E\hat{\mathbf{q}}$ are defined as equations 4.40, 4.41 and 4.42 :

$${}^E\hat{\mathbf{d}} = \begin{bmatrix} 0 & d_x & d_y & d_z \end{bmatrix} \quad (4.40)$$

$${}^S\hat{\mathbf{S}} = \begin{bmatrix} 0 & s_x & s_y & s_z \end{bmatrix} \quad (4.41)$$

$${}^S_E\hat{\mathbf{q}} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix} \quad (4.42)$$

Using these three vectors, the function f would be:

$$f({}^S_E\hat{\mathbf{q}}, {}^E\hat{\mathbf{d}}, {}^S\hat{\mathbf{S}}) = \begin{bmatrix} 2d_x(\frac{1}{2} - q_3^2 - q_4^2) + 2d_y(q_1q_4 + q_2q_3) + 2d_z(q_2q_4 - q_1q_3) - s_x \\ 2d_x(q_2q_3 - q_1q_4) + 2d_y(\frac{1}{2} - q_2^2 - q_4^2) + 2d_z(q_1q_2 + q_4q_3) - s_y \\ 2d_x(q_1q_3 + q_2q_4) + 2d_y(q_3q_4 - q_1q_2) + 2d_z(\frac{1}{2} - q_2^2 - q_3^2) - s_z \end{bmatrix} \quad (4.43)$$

The Gradient Descent algorithm for the n iteration is defined as equation 4.44 :

$${}^S_E \mathbf{q}_{k+1} = {}^S_E \hat{\mathbf{q}}_k - \mu \frac{\nabla f({}^S_E \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{S}})}{\left\| \nabla f({}^S_E \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{S}}) \right\|}, k = 0, 1, 2, \dots, n \quad (4.44)$$

Where μ is the adaptation step size and ${}^S_E \hat{\mathbf{q}}_0$ is the initial value for orientation. The gradient of surface representing the objective function is defined as:

$$\nabla f({}^S_E \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{S}}) = J^T({}^S_E \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}) f({}^S_E \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}, {}^S \hat{\mathbf{S}}) \quad (4.45)$$

and

$$J({}^S_E \hat{\mathbf{q}}, {}^E \hat{\mathbf{d}}) = \begin{bmatrix} 2d_y q_4 - 2d_z q_3 & 2d_y q_3 + 2d_z q_4 & -4d_x q_3 + 2d_y q_2 - 2d_z q_1 & -4d_x q_4 + 2d_y q_1 + 2d_z q_2 & -2d_x q_1 - 4d_y q_4 + 2d_z q_3 & 2d_x q_2 + 2d_y q_3 \\ -2d_x q_4 + 2d_z q_2 & 2d_x q_3 - 4d_y q_2 + 2d_z q_1 & 2d_x q_2 + 2d_z q_4 & -2d_x q_1 - 4d_y q_4 + 2d_z q_3 & 2d_x q_2 + 2d_y q_3 & \\ 2d_x q_3 - 2d_y q_2 & 2d_x q_4 - 2d_y q_1 - 4d_z q_2 & 2d_x q_1 + 2d_y q_4 - 4d_z q_3 & & & \\ & & & & & \end{bmatrix} \quad (4.46)$$

The step-size μ is defined by equation 4.47 :

$$\mu_t = \alpha \left\| {}^S_E \dot{\mathbf{q}}_{w,t} \right\| \Delta t, \alpha > 1 \quad (4.47)$$

Where ${}^S_E \dot{\mathbf{q}}_{w,t}$ is the rotation rate calculated using gyroscope measurements, Δt is the sampling rate.

The objective function f , and the Jacobian J can be simplified if it is defined that the direction of field has components only in one or two axis of the Earth frame.

In utilized sensor unit, the gravity vector in Earth frame can be defined as:

$${}^E \hat{\mathbf{g}} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.48)$$

and accelerometer vector in the sensor frame is defined as:

$${}^S \hat{\text{accel}} = \begin{bmatrix} 0 & a_x & a_y & a_z \end{bmatrix} \quad (4.49)$$

Then the objective function f , and the Jacobian J can be rewritten as:

$$f_g({}^S_E \hat{\mathbf{q}}, {}^S \hat{\text{accel}}) = \begin{bmatrix} 2(q_1 q_4 + q_2 q_3) - a_x \\ 2(0.5 - (q_2)^2 - (q_4)^2) - a_y \\ 2(q_3 q_4 - q_1 q_2) - a_z \end{bmatrix} \quad (4.50)$$

$$J_g({}^S_E \hat{\mathbf{q}}, {}^E \hat{\mathbf{g}}) = \begin{bmatrix} 2q_4 & 2q_3 & 2q_2 & 2q_1 \\ 0 & -4q_2 & 0 & -4q_4 \\ -2q_2 & -2q_1 & 2q_4 & 2q_3 \end{bmatrix} \quad (4.51)$$

The magnetic vector in the Earth frame can be described as:

$${}^E \hat{\text{mag}} = \begin{bmatrix} 0 & 0 & b_y & b_z \end{bmatrix} \quad (4.52)$$

The magnetometer measurement vector in the sensor frame is represented as:

$${}^S \hat{\text{mag}} = \begin{bmatrix} 0 & m_x & m_y & m_z \end{bmatrix} \quad (4.53)$$

Then using these two vectors, the f and J can be written as:

$$f({}^S_E \hat{\mathbf{q}}, {}^E \hat{\mathbf{b}}, {}^S \hat{\text{mag}}) = \begin{bmatrix} 2b_y(q_1 q_4 + q_2 q_3) + 2b_z(q_2 q_4 - q_1 q_3) - m_x \\ 2b_y(0.5 - (q_2^2) - (q_4^2)) + 2b_z(q_1 q_2 + q_3 q_4) - m_y \\ 2b_y(q_3 q_4 - q_1 q_2) + 2b_z(0.5 - (q_2^2) - (q_3^2)) - m_z \end{bmatrix} \quad (4.54)$$

$$J({}_E^S \hat{\mathbf{q}}, {}^E \hat{\mathbf{b}}) = \begin{bmatrix} 2b_y q_4 - 2b_z q_3 & 2b_y q_3 + 2b_z q_4 & 2b_y q_2 - 2b_z q_1 \\ 2b_z q_2 & -4b_y q_2 + 2b_z q_1 & 2b_z q_4 \\ -2b_y q_2 & -2b_y q_1 - 4b_z q_2 & 2b_y q_4 - 4b_z q_3 \\ 2b_y q_1 + 2b_z q_2 \\ -4b_y q_4 + 2b_z q_3 \\ 2b_y q_3 \end{bmatrix} \quad (4.55)$$

Neither the accelerometer measurements nor the magnetometer measurements can not provide the unique attitude. To achieve a unique orientation, both measurements and references should be combined as:

$$f_{g,b}({}_E^S \hat{\mathbf{q}}, {}^S \hat{\mathbf{a}}, {}^E \hat{\mathbf{m}}, {}^S \hat{\mathbf{m}}) = \begin{bmatrix} f_g({}_E^S \hat{\mathbf{q}}, {}^S \hat{\mathbf{a}}) \\ f_b({}_E^S \hat{\mathbf{q}}, {}^E \hat{\mathbf{m}}, {}^S \hat{\mathbf{m}}) \end{bmatrix} \quad (4.56)$$

$$J_{g,b}({}_E^S \hat{\mathbf{q}}, {}^E \hat{\mathbf{m}}) = \begin{bmatrix} J_g^T({}_E^S \hat{\mathbf{q}}) \\ J_b^T({}_E^S \hat{\mathbf{q}}, {}^E \hat{\mathbf{m}}) \end{bmatrix} \quad (4.57)$$

4.6.4 Observation Vector Determination Using Gauss-Newton Method

The Gauss-Newton algorithm is a technique for solving non-linear least square problems. In this case, the Gauss-Newton algorithm is used to minimize the cost function which is defined by the differences between the known gravitational and Earth magnetic vectors and the sensors (accelerometer and magnetometer) readings transformed to the Earth frame.

The data from the accelerometer and the magnetometer are used as observation vectors. The accelerometer and magnetometer vectors in the body frame are

symbolized as equations 4.58 and 4.59 :

$$A_{Body} = (a_{xb}, a_{yb}, a_{zb}) \quad (4.58)$$

$$M_{Body} = (m_{xb}, m_{yb}, m_{zb}) \quad (4.59)$$

The same acceleration and magnetic vectors in the Earth frame are represented as equations 4.60 and 4.61 :

$$A_{Earth} = (0, g, 0) \quad (4.60)$$

$$M_{Earth} = (m_{xE}, m_{yE}, m_{zE}) \quad (4.61)$$

Combining the two vectors in each reference frame, the following measurement vectors are constructed:

$$y_{Earth} = (0, g, 0, m_{xE}, m_{yE}, m_{zE}) \quad (4.62)$$

$$y_{Body} = (a_{xb}, a_{yb}, a_{zb}, m_{xb}, m_{yb}, m_{zb}) \quad (4.63)$$

The rotation matrix that rotates the body vector to the Earth vector, in quaternion form, is expressed as equation 4.64 :

$$R_t = \begin{bmatrix} M_t & 0 \\ 0 & M_t \end{bmatrix} \quad (4.64)$$

Where matrix M is described as:

$$M_t = \begin{bmatrix} q_4^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & q_4^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_4q_1) \\ 2(q_1q_3 - q_2q_4) & 2(q_3q_2 + q_1q_4) & q_4^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix} \quad (4.65)$$

The Gauss-Newton optimization method is used to minimize discrepancy between actual and computed measurement vectors as equation 4.66 :

$$\epsilon = y_{Earth} - R_t \cdot y_{Body} \quad (4.66)$$

The Gauss-Newton method executes following iteration:

$$q_t = q_{t-1} - (J_t^T \cdot J_t)^{-1} \cdot J_t^T \cdot \epsilon \quad (4.67)$$

Where J_k is the Jacobian of ϵ calculated in q_k as is shown in equation 4.68 :

$$J_t(q_k(t)) = \frac{\partial(\epsilon)}{\partial(q_k(t))} = -\left[\left(\frac{\partial R}{\partial q_1} \cdot y_{Body(t)}\right)\left(\frac{\partial R}{\partial q_2} \cdot y_{Body(t)}\right)\left(\frac{\partial R}{\partial q_3} \cdot y_{Body(t)}\right)\left(\frac{\partial R}{\partial q_4} \cdot y_{Body(t)}\right)\right] \quad (4.68)$$

The computed Jacobian matrix computed would be:

$$J_t(q_k(t)) = -2 \begin{bmatrix} (q_1 a_{xb} + q_2 a_{yb} + q_3 a_{zb}) & (-q_2 a_{xb} + q_1 a_{yb} + q_4 a_{zb}) \\ (q_2 a_{xb} - q_1 a_{yb} - q_4 a_{zb}) & (q_1 a_{xb} + q_2 a_{yb} + q_3 a_{zb}) \\ (q_3 a_{xb} + q_4 a_{yb} - q_1 a_{zb}) & (-q_4 a_{xb} + q_3 a_{yb} - q_2 a_{zb}) \\ (q_1 m_{xb} + q + 2m_{yb} + q_3 m_{zb}) & (-q_2 m_{xb} + q_1 m_{yb} + q_4 m_{zb}) \\ (q_2 m_{xb} - q_1 m_{yb} - q_4 m_{zb}) & (q_1 m_{xb} + q_2 m_{yb} + q_3 m_{zb}) \\ (q_3 m_{xb} + q_4 m_{yb} - q_1 m_{zb}) & (-q_4 m_{xb} + q_3 m_{yb} - q_2 m_{zb}) \\ (-q_3 a_{xb} - q_4 a_{yb} + q_1 a_{zb}) & (q_4 a_{xb} - q_3 a_{yb} + q_2 a_{zb}) \\ (q_4 a_{xb} - q_3 a_{yb} + q_2 a_{zb}) & (q_3 a_{xb} + q_4 a_{yb} - q_1 a_{zb}) \\ (q_1 a_{xb} + q_2 a_{yb} + q_3 a_{zb}) & (-q_2 a_{xb} + q_1 a_{yb} + q_4 a_{zb}) \\ (-q_3 m_{xb} - q_4 m_{yb} + q_1 m_{zb}) & (q_4 m_{xb} - q_3 m_{yb} + q_2 m_{zb}) \\ (q_4 m_{xb} - q_3 m_{yb} + q_2 m_{zb}) & (q_3 m_{xb} + q_4 m_{yb} - q_1 m_{zb}) \\ (q_1 m_{xb} + q_2 m_{yb} + q_3 m_{zb}) & (-q_2 m_{xb} + q_1 m_{yb} + q_4 m_{zb}) \end{bmatrix} \quad (4.69)$$

CHAPTER 5

FILTER IMPLEMENTATION

5.1 Introduction

In this chapter, the implementation of sensor fusion algorithms will be described. The chapter includes details about data acquisition, attitude determination from the sensor fusion algorithm and the Matlab code for the process.

5.2 Human Hand Movements

As it was mentioned in chapter one, the aim of this research is to monitor the movement of the human hand. For this purpose, it is necessary to describe briefly the human hand movements. The movements of the palm of the hand are the result of wrist rotation. This research also focuses on monitoring the motion of the thumb. Therefore, the movements of the wrist and the thumb are described in the following sections.

5.2.1 Human Wrist Anatomy and Movements

The bones of the human wrist consist of eight carpal bones, which are arranged in two rows [83]. The carpal bones connect proximally to the radius of the forearm and distally to the five metacarpals of the hand [20]. The palm motion occurs by translating and rotating the carpal bones relative to each other and to bones proximal and distal to the wrist.

The International Society of Biomechanics (ISB) has defined anatomical terms for hand movements [89]. The global motions involving many bones were de-

fined, including wrist flexion-extension and radial-ulnar deviation, as well as forearm pronation-supination movements.

The pronation-supination movements happen when the human wrist rotates, while the palm's direction changes with no variation in the palm-to-arm or thumb-to-arm angles [79]. For the right hand, the counterclockwise motion is defined as pronation, and the clockwise motion is called supination. In pronation, the position of the palm changes from facing sideways to facing down. Figures 5.1 and 5.2 show pronation and supination movements. The range of maximum motion for pronation and supination together is 125° , with 60 percentage of this range allocated to pronation. These motion correspond to the rotation about X axis of the utilized sensor unit.

Figure 5.1: Hand Pronation Motion

The vertical movements of the human wrist are called flexion-extension. The vertical movement of the wrist when it moves upward is called extension and the downward vertical movement is called flexion. In the extension-flexion movements the angle between the palm of the hand and the arm changes. Figures 5.3 and 5.4 illustrate the extension and flexion movements. The maximum rotation for flexion

Figure 5.2: Hand Supination Motion

is 60° and for the extension is equal to 45° . The extension and flexion motions correspond to the rotation around the Z axis of the sensor unit.

Figure 5.3: Hand Extension Motion

The horizontal tilt of the wrist makes the thumb-arm angle change. This movement is called radial when the hand rotates toward the thumb side and ulnar when the hand rotates to the opposite side [83]. Figures 5.5 and 5.6 show the radial and ulnar movements.

Figure 5.4: Hand Flexion Motion

The rotation ranges for the radial and ulnar movements are 30° and 15° , respectively [65]. The radial and ulnar motions are traced by calculating the rotation about the Y axis sensor unit attached to the glove.

Figure 5.5: Hand Radial Motion

Figure 5.6: Hand Ulnar Motion

5.2.2 Human thumb Anatomy and Movements

The human thumb with its unique musculature, opposability, flexibility and extensibility is one of the exclusive features of the human body [2]. The architecture of the human thumb consists of two phalanx bones and one metacarpal bone. These bones are attached through joints. There are three joints in the human thumb: the trapeziometacarpal (TMC) joint, the metacarpophalangeal (MCP) joint and the interphalangeal (IP) joint [66]. These bones and joints are shown in Figure 5.7.

These joints enable the thumb to perform the flexion/extension and abduct/adduct motions. During flexion the thumb bends into the palm and extension designates the opposite movement. Abduction and adduction are movements that the thumb performs while remaining in the same plane as the palm. During abduction the thumb moves away from the index. Adduction is opposite to abduction, bringing the thumb closer to the index. These motions are depicted in Figures 5.8, 5.9, 5.10 and 5.11.

In the experiment, the sensor unit is attached on the upper phalanx bone. Although the thumb is able to bend in interphalangeal joint, the two phalanx bones

Figure 5.7: Human Thumb Bones

Figure 5.8: Thumb Flexion Motion

Figure 5.9: Thumb Extension Motion

Figure 5.10: Thumb Abduct Motion

Figure 5.11: Thumb Adduct Motion

are considered as a one integrated section and the system tracks the upper phalanx bone's position with respect to the palm. The flexion and the extension of the thumb do not happen in a plane, so these motions results in rotation about more than one axis of sensor.

5.3 System Description

The motion monitoring system developed for this research consists of two sensor units mounted on the palm and the thumb finger of a glove. Each sensor unit incorporates a three-axis MEMS accelerometer, a three-axis MEMS gyroscope and a three axis MEMS magnetometer. The sensor units send data to the PC. The sensor unit is shown in Figure 5.12.

These sensors capture the external motion forces acting on them. These forces are related to linear acceleration and angular rotation. These sensors are available in miniature size, provide an acceptable sampling rate and have low power consumption, all of which make them suitable for human motion detection applications.

Figure 5.12: Sensor Unit

The sensor specifications are shown in tables 5.1, 5.2 and 5.3. In the experiment the magnetometer scale is ± 1.3 Ga, accelerometer range ± 2 g and the sensitivity is 0.00024g/digit. The gyroscope sensitivity and scale are $0.00833^\circ/\text{sec}/\text{digit}$ and ± 250 respectively. The data are sampled at a rate of 213 samples per second.

Gyroscope Specifications	
Resolution	16 bit
Bias Stability	$2.5^\circ/\text{hour}$ average
Sensitivity	$0.00833^\circ/\text{sec}/\text{digit}$ for $\pm 250^\circ/\text{sec}$ and $0.0666^\circ/\text{sec}/\text{digit}$ for $\pm 2000^\circ/\text{sec}$
Non-linearity	0.2% full scale
Temperature Sensitivity	$\pm 0.03\%/^\circ\text{C}$
Noise Density	$0.009^\circ/\text{sec}/\sqrt{\text{Hz}}$
Scale	$\pm 250/\pm 500/\pm 1000/\pm 2000^\circ/\text{sec}$

Table 5.1: Gyroscope Specifications

Accelerometer Specifications	
Resolution	14 bit, 12 bit
Scale	$\pm 2g, \pm 4g, \pm 8g$ for standard mode
Sensitivity	0.00024g/digit-0.00096g/digit
Noise Density	99g/ $\sqrt{Hz}$
Temperature Sensitivity	$\pm 0.008\%/^{\circ}C$
Sensitivity	0.00024g/digit-0.00096g/digit

Table 5.2: Accelerometer Specifications

Magnetometer Specifications	
Resolution	12 bit
Sensitivity	0.73 mGa/digit
Non-linearity	0.1% full scale
Scale	± 0.99 Ga to ± 8.1 Ga selectable

Table 5.3: Magnetometer Specifications

5.4 Data Acquisition

As it was mentioned in the previous section, the data from the sensors are collected by a PC. The data collected from the sensors, prior to the calibration process, are called raw data. Figure 5.13 shows the raw data extracted from sensors while sensors are in a stationary position. The MATLAB code to open the collected data is shown in Listing 5.1.

Listing 5.1: Data Extraction Code

```
% *****Data Extraction
% *****
close all;
% Data Reading
fid=fopen('Myfile.txt');
tline=fgets(fid);
```

```

d1=textscan(fid,'%f %f,%f,%f %f,%f,%f %f,%f,%f','delimiter','
    \n');
fclose(fid);
%convert cell format to matrix format
raw_out1=cell2mat(d1);
%computing the number of rows. X shows number of samples.
[x1,y1]=size(raw_out1);
sprintf('size of samples %f',x1)
%time
time1 =raw_out1(1:x1,1);
% Gyroscope Data
gyro1=raw_out1(1:x1,2:4);
% Accelerometer Data
accel1=raw_out1(1:x1,5:7);
% Magnetometer Data
compass1=raw_out1(1:x1,8:10);
% Plot the Data
figure;subplot(3,1,1);plot(accel1)
legend('X','Y','Z')
title('Accelerometer data');
subplot(3,1,2);plot(gyro1)
title('Gyroscope data');
subplot(3,1,3);plot(compass1)
title('Magnetometer data')
xlabel('samples')

```

Figure 5.13: Raw Sensor Data Example

5.5 Calibration Process

The raw data need to be corrected by a calibration process. The calibration process turns the raw data into the usable data. In the following, calibration process for each of the sensors will be described.

Gyroscope

The calibration equation for the gyroscope output can be expressed as equation 5.1 :

$$G_c = [M].(S.[G_r + b]) \quad (5.1)$$

where;

G_c is the calibrated data, G_r is raw data, S is the scale factor and b represents the

bias. In this experiment, the gyroscope calibration was performed as:

$$\begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \cdot (0.7) \cdot \left(\frac{\pi}{180} \right) \cdot \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} + \begin{bmatrix} 6.3930 \\ -37.9940 \\ 16.8820 \end{bmatrix} \quad (5.2)$$

where; G_x , G_y and G_z are calibrated gyroscope data in the X , Y and Z axes, and g_x , g_y and g_z are raw gyroscope data in the X , Y and Z axes. The results of calibration can be seen in Figure 5.14.

Figure 5.14: Calibrated Gyroscope Data

Note that even the calibrated gyroscope data are not reliable to use for calculating the rotation. The reason is that these data still contain stochastic noise. To prove the existence of stochastic noise in the calibrated gyroscope data, rotational angles were computed from gyroscope data while the sensor was stationary. As it is shown in equation 5.3, the trapezoidal integration method can be used to calculate

the rotation angle:

$$\int_a^b f(x)dx = (b-a)f(a) + \frac{1}{2}(b-a)(f(b) - f(a)) \quad (5.3)$$

The results of integration are shown in Figure 5.15. As it can be seen, the calculated results drift over time. The explanation for this phenomenon is that the integration accumulates the noise over time and turns noise into the drift, which yields unacceptable results.

Figure 5.15: Rotational Angles calculated by Gyroscope Data

Accelerometer

The following process was performed to calibrate the accelerometer data:

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} 1.0007 & 0.0234 & 0.0117 \\ -0.0324 & 0.9900 & 0.0244 \\ -0.0065 & 0.0037 & 1.0041 \end{bmatrix} \cdot \left(\frac{1}{16384} \right) \cdot \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} 200.3147 \\ 136.3133 \\ -83.4737 \end{bmatrix} \quad (5.4)$$

where; A_x , A_y and A_z are calibrated accelerometer data in the X , Y and Z axes, and a_x , a_y and a_z are raw accelerometer data in the X , Y and Z axes. Calibrated

accelerometer data is shown in Figure 5.16. The roll and pitch rotations can be

Figure 5.16: Calibrated Accelerometer Data

obtained from the accelerometer data. Unlike gyroscope results the results from the accelerometer, which are shown in figure 5.17, do not drift but they are noisy and not reliable. To calculate the pitch and roll rotations, equations (5.5) and (5.6) are used:

$$Pitch = \arctan\left(\frac{A_x}{A_y^2 + A_z^2}\right) \quad (5.5)$$

$$Roll = \arctan\left(\frac{A_y}{A_x^2 + A_z^2}\right) \quad (5.6)$$

Figure 5.17: Rotational Angles Using Accelerometer Data

Magnetometer

The calibration process for the magnetometer was performed as:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 0.9409 & -0.0484 & 0.0075 \\ 0.0111 & 1.0518 & -0.0039 \\ 0.0441 & 0.0047 & 0.9426 \end{bmatrix} \cdot \left(\frac{1}{1090} \right) \cdot \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} + \begin{bmatrix} -31.7043 \\ -10.7546 \\ 71.9295 \end{bmatrix} \quad (5.7)$$

where; M_x , M_y and M_z are calibrated magnetometer data in the X , Y and Z axes, and m_x , m_y and m_z are raw magnetometer data in the X , Y and Z axes.

Figure 5.18 shows the results of the magnetometer calibration process.

The tri-axis magnetometer projects the Earth magnetic field vector in three orthogonal vectors. This data can be described in the following two ways:

- The magnetic field can be described by three orthogonal components. In this arrangement, the positive values point northward in the X -axis, eastward in

Figure 5.18: Calibrated Magnetometer Data

the Y-axis and downward in the Z-axis. This arrangement uses the right-hand system to represent the magnetic field. Figure 5.19 illustrates this representation of the magnetic field. The total field strength is calculated as equation 5.8:

$$F = \sqrt{X^2 + Y^2 + Z^2} = \sqrt{H^2 + Z^2} \quad (5.8)$$

- In the second representation the components include the horizontal magnitude (H), the declination angle (D) and the inclination angle (I). The inclination angle is the angle between horizontal plane and the field vector, which is measured positive downwards. The declination and the inclination angles can be computed by equations 5.11 and 5.12:

$$X = H \cos(D) \quad (5.9)$$

$$Y = H \sin(D) \quad (5.10)$$

$$D = \arctan\left(\frac{Y}{X}\right) \quad (5.11)$$

$$I = \arctan\left(\frac{Z}{H}\right) \quad (5.12)$$

Figure 5.19: Earth Magntic Vector

The recorded magnetometer raw data should be corrected for soft and hard iron effects.

Magnetometer Soft and Hard Iron Compensation

As it was explained in Chapter Three, soft and hard iron effects can affect the accuracy of MEMS magnetometers. Therefore, it is necessary to compensate for these errors in a calibration process. Magnetometer which has been properly compensated for soft and hard iron effects is turned around the vertical axis for 360 degrees, in a horizontal surface, it will yield X and Y data values that will create a circle, centered at zero when plotted [73]. Figure 5.20 depicts the ideal plot of X axis data with respect to the Y axis data when hard and soft iron disturbances are removed.

Presence of hard iron error would shift the center of the circle, as it is shown in Figure 5.21. In fact the hard iron effect adds offset to the data. This offset can be

Figure 5.20: Unaffected Magnetometer

compensated in the calibration process with measuring the maximum and minimum values for each axis after turning the sensor about 360 degree. If the maximum and minimum values for each specific axis are the same with different signs, then there is not hard iron error. Otherwise the offset is calculated and removed with following steps, for each axis:

- $sum = data_{maximumvalue} + data_{minimumvalue}$
- $bias = sum/2$
- $data = data - bias$

Figure 5.21: Hard Iron Effect on Magnetometer

Soft Iron disturbance would disturb the circle into an elliptic shape, Figure 5.22. It is recommended to carry out the hard iron and tilt correction prior to performing the soft iron correction. In this way the origin of the ellipse is located at the center, (Figure 5.23). A simple approach to make up for soft iron is presented here [48].

Figure 5.22: Soft Iron Effect on Magnetometer

First the rotation angle from X axis is calculated as:

$$r = \sqrt{(x_1^2 + y_1^2)} \quad (5.13)$$

$$\theta = \arcsin\left(\frac{y_1}{r}\right) \quad (5.14)$$

Now, the rotation matrix is defined as equation 5.15:

$$R = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (5.15)$$

Now the ellipse should be turned using rotational matrix, equation 5.16. Once the rotation is done the major and minor axes of the ellipse will be aligned with the X and Y axes.

$$\hat{v} = Rv \quad (5.16)$$

The next step is to scale the major axis of the ellipse to reshape it into a circle. The scale factor is calculated as equation 5.17:

$$S = \frac{q}{r} \quad (5.17)$$

Figure 5.23: Soft Iron Effect on Magnetometer

To test the magnetometer contained in the device used for this research, the horizontal 360 degree rotation was performed and the results are depicted in Figure 5.24). It can be clearly realized that the data is disturbed with both soft and hard iron errors (The tilt compensation was not performed yet). In other test the

Figure 5.24: X axis with respect to Y axis Uncompensated Magnetometer Data

magnetometer was moved randomly in different orientations and data were collected

(Figure 5.25). It can be seen that the data lie on an ellipsoid instead of a sphere. To correct the data the ellipsoid form should be transformed into a sphere. In [94] a MATLAB program is provided to fit an ellipsoid to a sphere. The corrected results are shown in Figures 5.26 and 5.27.

Figure 5.25: Soft and Hard Iron Effect on Magnetometer Data

Figure 5.26: Soft and Hard Iron Error Compensated 2D Magnetometer Data

Figure 5.28 shows the magnetometer data after the calibration process.

Figure 5.27: Sphere-Fitted Calibrated Magnetometer Data

Figure 5.28: Three Axis Calibrated Magnetometer Data

5.6 Test of Gauss-Newton Algorithm

The Gauss-Newton algorithm was written in MATLAB. The code for this algorithm is shown in Listing 5.2. As it can be seen in the code, inputs of this function are the accelerometer data, the magnetometer data and also a quaternion which was calculated in prediction step of the Kalman filter. The following vectors are used as reference vectors:

$$Earth - magnetic_{ref} = (0, -0.03751, 0.92696) \quad (5.18)$$

$$Gravity_{ref} = (0, 1, 0) \quad (5.19)$$

$$Reference_{vector} = (0, 1, 0, 0, -0.03751, 0.92696); \quad (5.20)$$

Listing 5.2: Newton-Gauss Code

```
function [ output ] = MyNewtonGauss(Q, accel, compass)

q1=Q(1,1); q2=Q(2,1); q3=Q(3,1); q4=Q(4,1);
Q_now=[q1;q2;q3;q4];
Q_now=Q_now/norm(Q_now);
q1=Q_now(1,1);
q2=Q_now(2,1);
q3=Q_now(3,1);
q4=Q_now(4,1);
temp=Q_now;

for i=1:3
    %*****Compute M Matrix*****
```



```

% M matrix includes of two direct cosine matrix to rotate
    accel and mag

% data
xx1=q4^2+q1^2-q2^2-q3^2; xx2=2*(q1*q2-q3*q4); xx3=2*(q1*
    q3+q2*q4);
xx4=2*(q1*q2+q3*q4); xx5=q4^2+q2^2-q1^2-q3^2; xx6=2*(q2*
    q3-q1*q4);
xx7=2*(q1*q3-q2*q4); xx8=2*(q2*q3+q1*q4); xx9=q4^2+q3^2-
    q2^2-q1^2;

X_matrix=[xx1 xx2 xx3;xx4 xx5 xx6;xx7 xx8 xx9];
XX=[X_matrix zeros(3,3);zeros(3,3) X_matrix];

%*****Jacubian Computation*****

a_x=accel(1,1);a_y=accel(2,1);a_z=accel(3,1);
m_x=compass(1,1); m_y=compass(2,1); m_z=compass(3,1);

j1=(2*q1*a_x+2*q2*a_y+2*q3*a_z);
j2=(-2*q2*a_x+2*q1*a_y+2*q4*a_z);
j3=(-2*q3*a_x-2*q4*a_y+2*q1*a_z);
j4=(2*q4*a_x-2*q3*a_y+2*q2*a_z);

j5=(2*q2*a_x-2*q1*a_y-2*q4*a_z);
j6=(2*q1*a_x+2*q2*a_y+2*q3*a_z);
j7=(2*q4*a_x-2*q3*a_y+2*q2*a_z);
j8=(2*q3*a_x+2*q4*a_y-2*q1*a_z);

```

```

j9=(2*q3*a_x+2*q4*a_y-2*q1*a_z);
j10=(-2*q4*a_x+2*q3*a_y-2*q2*a_z);
j11=(2*q1*a_x+2*q2*a_y+2*q3*a_z);
j12=(-2*q2*a_x+2*q1*a_y+2*q4*a_z);

j13=(2*q1*m_x+2*q2*m_y+2*q3*m_z);
j14=(-2*q2*m_x+2*q1*m_y+2*m_z*q4);
j15=(-2*q3*m_x-2*q4*m_y+2*q1*m_z);
j16=(2*q4*m_x-2*q3*m_y+2*q2*m_z);

j17=(2*q2*m_x-2*q1*m_y-2*q4*m_z);
j18=(2*q1*m_x+2*q2*m_y+2*q3*m_z);
j19=(2*q4*m_x-2*q3*m_y+2*q2*m_z);
j20=(2*q3*m_x+2*q4*m_y-2*q1*m_z);

j21=(2*q3*m_x+2*q4*m_y-2*q1*m_z);
j22=(-2*q4*m_x+2*q3*m_y-2*q2*m_z);
j23=(2*q1*m_x+2*q2*m_y+2*q3*m_z);
j24=(-2*q2*m_x+2*q1*m_y+2*q4*m_z);

Jacobian_matrix=-[j1 j2 j3 j4;j5 j6 j7 j8;j9 j10 j11 j12;
j13 ...
j14 j15 j16;j17 j18 j19 j20;j21 j22 j23 j24];
%*****Define Reference Attitude*****
Earthframe=[0;1;0;0;-0.03751;0.92696];
Bodyframe=[a_x;a_y;a_z;m_x;m_y;m_z];

f=Earthframe-(XX*Bodyframe);

```

```

%*****Gauss Newton *****

Q_next=Q_now-((Jacobian_matrix'*Jacobian_matrix)^-1)*...
    (Jacobian_matrix)*f;
Q_next=Q_next/norm(Q_next);

q1=Q_next(1,1);q2=Q_next(2,1);q3=Q_next(3,1);q4=Q_next
    (4,1);
Q_now=Q_next;
temp=[temp,Q_now];
end

output=[Q_next(1,1);Q_next(2,1);Q_next(3,1);Q_next(4,1)];
end

```

Gauss-Newton Algorithm Convergence

Some tests were performed to find the number of loops needed for the algorithm to converge. To test the convergence, a six element body data and an arbitrary quaternion was chosen and were utilized as the accelerometer data, the magnetometer data and predicted quaternions. These values were chosen as:

$$\textit{Bodyframe} = (-0.01, 0.988, 0.1, 0, -0.02751, 0.97696) \quad (5.21)$$

$$\textit{Predicted - quaternion} = (0.1, -0.05, 0.4, 0.63) \quad (5.22)$$

Figure 5.29 depicts the convergence of the Gauss-Newton algorithm utilizing the selected values. As it can be seen, this algorithm converges after three iterations. The utilized data have measurement noise variance $6.4792e^{-6}$.

Figure 5.29: Newton-Gauss Algorithm Convergence

In other test, the convergence of Gauss-Newton algorithm in different rotations was checked. A vector was rotated with different rotational angles in the presence of measurement noise. The algorithm converged after three iterations for all rotational angles. Different rotations were performed about the axis $(-2, 1, 4)$. The variance of measurement noise was $6.4792e^{-06}$ for all the cases. The results are shown in Table 5.4.

The number of iterations for the algorithm to converge depended on the measurement noise. Hence, different measurement variances were tested to observe the convergence of the algorithm. Results are depicted in Figure 5.30.

This experiment revealed that convergence of Gauss-Newton algorithm is influenced by sensor measurement noise. To guarantee the algorithm to converge, the iteration number was fix for ten loops.

Angle	Variance	Predicted Quaternion				Converged Quaternion			
5	6.4792e-06	-0.0398	-0.0107	0.0184	0.9990	-0.0190	0.0095	0.0381	0.9990
20	6.4792e-06	-0.0979	0.0183	0.1344	0.9859	-0.0758	0.0379	0.1516	0.9848
35	6.4792e-06	-0.1547	0.0467	0.2480	0.9552	-0.1312	0.0656	0.2625	0.9537
50	6.4792e-06	-0.2093	0.0739	0.3572	0.9073	-0.1844	0.0922	0.3689	0.9063
65	6.4792e-06	-0.2606	0.0996	0.4597	0.8431	-0.2345	0.1172	0.4690	0.8434
80	6.4792e-06	-0.3077	0.1231	0.5539	0.7638	-0.2805	0.1403	0.5611	0.7660
95	6.4792e-06	-0.3497	0.1441	0.6380	0.6708	-0.3218	0.1609	0.6435	0.6756
110	6.4792e-06	-0.3859	0.1623	0.7104	0.5658	-0.3575	0.1788	0.7150	0.5736
125	6.4792e-06	-0.4155	0.1771	0.7698	0.4509	-0.3871	0.1936	0.7742	0.4617
140	6.4792e-06	-0.4382	0.1885	0.8153	0.3281	-0.4101	0.2051	0.8202	0.3420
155	6.4792e-06	-0.4535	0.1963	0.8461	0.1997	-0.4261	0.2130	0.8522	0.2164
170	6.4792e-06	-0.4613	0.2002	0.8617	0.0681	-0.4348	0.2174	0.8696	0.0872
185	6.4792e-06	-0.4613	0.2003	0.8619	-0.0644	-0.4360	0.2180	0.8720	-0.0436
200	6.4792e-06	-0.4537	0.1966	0.8469	-0.1953	-0.4298	0.2149	0.8596	-0.1736
215	6.4792e-06	-0.4387	0.1892	0.8171	-0.3225	-0.4162	0.2081	0.8325	-0.3007
230	6.4792e-06	-0.4166	0.1782	0.7731	-0.4438	-0.3955	0.1978	0.7911	-0.4226
245	6.4792e-06	-0.3879	0.1640	0.7158	-0.5570	-0.3681	0.1840	0.7362	-0.5373
260	6.4792e-06	-0.3531	0.1466	0.6463	-0.6604	-0.3343	0.1672	0.6687	-0.6428
275	6.4792e-06	-0.3128	0.1266	0.5659	-0.7523	-0.2949	0.1474	0.5897	-0.7373
290	6.4792e-06	-0.2678	0.1042	0.4761	-0.8312	-0.2503	0.1252	0.5007	-0.8192
305	6.4792e-06	-0.2188	0.0798	0.3783	-0.8959	-0.2015	0.1008	0.4030	-0.8870
320	6.4792e-06	-0.1668	0.0538	0.2744	-0.9455	-0.1493	0.0746	0.2985	-0.9397
335	6.4792e-06	-0.1125	0.0268	0.1660	-0.9793	-0.0945	0.0472	0.1889	-0.9763
350	6.4792e-06	-0.0569	-0.0010	0.0550	-0.9969	-0.0380	0.0190	0.0761	-0.9962

Table 5.4: Convergence of Gauss-Newton for Different Rotations

Figure 5.30: Iteration Convergence Number for Different Noises

5.7 Test of Gradient Descent Algorithm

The MATLAB code for Gradient Descent algorithm can be seen in Listing 5.3. This function has four inputs including the the accelerometer data, the magnetometer data, a predicted quaternion from the prediction phase of the Kalman filter, and the step size, μ . As it can be seen in the code, the reference vectors utilized are same as the reference vectors which were used in the Gauss-Newton algorithm.

Listing 5.3: Gradient Descent Code

```
function [optimized_Q] = MyGradientDescent(A_s,M_s,Q_predict,
    Mu)
% in this function quaternion order is as[real,i,j,k]
temp=Q_predict;
Q_predict(1,1)=temp(4,1);
Q_predict(2,1)=temp(1,1);
Q_predict(3,1)=temp(2,1);
Q_predict(4,1)=temp(3,1);
temp1=[];
b=[0;-0.03751;0.92696];

for i=1:20
    Q_predict=Q_predict/norm(Q_predict);
    q1=Q_predict(1,1);
    q2=Q_predict(2,1);
    q3=Q_predict(3,1);
    q4=Q_predict(4,1);
```

```

temp1=[temp1;Q_predict(2,1) Q_predict(3,1) Q_predict(4,1)
...
Q_predict(1,1)];

%***** F MATRIX *****

% F_g, [0,dx,dy,dz]=[0,0,1,0]; Assuming there is just
accelerometer
F_g=[2*(q1*q4+q2*q3)-A_s(1,1);...
2*(0.5-(q2)^2-(q4)^2)-A_s(2,1);...
2*(q3*q4-q1*q2)-A_s(3,1)];

% F_b, b=[0;-0.03751;0.92696]=[0,0,by,bz
];0;-0.03751;0.92696]; Assuming
% there is just magnetometer.

F_b=[2*b(2,1)*(q1*q4+q2*q3)+2*b(3,1)*(q2*q4-q1*q3)-M_s
(1,1);...
2*b(2,1)*(0.5-(q2^2)-(q4^2))+2*b(3,1)*(q1*q2+q3*q4)-
M_s(2,1);...
2*b(2,1)*(q3*q4-q1*q2)+2*b(3,1)*(0.5-(q2^2)-(q3^2))-
M_s(3,1)];

%***** J MATRIX *****

% J_g, [0,dx,dy,dz]=[0,0,1,0] % assuming there is just
accel

```

```

J_g=[2*q4 2*q3 +2*q2 2*q1;...
      0 -4*q2 0 -4*q4;...
      -2*q2 -2*q1 2*q4 2*q3];

% J_b, b=[0;-0.03751;0.92696]=[0,0,by,bz]; Assuming
% there is just magnetometer.
J_b=[2*b(2,1)*q4-2*b(3,1)*q3 2*b(2,1)*q3+2*b(3,1)*q4...
      2*b(2,1)*q2-2*b(3,1)*q1 2*b(2,1)*q1+2*b(3,1)*q2;
      2*b(3,1)*q2 -4*b(2,1)*q2+2*b(3,1)*q1 2*b(3,1)*q4 ...
      -4*b(2,1)*q4+2*b(3,1)*q3;
      -2*b(2,1)*q2 -2*b(2,1)*q1-4*b(3,1)*q2 ...
      2*b(2,1)*q4-4*b(3,1)*q3 2*b(2,1)*q3];

% Combination the measurement of gravity and Earth's
% magnetic field
F_gb=[F_g;F_b];
J_gb=[J_g;J_b];
% Function gradient
fun_gra=J_gb'*F_gb;

% Estimated orientation at time t
Q_estimated=Q_predict-Mu*(fun_gra/norm(fun_gra));
Q_predict=Q_estimated;

```

end


```

optimized_Q=[Q_estimated(2,1);Q_estimated(3,1);Q_estimated
(4,1);...
Q_estimated(1,1)];
end

```

Gradient Descent Algorithm Convergence

To test the convergence of the Gradient Descent algorithm, the same inputs used to test the Gauss-Newton were utilized. As it was mentioned in the previous section, the measurement noise of the utilized data was $6.4792e^{-6}$. The value of adaptation step μ was set to 0.07. The results of the simulation can be seen in Figure 5.31. This algorithm converges after 12 iterations but a zigzag effect appears in the results.

Figure 5.31: Gradient descent Algorithm Convergence

The convergence of algorithm was tested for different measurements noise vales. The results of these tests are shown in Figure 5.32.

Figure 5.32: Iteration Convergence Number for Different Noises

5.8 Kalman Filter Implementation

The Kalman filter was implemented using both the Gauss-Newton and the Gradient Descent algorithms. The initial values for H , P_0 and $\hat{x}_0$ were set as:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.23)$$

$$P_0^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times 100 \quad (5.24)$$

$$\hat{x}_0^- = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.25)$$

The initial value for the observation vector was chosen as:

$$Q_{observation} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix} \quad (5.26)$$

The process and measurement noise covariance matrices were written as:

$$Q = \begin{bmatrix} 0.0001 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix} \quad (5.27)$$

$$R = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix} \quad (5.28)$$

The gyroscope variances, in stationary situation, for the X, Y and Z axes are shown in Table 5.5.

Gyroscope Variance	
X Axis	0.001
Y Axis	0.0007
Z Axis	0.0007

Table 5.5: Gyroscope Variances in Stationary Situation

The Matlab code for implementing the Kalman filter is shown in Listing 5.4.

Listing 5.4: Kalman Filtering Code

```
% ***** Quaternion-based Kalman Filter*****

dt=1/213;

datalength=x1;
```

```

C=0.5*dt;

% Compensate for the gyro bias
gyro1(:,1)=gyro1(:,1)-mean(gyro1(:,1));
gyro1(:,2)=gyro1(:,2)-mean(gyro1(:,2));
gyro1(:,3)=gyro1(:,3)-mean(gyro1(:,3));


Q_update=zeros(4,datalength);
G_rate=gyro1.';
accel=accel1.';
compass=compass1.';


% Initial value for quaternion
Q_update(:,1)=[0.5 0.5 0.5 0.5];


% Q matrix
Q_matrix=zeros(4,4);
Q_matrix(1,1)=0.0001;
Q_matrix(2,2)=0.0001;
Q_matrix(3,3)=0.0001;
Q_matrix(4,4)=0.0001;


% H matrix:
H=eye(4,4);


% R matrix:
R=[0.001 0 0 0 ; 0 0.001 0 0 ; 0 0 0.001 0; 0 0 0 0.001];

```

```

% error covariance
P1=eye(4,4)*100;

Q_dot=zeros(4,datalength);
Q_norm=zeros(1,datalength);
Mu=zeros(1,datalength);
alpha=40;

% Observation quaternion
Q_predict=zeros(4,datalength);
Q_predict(:,1)=[1,0,0,0];
Q_observation=zeros(4,datalength);
Q_observation(:,1)=[0.5,0.5,0.5,0.5];

%test_vector=zeros(datalength);
accel1=accel1.';
compass1=compass1.';
for i=2:datalength

    % Quaternion variation using gyro data:(1/2).Q_vector.
    G_rate
    Q_dot(:,i)=(1/2)*Q_product(Q_update(:,i-1),[0 G_rate(1,i)
        G_rate(2,i) ...
        G_rate(3,i)]');

    % Mu=alpha.norm(Q_dot).delta_t;
    Q_norm(1,i)=norm(Q_dot(:,i));

```

```

Mu(1,i)=alpha*Q_norm(1,i)*dt;

%Observation
Q_observation(:,i)=MyGradientDescent(accel1(:,i),compass1
(:,i),...
    Q_observation(:,i-1),Mu(1,i));
Q_observation(:,i)=Q_observation(:,i)/norm(Q_observation
(:,i));

% F matrix:  $Q_{predict}(t)=Q(t-1)+Q_{dot}(t).dt=F.Q_{update}(t-1)$ 
F=[1 -C*G_rate(1,i) -C*G_rate(2,i) -C*G_rate(3,i);...
    C*G_rate(1,i) 1 C*G_rate(3,i) -C*G_rate(2,i);...
    C*G_rate(2,i) -C*G_rate(3,i) 1 C*G_rate(1,i);...
    C*G_rate(3,i) C*G_rate(2,i) C*G_rate(1,i) 1];

Q_predict(:,i)=F*Q_update(:,i-1);

% kalman gain
K=P1*H'*(H*P1*H'+R)^(-1);
%sprintf('K is %f',K)

% Update the system
Q_update(:,i)=Q_predict(:,i)+K*(Q_observation(:,i)-H*
    Q_predict(:,i));
Q_update(:,i)=Q_update(:,i)/norm(Q_update(:,i));

```

```

    % Error covariance is updated

    P=(eye(4,4)-K*H)*P1;

    P1=F*P*F'+Q_matrix;

end

```

Error Matrix Covariance Trace

To test the performance of the Kalman filter, the error matrix covariance (P) was traced. This matrix is used to calculate the Kalman gain. As it was discussed earlier, to calculate the Kalman gain, the sum of squared errors should be minimized. Figure 5.33 depicts the evaluation of the sum of squared error as adaption progressed. As it can be seen, the sum of squared errors was minimized and reached a steady state.

Figure 5.33: Trace of Error Covariance Matrix

Smoothing the Kalman Filter Output

Orientation data calculated by Kalman filter needed further smoothing. To smooth the quaternion data, a moving average filter was implemented. Matlab code for this moving average filter is shown in Listing 5.5.

Listing 5.5: Moving Average Filtering Code

```
function [ave] = movingaverage(mysignal, n, length)
temp=0;
ave=zeros(4, length);
for i=1:length-(n-1)
    for j=0:n-1
        temp=mysignal(:,j+i)+temp;
    end
    ave(:,i)=temp/n;
    temp=0;
end
B=ave(:, length-(n-1));
for i=length-(n-2):length
    ave(:,i)=B;
end
```

Figure 5.34 shows how this filter smooths the quaternion orientation data. At the top, orientation data computed by the Kalman filter is shown and the bottom part of the figure displays orientation data after passing through the moving average filter.

Kalman filter calculates the attitude in quaternion form. To convert the quaternion results to the Euler angles the approach which was discussed in [40] was used.

Figure 5.34: Smoothing the Filter Output

Kalman Filter Results

The Kalman filter generates the quaternion results for orientation. Figures 5.35 to 5.40 show the Kalman filter results for random hand rotations in six trials.

Figure 5.35: Kalman Filter Output- Trial 1

Figure 5.36: Kalman Filter Output- Trial 2

Figure 5.37: Kalman Filter Output- Trial 3

Figure 5.38: Kalman Filter Output- Trial 4

Figure 5.39: Kalman Filter Output- Trial 5

Figure 5.40: Kalman Filter Output- Trial 6

Figures 5.41 to 5.41 depict the Kalman filter quaternion output using the Gradient Descent and the Gauss-Newton methods.

Figure 5.41: Gauss-Newton and Gradient Descent Comparison (q1)

Figure 5.42: Gauss-Newton and Gradient Descent Comparison (q2)

Figure 5.43: Gauss-Newton and Gradient Descent Comparison (q3)

Figure 5.44: Gauss-Newton and Gradient Descent Comparison (q4)

CHAPTER 6

EXPERIMENTAL RESULTS

6.1 Introduction

This chapter describes the design of the experiment used to evaluate the 3D human hand rotation monitoring approach using inertial and magnetic MEMS units. In order to evaluate the performance of the implemented algorithms, three experiments were designed. Statistical data analyses for the experiment results, including t-test were performed. The Microsoft Excel and IBM SPSS 21 programs were used to perform data analysis.

The experiments were approved by the Internal Review Board (IRB) of Florida International University IRB. The IRB approval number is IRB-15-0008. The corresponding approved memo is shown in Appendix A.

6.2 Pool of Experimental Subjects

The requirements for participating in the experiment were the age of the subject, which required to be between 18 to 65 years old, and ability to move the hand. Volunteers from both genders were recruited to participate in the experiment. 25 individuals volunteered for the test, including 13 males and 12 females. The average age of volunteers was 25.48 years old, the standard deviation was 4.04. The volunteers were primarily FIU graduate and undergraduate students. No prior experience with wearable devices was required for participating in the experiment. The volunteers participated in Experiment 2 and Experiment 3.

6.3 Experiment Procedure

The following steps were pursued to performed the experiments:

1. Volunteers were asked to read and sign the FIU IRB user's consent form for these experiments. The consent form is shown in appendix B.
2. Each subject was assigned an identification number.
3. The purpose of experiment was explained to each subject.
4. Each subject was trained with the device and the experiments. The subject was asked to wear the glove and perform some hand movements which acquainted the subject with the device, (Figure 6.1).
5. The subject performed the experiments and the related data were recorded.
6. The subject was asked to fill out the exit questionnaire.
7. The experimental session was concluded.

Figure 6.1: Volunteer Training with the Instrumented Glove

6.4 Experiment 1: Performance Test and Accuracy Comparison

The purpose of this experiment was to measure the accuracy of the system for both proposed approaches: Quaternion-based Kalman filter with Gradient Descent optimization method and Quaternion-based Kalman filter with Gauss-Newton optimization method. Rotations about X , Y and Z axes were performed 30 times for each of the axes. Figure 6.2 shows the setup which was designed to carry out this experiment.

Figure 6.2: Experiment 1

The recorded data were processed with both algorithms (Quaternion-based Kalman filter with Gradient Descent optimization method and Quaternion-based Kalman filter with Gauss-Newton optimization method).

6.4.1 Quaternion-based Kalman filter with Gradient Descent optimization method-Results

Each of roll, pitch and yaw rotations were repeated 30 times. The results (i.e., the angle value at which the method converged) for 20 degree rotations using Quaternion-based Kalman filter with Gauss-Newton optimization method are shown in Tables 6.1, 6.2 and 6.3.

20 Degree Roll Rotation Using Gradient Descent Method					
Number	Rotation Result	Number	Rotation Result	Number	Rotation Result
1	19.90	11	20.90	21	20.40
2	19.90	12	20.40	22	20.30
3	20.00	13	20.30	23	19.70
4	20.00	14	20.10	24	20.00
5	19.80	15	20.10	25	20.10
6	20.00	16	20.20	26	19.40
7	19.30	17	18.90	27	21.70
8	20.00	18	19.50	28	20.10
9	20.00	19	20.30	29	20.30
10	20.00	20	20.40	30	20.60

Table 6.1: 20 degree rotation about X axis

6.4.2 Quantitative Analysis for Approach Using Gradient Descent

The descriptive statistics for the results of rotations using the Quaternion Kalman filter with the gradient descent method are shown in Tables 6.4, 6.5 and 6.6. The mean for roll is 20.12, the mean for pitch is 20.04 and the mean for yaw is 19.91.

It should be noted that, inspite of the clear markings provided for the execution of the correct amount of rotational movement, the subjects may have introduced a

20 Degree Pitch Rotation Using Gradient Descent Method					
Number	Rotation Result	Number	Rotation Result	Number	Rotation Result
1	19.83	11	21.17	21	19.88
2	19.59	12	20.12	22	21.23
3	19.92	13	19.76	23	19.76
4	19.88	14	19.62	24	19.34
5	19.76	15	19.49	25	19.61
6	19.91	16	21.21	26	20.43
7	21.06	17	20.13	27	19.69
8	19.65	18	20.30	28	19.66
9	19.67	19	21.49	29	19.70
10	20.92	20	19.46	30	19.13

Table 6.2: 20 degree rotation about Y axis

20 Degree Yaw Rotation Using Gradient Descent Method					
Number	Rotation Result	Number	Rotation Result	Number	Rotation Result
1	20.40	11	19.35	21	20.64
2	19.53	12	20.42	22	19.64
3	19.78	13	20.55	23	19.49
4	20.52	14	20.11	24	20.15
5	19.67	15	19.65	25	20.69
6	20.52	16	20.39	26	19.36
7	19.92	17	19.45	27	20.10
8	19.85	18	19.51	28	20.06
9	19.45	19	19.67	29	21.02
10	19.47	20	19.50	30	19.50

Table 6.3: 20 degree rotation about Z axis

level of inaccuracy in the rotation amount, which may contribute to the dispersion of results values around the nominal rotation angle.

Figures 6.3, 6.4 and 6.5 show the histograms of the processed data with a fitted Gaussian curve superimposed for the roll, pitch and yaw rotations, respectively.

20 Degree Roll Rotation Using Gradient Descent Method	
Number of samples	30
Mean	20.1200
Median	20.1
Maximum	21.70
Minimum	18.90
Range	2.80
Std.Error	0.094
Std. Deviation	0.5188
Variance	0.269
Skewness	0.525
Skewness Std.Error	0.427
Kurtosis	2.618
Kurtosis Std.Error	0.833

Table 6.4: Descriptive Statistics for 20 Degree Rotation (Roll)

20 Degree Pitch Rotation Using Gradient Descent Method	
Number of samples	30
Mean	20.0475
Median	19.795
Maximum	21.49
Minimum	19.13
Range	2.36
Std.Error	0.1163
Std. Deviation	0.6369
Variance	0.406
Skewness	1.052
Skewness Std.Error	0.427
Kurtosis	-0.034
Kurtosis Std.Error	0.833

Table 6.5: Descriptive Statistics for 20 Degree Rotation (Pitch)

20 Degree Yaw Rotation Using Gradient Descent Method	
Number of samples	30
Mean	19.9123
Median	19.815
Maximum	20.69
Minimum	19.35
Range	1.34
Std.Error	0.07962
Std. Deviation	0.4361
Variance	0.190
Skewness	0.408
Skewness Std.Error	0.427
Kurtosis	-1.294
Kurtosis Std.Error	0.833

Table 6.6: Descriptive Statistics for 20 Degree Rotation (Yaw)

Figure 6.3: Histogram for 20 Degree Roll Rotation Using Gradient Descent Method

Figure 6.4: Histogram for 20 Degree Pitch Rotation Using Gradient Descent Method

Figure 6.5: Histogram for 20 Degree Yaw Rotation Using Gradient Descent Method

One-Sample t-Test

A One-Sample t-test was calculated for each of the rotational motions. The results are stated in Table 6.7. The results of the t-test show that for all three cases the angular estimates obtained are not significantly different from the nominal angle value. The p values(Sig.) for the roll, pitch and yaw rotations are 0.215, 0.697 and 0.280, all of which are bigger than cut off value of 0.05. Therefore it can be concluded that the results of this algorithm are reliable. Figures 6.6, 6.7, and 6.8 show the QQ plots for the Roll, Pitch and Yaw data sets.

Figure 6.6: QQ plot for Roll Data Using Gradient Descent Method

One-Sample T test for Gradient Descent Method						
Rotation	t	df	Sig.	Mean Difference	95% Interval Lower Bound	95% Interval Upper Bound
Roll	1.262	29	0.215	0.12	-0.0738	0.3138
Pitch	0.393	29	0.697	0.04567	-0.1922	0.2835
Yaw	-1.101	29	0.280	-0.08767	-0.2505	0.0752

Table 6.7: T Test Results

Figure 6.7: QQ plot for Pitch Data Using Gradient Descent Method

Figure 6.8: QQ plot for Yaw Data Using Gradient Descent Method

6.4.3 Quaternion-based Kalman filter with Gauss-Newton optimization method-Results

The results of rotations using Quaternion-based Kalman filter with the Gauss-Newton optimization method are shown in tables 6.8, 6.9 and 6.10.

20 Degree Roll Rotation Using Gauss-Newton Method					
Number	Rotation Result	Number	Rotation Result	Number	Rotation Result
1	20.30	11	20.70	21	20.40
2	20.60	12	20.40	22	20.10
3	20.00	13	20.40	23	20.00
4	19.90	14	19.90	24	19.90
5	20.00	15	20.10	25	20.10
6	20.10	16	20.30	26	19.40
7	19.90	17	19.30	27	21.80
8	20.00	18	19.30	28	20.10
9	20.10	19	20.40	29	19.60
10	20.00	20	20.40	30	20.30

Table 6.8: 20 degree rotation about X axis

20 Degree Pitch Rotation Using Gauss-Newton Method					
Number	Rotation Result	Number	Rotation Result	Number	Rotation Result
1	19.40	11	20.76	21	19.64
2	19.33	12	19.57	22	20.89
3	19.81	13	19.36	23	19.64
4	19.94	14	19.47	24	19.36
5	19.94	15	19.60	25	19.37
6	19.82	16	21.33	26	20.42
7	20.79	17	19.79	27	19.62
8	19.24	18	20.22	28	19.57
9	19.57	19	20.99	29	19.56
10	20.63	20	20.02	30	19.31

Table 6.9: 20 degree rotation about Y axis

20 Degree Yaw Rotation Using Gauss-Newton Method					
Number	Rotation Result	Number	Rotation Result	Number	Rotation Result
1	20.30	11	19.52	21	20.73
2	19.44	12	20.33	22	19.59
3	19.52	13	20.52	23	19.28
4	20.38	14	20.02	24	19.77
5	19.68	15	19.52	25	20.48
6	19.19	16	20.29	26	19.24
7	19.73	17	19.24	27	20.02
8	19.80	18	19.36	28	19.89
9	19.41	19	19.66	29	19.90
10	19.67	20	19.47	30	21.22

Table 6.10: 20 degree rotation about Z axis

6.4.4 Quantitative Analysis for the Approach Using Gauss-Newton Optimization

The descriptive statistics for results of rotations using Quaternion Kalman filter with the Gauss-Newton method are shown in Tables 6.11, 6.12 and 6.13. Figures 6.9, 6.10 and 6.11 show the histograms of the processed data for roll, pitch and yaw rotations respectively.

t-Test

A One-Sample t-test was performed for the data calculated by the Gauss-Newton algorithm. The results of t-test are shown in Table 6.14. The calculated p values(Sig.) for the roll, pitch and yaw rotations are 0.147, 0.349 and 0.088, which all of them are bigger than threshold value 0.05. The results of the t-test prove that for all three cases the results are not significantly different from the population mean value and it can be concluded that the results of algorithm are reliable. Figures 6.12, 6.13, and 6.14 show the QQ plots for the Roll, Pitch and Yaw data.

20 Degree Roll Rotation Using Gauss-Newton Method	
Number of samples	30
Mean	20.1267
Median	20.1
Maximum	21.80
Minimum	19.30
Range	2.50
Std.Error	0.084
Std. Deviation	0.4653
Variance	0.217
Skewness	1.240
Skewness Std.Error	0.427
Kurtosis	5.107
Kurtosis Std.Error	0.833

Table 6.11: Descriptive Statistics for 20 Degree Rotation

20 Degree Pitch Rotation Using Gauss-Newton Method	
Number of samples	30
Mean	19.8987
Median	19.64
Maximum	21.33
Minimum	19.24
Range	2.09
Std.Error	0.1063
Std. Deviation	0.5827
Variance	0.340
Skewness	1.037
Skewness Std.Error	0.427
Kurtosis	-0.044
Kurtosis Std.Error	0.833

Table 6.12: Descriptive Statistics for 20 Degree Rotation

20 Degree Yaw Rotation Using Gauss-Newton Method	
Number of samples	30
Mean	19.8390
Median	19.70
Maximum	21.22
Minimum	19.19
Range	2.03
Std.Error	0.0911
Std. Deviation	0.4992
Variance	0.249
Skewness	0.917
Skewness Std.Error	0.427
Kurtosis	0.475
Kurtosis Std.Error	0.833

Table 6.13: Descriptive Statistics for 20 Degree Rotation

Figure 6.9: Histogram for 20 Degree Roll Rotation Using Gauss-Newton Method

Figure 6.10: Histogram for 20 Degree Pitch Rotation Using Gauss-Newton Method

Figure 6.11: Histogram for 20 Degree Yaw Rotation Using Gauss-Newton Method

Figure 6.12: QQ plot for Roll Data Using Gauss-Newton Method

Figure 6.13: QQ plot for Pitch Data Using Gauss-Newton Method

Figure 6.14: QQ plot for Yaw Data Using Gauss-Newton Method

One-Sample T test for Gauss-Newton Method						
Rotation	t	df	Sig.	Mean Difference	95% Interval Lower Bound	95% Interval Upper Bound
Roll	1.491	29	0.147	0.1266	-0.0471	0.3004
Pitch	-0.952	29	0.349	-0.1013	-0.3189	0.1163
Yaw	-1.766	29	0.088	-0.1610	-0.3474	0.0254

Table 6.14: T Test Results

6.4.5 Statistic Comparison for Gauss-Newton and Gradient Descent

After analyzing the reliability of each algorithm for all three rotational motions, there is the need to determine whether one algorithm delivers significantly different results or whether both algorithms perform similarly. In order to compare the performance of the two implemented algorithms, an independent-Samples t-test was performed. For this test, all processed rotational data from the Gradient Descent method was grouped in one category (Group 1) and all processed rotational data Gauss-Newton algorithm was put in a second group (Group 2). Figure 6.15 shows the box plot for these two groups of data. The sample size in each group is 90. The calculated mean of data for Group 1 is 20.0260 with standard deviation 0.53793. Group 2 has mean equal to 19.9548 with standard deviation 0.52722.

Figure 6.15: Box Plot for Comparison between Optimization Methods

Tests of Normality

To check the normality of the experimental data, both QQ plots and Shapiro-Wilk tests were used. Looking at the Figures 6.16 and 6.17 the approximate normality of the data can be visualized. Table 6.4.5 shows the Shapiro-Wilk normality test results. The p value is 0.003 for Group 1 and 0.001 for Group 2 which affirms that data is normally distributed.

Shapiro-Wilk Normality Test			
Group	Statistic	df	Sig.
1	0.935	90	0.003
2	0.943	90	0.001

Table 6.15: Test of Normality

Figure 6.16: QQ plot for Experimental Data Using Gauss-Newton Method

Figure 6.17: QQ plot for Experimental Data Using Gauss-Newton Method

Independent t-Test

Table 6.16 shows the results of Leven's test for equality of variances. The independent t-test assumes that the variances are the same in both samples to the extent that they are not significantly different from each other. The significant value from Leven's test is 0.679 which shows the variances in both groups are not significantly different from each other. If this number was smaller than 0.05 it would indicate that these variances are significantly different.

The t-test results are shown in Table 6.17. Because, as it can be seen in Table 6.17, the significant value of the t-test is bigger than 0.05. This implies that the means of the two groups are not significantly different.

Independent-Sample T test		
Levene's Test for Equality of Variances		
	F	Sig.
Equal Variances Assumed	0.001	0.974
Equal Variances not Assumed		

Table 6.16: Results from Levene's Test

Independent-Sample T test						
T test for Equality of Mean						
t	df	Sig.	Mean Difference	Std. Error Difference	95% Interval Lower Bound	95% Interval Upper Bound
0.897	178	0.371	0.07122	0.7940	-0.08546	0.22790
0.897	177.928	0.371	0.07122	0.07940	-0.08546	0.22790

Table 6.17: t-Test Results

6.5 Experiment 2: Movement Classification

The purpose of this experiment was to classify the human thumb motion using the rotation signal calculated by the proposed algorithms. The potential application for this experiment can be in utilizing thumb movement for controlling devices. It can be used by people with disabilities to simplify some daily tasks. Each of the subjects was asked to perform abduction/ adduction and flexion/ extension motions. The calculated rotation data about the Z-axis was used as classification process input. The Matlab code for thumb movement classification can be seen in Listing (6.1), part A.

Listing 6.1: Motion Classification and Motion Repetition Code

```
% Part A: Motion Classification
Bendingdata=Rotation_signal(50:x1-10,2);
min_value=min(Bendingdata);
max_value=max(Bendingdata);
```

```

sumy=abs(max_value-min_value);
sprintf('sumy %f',sumy)
Threshold1=56;
if (sumy<Threshold1)
    sprintf('Abduction/Adduction')
else
    sprintf('Flexion/Extension')
end
%Part B: Repetition counting
intervals=(sumy/10);
thumb_circle=min_value:intervals:max_value;
thumb_position=zeros(length(Bendingdata),1);
x=length(Bendingdata);
for i=1:x-1
    if (Bendingdata<thumb_circle(2))
        thumb_position(i)=1;
    end
    if ( thumb_circle(2) <= Bendingdata(i) && ...
        Bendingdata(i) < thumb_circle(3))
        thumb_position(i)=2;
    end
    if (thumb_circle(3)<= Bendingdata(i) && ...
        Bendingdata(i) < thumb_circle(4))
        thumb_position(i)=3;
    end
    if (thumb_circle(4)<= Bendingdata(i) && ...
        Bendingdata(i) < thumb_circle(5))
        thumb_position(i)=4;

```

```

end
if      thumb_circle(5) <= Bendingdata(i) && ...
        Bendingdata(i) < thumb_circle(6)
    thumb_position(i)=5;
end
if      thumb_circle(6) <= Bendingdata(i) && ...
        Bendingdata(i) < thumb_circle(7)
    thumb_position(i)=6;
end
if      thumb_circle(7) <= Bendingdata(i) && ...
        Bendingdata(i) < thumb_circle(8)
    thumb_position(i)=7;
end
if      thumb_circle(8) <= Bendingdata(i) && ...
        Bendingdata(i) < thumb_circle(9)
    thumb_position(i)=8;
end
if      thumb_circle(9) <= Bendingdata(i) && ...
        Bendingdata(i) < thumb_circle(10)
    thumb_position(i)=9;
end
if thumb_circle(10) <= Bendingdata(i) && ...
        Bendingdata(i) <= thumb_circle(11)
    thumb_position(i)=10;
end

end

figure;plot(thumb_position)

```

```

pos=0;neg=0;
tip_vec=0;
ori_vec=zeros(x-1,1);
collection=0;

for i=2:x-1
    ori_vec(i)=thumb_position(i)-thumb_position(i-1);
    if ori_vec(i)==1
        collection=[collection,ori_vec(i)];
        pos=pos+1;
        tip_vec=[tip_vec,i];
    end
    if ori_vec(i)==-1
        collection=[collection,ori_vec(i)];
        neg=neg+1;
        tip_vec=[tip_vec,-i];
    end
end

end

co_lenght=length(collection);
collection=collection(2:co_lenght);co_lenght=co_lenght-1;
%
i=1;final=0;temp=0;
while (i < co_lenght-1)
    while(collection(i)==collection(i+1))
        temp=temp+collection(i);i=i+1;
    end
    if i== (co_lenght-1)
        break
    end
end

```

```

        end
    end
    if(temp~=0)
        final=[final,temp];
    end
    temp=0;
    i=i+1;
end
T1=0;T2=0;
final=final(1:end);
final=final+1*sign(final);
for i=1:length(final)
    if final(i)<-3
        T1=T1+1;
    end
    if final(i)> 3
        T2=T2+1;
    end
end

sprintf('Number of bending is %d',T1)
sprintf('Number of bending-back is %d',T2)

```

6.5.1 Data Analysis for Experiment 2

25 subjects participated in this experiment. Each subject performed the abduction/ adduction and flexion/ extension motion. The aggregate results of the motion classification process are shown in Table 6.18 and Table 6.19 includes individual results

for each subject. As it can be seen from the table, in 49 cases the program classified the motion successfully and there was 1 case, (ID number 15) where the program failed to classify the abduction/ adduction motion correctly. Therefore the proposed system achieved correct classifications in 98 percent of the tests.

	Abduction/ Adduction	Flexion/ Extension
Abduction/ Adduction	24	1
Flexion/ Extension	0	25

Table 6.18: Confusion Matrix for Experiment 2

6.6 Experiment 3: Thumb Motion Repetition

In this experiment each subject was asked to perform each of the abduction, adduction, flexion and extension motions 3 times. The algorithm for identifying the movements and counting them is shown in Listing 6.1, part B.

6.6.1 Data Analysis for Experiment 3

The results of the motion repetition counting process are shown in Table 6.20. The proposed algorithm could count the number of repetitions with 97 percent accuracy. As it can be seen, 100 cases of motion were recorded which 97 cases were correctly identified and counted by the algorithm. There are 3 cases in which the algorithm delivered the wrong answer.

6.7 Questionnaires

Each subject filled out a questionnaire after participating in the experiment. The purpose of the questionnaire was to obtain feedback from subjects about their ex-

Motion Classification Results		
ID	Abduction/ Adduction Motion	Flexion/ Extension Motion
1	True	True
2	True	True
3	True	True
4	True	True
5	True	True
6	True	True
7	True	True
8	True	True
9	True	True
10	True	True
11	True	True
12	True	True
13	True	True
14	True	True
15	False	True
16	True	True
17	True	True
18	True	True
19	True	True
20	True	True
21	True	True
22	True	True
23	True	True
24	True	True
25	True	True

Table 6.19: Motion Classification Results

Motion counting Results				
ID	Abduction	Adduction	Flexion	Extension
1	3	3	3	3
2	3	3	3	3
3	3	3	3	3
4	3	3	3	3
5	3	3	3	2
6	3	3	3	3
7	3	3	3	3
8	3	3	3	3
9	3	3	3	3
10	3	3	3	3
11	3	3	4	3
12	3	3	3	3
13	3	3	3	3
14	3	3	3	3
15	3	3	3	3
16	3	3	3	3
17	3	3	3	3
18	3	3	3	3
19	3	3	3	3
20	3	3	3	3
21	3	3	4	3
22	3	3	3	3
23	3	3	3	3
24	3	3	3	3
25	3	3	3	3

Table 6.20: Motion Repetition Counting Results

perience with the instrumented glove. The questionnaires contained 5 questions, as follows:

1. In a scale from 1 to 10, please rank how convenient you found using our smart glove. The higher you rank (e.g., 10), the more convenient you found the device.
2. In a scale from 1 to 10, please rank how interesting you found our smart glove? The higher you rank (e.g., 10), the more interesting you found the device.
3. In a scale from 1 to 10, please rank how likely you would be to use this device for every day use if you had access to it. The higher you rank, the more you would expect to use it, if it was available to you.
4. In a scale from 1 to 10, rank how likely you would be to use this device for medical purposes, if you had access and need to. The higher you rank, the more you would expect to use if it was available to you.
5. In a scale from 1 to 10, please rank what is the possibility that you would prefer to use device, compared to staying in front of a camera for vision-based motion detection?

Figures 6.18 to 6.22 depict the histograms derived from the questionnaire answers to each question.

6.8 Hand Motion Visualization

To visualize the results of the proposed algorithms, a graphical interface was designed using "Processing" software. This program receives the palm and the thumb rotational data which was produced by MATLAB post-processing and animates the motion of the hand, (Figure 6.23). Listing 6.2 shows the code for this program. The

Figure 6.18: Histogram (Answers to Question 1)

Figure 6.19: Histogram (Answers to Question 2)

Figure 6.20: Histogram (Answers to Question 3)

Figure 6.21: Histogram (Answers to Question 4)

Figure 6.22: Histogram (Answers to Question 5)

output of this program is shown in Figure 6.24. This figure contains a screen captures acquired during the animation of the hand driven by orientation estimations obtained by the algorithms implemented in this research.

Figure 6.23: Hand Motion Visualization

Listing 6.2: Hand Motion Visualization Code

```
//Rotation data from matlab
float [] Xrot;
float [] Yrot;
float [] Zrot;

int initTime=0;
int t=initTime;
int stepSize=5;
int runTill;
int i;
```

```

float R0=40;
float R1=37.5,R2=35; //finger and thumb radius
float angX=0,angY=0,angXinst=0,angYinst=0;
float distZ=-400,distZinst=0;

void setup(){
    // Reading palm data
    String[] readLog1=loadStrings("xrotation.txt");
    Xrot= new float[readLog1.length];

    String[] readLog2=loadStrings("yrotation.txt");
    Yrot= new float[readLog2.length];

    String[] readLog3=loadStrings("zrotation.txt");
    Zrot= new float[readLog3.length];

    for(i=0; i<readLog1.length-1; i++)
    {
        Xrot[i] = float(readLog1[i]);
        Yrot[i] = float(readLog2[i]);
        Zrot[i] = float(readLog3[i]);
    }
    runTill=readLog1.length;

    //noLoop();
    size(1280,1024,P3D);

```

```

colorMode(RGB);

}

void draw() {

//palm
background(145,226,356);
float Xradp=-Xrot[t];//-PI/2;//Sould be minus
float Yradp=-Yrot[t];//sould be minus
float Zradp=Zrot[t];

//thumb
/float Xradt=0;
/float Yradt=0;
/float Zradt=0;

//moving the origin
  translate(640,500,distZ+distZinst);
  noStroke();lights();
  // Paint palm

  rotateZ(PI/2); //
  rotateY(-PI/2); //
  rotateZ(-3*PI/2);

//Draw Palm
  rotateZ(Zradp); //Rotate Palm about Z

```



```

rotateY(Yradp); //Rotate Palm about Y
rotateX(Xradp); //Rotate Palm about X

box(300,400,90);

beginShape(QUAD);
vertex(-70,200,125);
vertex(-150,0,125);
vertex(-230,0,125);
vertex(-150,200,125);

vertex(-150,200,125);
vertex(-230,0,125);
vertex(-230,0,45);
vertex(-150,200,45);

vertex(-230,0,45);
vertex(-150,200,45);
vertex(-150,200,-45);
vertex(-150,0,-45);

vertex(-70,200,45);
vertex(-150,200,45);
vertex(-150,200,125);
vertex(-70,200,125);

vertex(-70,200,125);
vertex(-150,0,125);
vertex(-150,0,45);

```

```

vertex(75,200,85);

vertex(-150,0,45);
vertex(75,200,85);
vertex(150,100,45);
vertex(0,0,45);

vertex(150,200,45);
vertex(75,200,85);
vertex(-70,200,125);
vertex(-70,200,45);

vertex(-150,0,45);
vertex(-150,0,125);
vertex(-230,0,125);
vertex(-230,0,45);
endShape();

beginShape(TRIANGLES);
vertex(-230,0,45);
vertex(-150,0,-45);
vertex(-150,-80,45);

vertex(-230,0,45);
vertex(-150,-80,45);
vertex(-150,0,45);

vertex(75,200,85);

```

```

vertex(150,100,45);
vertex(150,200,45);
endShape();
//Draw Thumb
pushMatrix();translate(-150-R0,-10,85);

//*****
// when data is from one sensor the numbers should be zero;
pushMatrix();
rotateX(0);
//rotateX(-Xradp);
pushMatrix();
rotateY(0);
//rotateY(-Yradp);
pushMatrix();
rotateZ(-0);
//rotateZ(-Zradp);
//*****
pushMatrix();rotateZ(0);
//Rotate Z 1st Joint -0.688366 0.787209 1.258135
pushMatrix();rotateY(0);
//Rotate Y 1st Joint -0.565934 -0.281885 0.648590
pushMatrix();rotateX(0); //Rotate X 1st Joint

sphere(R0);drawCylinder(R0,R1,-100);
pushMatrix();translate(0,-100,0);
pushMatrix();rotateZ(0); //Rotate Z 2nd Joint
sphere(R1);drawCylinder(R1,R1,-100);

```

```

pushMatrix();translate(0,-100,0);
sphere(R1);
popMatrix();popMatrix();popMatrix();popMatrix();
popMatrix();popMatrix();popMatrix();
popMatrix();popMatrix();popMatrix();

//Draw Index
translate(-150+R1,-200,0);
pushMatrix();rotateZ(0); //Rotate Z 1st Joint
pushMatrix();rotateX(-PI/18); //Rotate X 1st Joint
sphere(R1);drawCylinder(R1,R1,-100);
pushMatrix();translate(0,-100,0);
pushMatrix();rotateX(-PI/18); //Rotate 2nd Joint
sphere(R1);drawCylinder(R1,R2,-100);
pushMatrix();translate(0,-100,0);
pushMatrix();rotateX(-PI/18); //Rotate 3rd Joint
sphere(R2);drawCylinder(R2,R2,-80);
pushMatrix();translate(0,-80,0);
sphere(R2);
popMatrix();popMatrix();popMatrix();popMatrix();
popMatrix();popMatrix();popMatrix();

//Draw Middle
translate(2*R1,0,0);
pushMatrix();rotateZ(0); //Rotate Z 1st Joint
pushMatrix();rotateX(-PI/18); //Rotate X 1st Joint
sphere(R1);drawCylinder(R1,R1,-150);
pushMatrix();translate(0,-150,0);

```

```

pushMatrix();rotateX(-PI/18); //Rotate X 2nd Joint
sphere(R1);drawCylinder(R1,R2,-100);
pushMatrix();translate(0,-100,0);
pushMatrix();rotateX(-PI/18); //Rotate X 3rd Joint
sphere(R2);drawCylinder(R2,R2,-80);
pushMatrix();translate(0,-80,0);
sphere(R2);
popMatrix();popMatrix();popMatrix();popMatrix();
popMatrix();popMatrix();popMatrix();

//Draw Ring
translate(2*R1,0,0);
pushMatrix();rotateZ(0); //Rotate Z 1st Joint
pushMatrix();rotateX(-PI/18); //Rotate X 1st Joint
sphere(R1);drawCylinder(R1,R1,-120);
pushMatrix();translate(0,-120,0);
pushMatrix();rotateX(-PI/18); //Rotate X 2nd Joint
sphere(R1);drawCylinder(R1,R2,-100);
pushMatrix();translate(0,-100,0);
pushMatrix();rotateX(-PI/18); //Rotate X 3rd Joint
sphere(R2);drawCylinder(R2,R2,-80);
pushMatrix();translate(0,-80,0);
sphere(R2);
popMatrix();popMatrix();popMatrix();popMatrix();
popMatrix();popMatrix();popMatrix();

//Draw Pinky
translate(2*R1,0,0);

```

```

pushMatrix();rotateZ(0); //Rotate Z 1st Joint
pushMatrix();rotateX(-PI/18); //Rotate X 1st Joint
sphere(R1);drawCylinder(R1,R1,-80);
pushMatrix();translate(0,-80,0);
pushMatrix();rotateX(-PI/18); //Rotate X 2nd Joint
sphere(R1);drawCylinder(R1,R2,-80);
pushMatrix();translate(0,-80,0);
pushMatrix();rotateX(-PI/18); //Rotate X 3rd Joint
sphere(R2);drawCylinder(R2,R2,-80);
pushMatrix();translate(0,-80,0);
sphere(R2);
popMatrix();popMatrix();popMatrix();popMatrix();
popMatrix();popMatrix();popMatrix();

if(t<runTill-stepSize)
{
t=t+stepSize;
}

}

void drawCylinder(float R_end1,float R_end2,float H){
float angle=0;
beginShape(QUAD_STRIP);
for (int i=0;i<64+1;++i){
vertex(R_end1*cos(angle),0,R_end1*sin(angle));
vertex(R_end2*cos(angle),H,R_end2*sin(angle));
angle = angle+TWO_PI/64;
}
}

```

```
}  
endShape();  
}
```

Figure 6.24: Motions Snapshots

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusions

This dissertation presented a novel approach to detect the human hand motion using MEMS inertial and magnetic sensors. In order to combine the outputs delivered by these sensors, and also for compensating the stochastic errors of the sensors a sensor fusion algorithm was designed. The quaternion-based Kalman filter approach was chosen for this study.

The quaternion form of tri-axis gyroscope data was used as an input for the prediction step of the Kalman filter and the accelerometer and the magnetometer data were used as observation data for the correction step of the filter. To compute a unique true quaternion orientation from observation data, optimization algorithms were used. The orientation of the sensor frame relative to the Earth frame was extracted using the recorded field's direction in the sensor frame.

The experiments conducted and associated data analyses, revealed that the proposed algorithm yielded reliable results. The One-Sample t-test results for the Quaternion-based Kalman filter with Gauss-Newton optimization method indicate that the outcome of this algorithm is not significantly different from the population mean value for all cases of roll, pitch and yaw rotations. The same test was carried out for evaluating the results yielded by the Quaternion-based Kalman filter with Gradient Descent optimization, and the statistical analysis also showed that the results of this algorithm are not significantly different from the population mean values.

Both optimization approaches were tested for the convergence. The number of iterations for the algorithms to converge depended on the measurement noise.

Therefore different values for measurement noise were used to test the algorithms. For the same level of the measurement noise, the Gauss-Newton approach converged faster than Gradient Descent method. Also, a zigzag effect was observed in the results from the Gradient Descent method. However, the results from the independent t-test comparison between those two methods showed that these optimization approaches do not deliver significantly different results.

7.2 Future Work

It is expected that wearable devices will become popular and pervasive in the near future. The opinions from experimental volunteers after participating in this study demonstrated that the people are interested using such wearable devices. Based on this author's opinion, further advancements in sensor manufacturing will definitely lead to wearable devices which are more convenient for the user and provide less noisy output.

Since this study did not focus on any specific application, the next step of this work can be fitting the instrumented glove for a specific application. Furthermore, applying the proposed algorithm to the remaining fingers (pinky, ring, middle and index) and implementing the system in real-time are suggested to improve the device.

Considering fast emergence of the "Internet of Things" industry, the instrumented glove developed here can be improved as a cloud-based device in order to be adopted for several applications, specially remote rehabilitation solutions. Processing of sensor data sometimes may have very high computing and storage requirements. Using cloud infrastructure for sensor-based applications has various advantages, including data scalability, access to visualization tools, mobile accessibility, cost reduction, storage availability for big sensor data, higher computational

abilities, and availability of mechanism for information sharing and remote management. Large amount of real-time motion related data can be collected by the instrumented glove. Smartphones are convenient means for collecting raw data from sensors. The smartphone can check the content of the raw data, and present the raw data in suitable structure for further processing in a cloud computing infrastructure.

APPENDIX A

The following appendix provides memorandums from the Office of Research Integrity, sent to the principal investigator, Dr. Armando Barreto, and this author (Fatemeh Abyarjoo).

MEMORANDUM

To: Dr. Armando Barreto
CC: File
From: Maria Melendez-Vargas, MIBA, IRB Coordinator
Date: January 15, 2015
Protocol Title: "Human Hand Motion Detection Using MEMS Inertial/Magnetic Sensors"

The Health Sciences Institutional Review Board of Florida International University has approved your study for the use of human subjects via the **Expedited Review** process. Your study was found to be in compliance with this institution's Federal Wide Assurance (00000060).

IRB Protocol Approval #:	IRB-15-0008	IRB Approval Date:	01/09/15
TOPAZ Reference #:	102930	IRB Expiration Date:	01/09/16

As a requirement of IRB Approval you are required to:

- 1) Submit an IRB Amendment Form for all proposed additions or changes in the procedures involving human subjects. All additions and changes must be reviewed and approved by the IRB prior to implementation.
- 2) Promptly submit an IRB Event Report Form for every serious or unusual or unanticipated adverse event, problems with the rights or welfare of the human subjects, and/or deviations from the approved protocol.
- 3) Utilize copies of the date stamped consent document(s) for obtaining consent from subjects (unless waived by the IRB). Signed consent documents must be retained for at least three years after the completion of the study.
- 4) **Receive annual review and re-approval of your study prior to your IRB expiration date.** Submit the IRB Renewal Form at least 30 days in advance of the study's expiration date.
- 5) Submit an IRB Project Completion Report Form when the study is finished or discontinued.

Special Conditions: N/A

For further information, you may visit the IRB website at <http://research.fiu.edu/irb>.

APPENDIX B

The following appendix provides the FIU IRB users consent form for participating in research study.

FIU IRB Approval:	1/9/2015
FIU IRB Expiration:	1/9/2016
FIU IRB Number:	IRB-15-0008

ADULT CONSENT TO PARTICIPATE IN A RESEARCH STUDY
Title: Human Hand Motion Detection Using MEMS Inertial/ Magnetic Sensors

PURPOSE OF THE STUDY

You are being asked to participate in a research study. The purpose of this study is to test the accuracy of implemented algorithms to calculate the human hand rotation. The inertial and magnetic MEMS sensors are attached to an ordinary glove. We want to test how the algorithm responds when a normal user wears the glove and moves the hand in different directions.

NUMBER OF STUDY PARTICIPANTS

If you decide to be in this study, you will be one of 30 people in this research study.

DURATION OF THE STUDY

Your participation will require 60 minutes of your time. The first 15 minutes will be used for a brief tutorial. Then, the experiment will be divided in 2 halves of 20 minutes with a 5 minute break. The entire experiment will be held in one day and there will be no need to come back to our laboratory.

PROCEDURES

If you agree to be in the study, the experiment will be as follows:

- 1) Tutorial: This time will be used to get you acquainted with the systems. **15 minutes.**
 - a. We will describe the procedure for the subject. We will check if the subject feels comfortable wearing the glove and explain different type of hand movements.
 - b. We will ask the subject to practice the different motions before starting to record the data.
 - c. We will give you 5 additional minutes for questions about the glove, to let you practice with it or rest before we start the actual experiment.
- 2) Experiment: We will ask the user to wear the glove and perform the following movements: **(45 minutes)**
 - a. You will be asked to move your thumb in different directions.
 - b. You will be given 5 minutes to rest if you need them.
 - c. You will be asked to rotate your palm in three different dimensions.

The information that will be recorded will be anonymous. This information includes the time that it takes to accomplish a task, the amount of steps that it took and the accuracy of the movements. Your data will only be identified with a private identification number, which will not be linked to your name.

FIU IRB Approval:	1/9/2015
FIU IRB Expiration:	1/9/2016
FIU IRB Number:	IRB-15-0008

RISKS AND/OR DISCOMFORTS

There are no known risks with our experiment. The only risk that may exist is the same risk that any user may be exposed to when using a computer at their work or home. The reason for the no-risk or minimal risk involved to the subject is because the equipment involved is pervasive in everyday use. All pieces of equipment are non-invasive. You are not expected to experience any discomfort.

BENEFITS

The following benefits may be associated with your participation in this study: There is no cost or payment to you as a subject. You will not get any direct benefit from being in the study. However, your help will give us valuable data for our research.

ALTERNATIVES

There are no known alternatives available to you other than not taking part in this study. However, any significant new findings developed during the course of the research which may relate to your willingness to continue participation will be provided to you.

CONFIDENTIALITY

The records of this study will be kept private and will be protected to the fullest extent provided by law. In any sort of report we might publish, we will not include any information that will make it possible to identify a subject. Research records will be stored securely and only the researcher team will have access to the records. However, your records may be reviewed for audit purposes by authorized University or other agents who will be bound by the same provisions of confidentiality.

Your participation will be identified by a private code number, not by your name. All of the data collected in the experiment is private and will not be shared with anyone unless required by law. Your data will be compared to the data of the other subjects. We will present the research results as a group.

COMPENSATION & COSTS

You will not receive a payment for your participation. You will not be responsible for any costs to participate in this study.

RIGHT TO DECLINE OR WITHDRAW

Your participation in this study is voluntary. You are free to participate in the study or withdraw your consent at any time during the study. Your withdrawal or lack of participation will not affect any benefits to which you are otherwise entitled. The investigator reserves the right to remove you without your consent at such time that they feel it is necessary.

FIU IRB Approval:	1/9/2015
FIU IRB Expiration:	1/9/2016
FIU IRB Number:	IRB-15-0008

RESEARCHER CONTACT INFORMATION

If you have any questions about the purpose, procedures, or any other issues relating to this research study you may contact Dr, Armando Barreto at his office in the FIU Engineering Center: 10555 W. Flagler Street, Room EC-3981, Miami, FL, 33174, (305) 348-3711, barretoa@fiu.edu.

IRB CONTACT INFORMATION

If you would like to talk with someone about your rights of being a subject in this research study or about ethical issues with this research study, you may contact the FIU Office of Research Integrity by phone at 305-348-2494 or by email at ori@fiu.edu.

PARTICIPANT AGREEMENT

I have read the information in this consent form and agree to participate in this study. I have had a chance to ask any questions I have about this study, and they have been answered for me. I understand that I am entitled to a copy of this form after it has been read and signed.

Signature of Participant

Date

Printed Name of Participant

Signature of Person Obtaining Consent

Date

BIBLIOGRAPHY

- [1] F. Abyarjoo, A. Barreto, S. Abyarjoo, F. R. Ortega, and J. Cofino. Monitoring human wrist rotation in three degrees of freedom. pages 1–5, 2013.
- [2] E. J. Alpenfels. The anthropology and social significance of the human hand. *Artificial limbs*, 2(2):4–21, 1955.
- [3] M. Alsheakhali, A. Skaik, M. Aldahdouh, and M. Alhelou. Hand gesture recognition system. *Information & Communication Systems*, page 132, 2011.
- [4] J. C. Alvarez, R. C. González, D. Alvarez, A. M. López, and J. Rodríguez-Uriá. Multisensor approach to walking distance estimation with foot inertial sensing. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 5719–5722. IEEE, 2007.
- [5] T. Auer and A. Pinz. Building a hybrid tracking system: Integration of optical and magnetic tracking. In *Augmented Reality, 1999.(IWAR’99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 13–22. IEEE, 1999.
- [6] B. Auvinet, E. Gloria, G. Renault, and E. Barrey. Runner’s stride analysis: comparison of kinematic and kinetic analyses under field conditions. *Science & Sports*, 17(2):92–94, 2002.
- [7] E. R. Bachmann, R. B. McGhee, X. Yun, and M. J. Zyda. Inertial and magnetic posture tracking for inserting humans into networked virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 9–16. ACM, 2001.
- [8] D. Bannach, O. Amft, K. S. Kunze, E. A. Heinz, G. Troster, and P. Lukowicz. Waving real hand gestures recorded by wearable motion sensors to a virtual car and driver in a mixed-reality parking game. In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pages 32–39. IEEE, 2007.
- [9] W. Barfield and T. Caudell. *Fundamentals of wearable computers and augmented reality*. Taylor & Francis, 2001.
- [10] P. Barralon, N. Noury, and N. Vuillerme. Classification of daily physical activities from a single kinematic sensor. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 2447–2450. IEEE, 2006.

- [11] G. Becq, S. Bonnet, L. Minotti, M. Antonakios, R. Guillemaud, and P. Kahane. Classification of epileptic motor manifestations using inertial and magnetic sensors. *Computers in biology and medicine*, 41(1):46–55, 2011.
- [12] S. S. Bhattacharyya, E. F. Deprettere, R. Leupers, and J. Takala. *Handbook of signal processing systems*.
- [13] A. Bourke, J. O'Brien, and G. Lyons. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & posture*, 26(2):194–199, 2007.
- [14] T. Bui and L. T. Nguyen. Recognizing postures in vietnamese sign language with mems accelerometers. *Sensors Journal, IEEE*, 7(5):707–712, 2007.
- [15] T. R. Burchfield and S. Venkatesan. Accelerometer-based human abnormal movement detection in wireless sensor networks. In *Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*, pages 67–69. ACM, 2007.
- [16] J. Bussmann, W. Martens, J. Tulen, F. Schasfoort, H. Van Den Berg-Emons, and H. Stam. Measuring daily behavior using ambulatory accelerometry: the activity monitor. *Behavior Research Methods, Instruments, & Computers*, 33(3):349–356, 2001.
- [17] R. F. Butler and R. F. Butler. *Paleomagnetism: magnetic domains to geologic terranes*, volume 319. Blackwell Scientific Publications Boston, 1992.
- [18] W. H. Campbell. *Introduction to geomagnetic fields*.
- [19] P. Cech, J. Dlouhy, M. Cizek, I. Vicha, and J. Rozman. Head position monitoring system design. In *Radioelektronika (RADIOELEKTRONIKA), 2010 20th International Conference*, pages 1–4. IEEE, 2010.
- [20] S. K. Charles. *It's all in the wrist: a quantitative characterization of human wrist control*. PhD thesis, Massachusetts Institute of Technology, 2008.
- [21] K.-Y. Chen, K. Lyons, S. White, and S. Patel. utrack: 3d input using two magnetic sensors. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 237–244. ACM, 2013.
- [22] J. Dlouhy, M. Cizek, and J. Rozman. Patients head position monitoring. In *Radioelektronika, 2007. 17th International Conference*, pages 1–3. IEEE, 2007.

- [23] R. Dorobantu and C. Gerlach. *Investigation of a navigation grade RLG SIMU type INAV-RQH*.
- [24] M. El-Diasty and S. Pagiatakis. Calibration and stochastic modelling of inertial navigation sensor errors. *Journal of Global Positioning Systems*, 7(2):170–182, 2008.
- [25] N. El-Sheimy, H. Hou, and X. Niu. Analysis and modeling of inertial sensors using allan variance. *Instrumentation and Measurement, IEEE Transactions on*, 57(1):140–149, 2008.
- [26] M. Elgendi, F. Picon, and N. Magenant-Thalmann. Real-time speed detection of hand gesture using, kinect. In *Proc. Workshop on Autonomous Social Robots and Virtual Humans, The 25th Annual Conference on Computer Animation and Social Agents (CASA 2012)*, 2012.
- [27] C.-S. Fahn and H. Sun. Development of a fingertip glove equipped with magnetic tracking sensors. *Sensors*, 10(2):1119–1140, 2010.
- [28] A. Fleury, N. Noury, and N. Vuillerme. A fast algorithm to track changes of direction of a person using magnetometers. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 2311–2314. IEEE, 2007.
- [29] M. Garton, A. Wutka, and A. Leuzinger. Local magnetic distortion effects on 3-axis compassing. *PNI Sens Corp Tech Reports*, 2009.
- [30] H. Ghasemzadeh, V. Loseu, E. Guenterberg, and R. Jafari. Sport training using body sensor networks: A statistical approach to measure wrist rotation for golf swing. In *Proceedings of the Fourth International Conference on Body Area Networks*, page 2. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.
- [31] D. Giansanti, G. Maccioni, F. Benvenuti, and V. Macellari. Inertial measurement units furnish accurate trunk trajectory reconstruction of the sit-to-stand manoeuvre in healthy subjects. *Medical & biological engineering & computing*, 45(10):969–976, 2007.
- [32] B. P. Gibbs. *Advanced Kalman filtering, least-squares and modeling: a practical handbook*. John Wiley & Sons, 2011.

- [33] M. S. Grewal and A. P. Andrews. *Kalman filtering: theory and practice using MATLAB*. John Wiley & Sons, 2011.
- [34] M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global positioning systems, inertial navigation, and integration*.
- [35] P. D. Groves. *Principles of GNSS, inertial, and multisensor integrated navigation systems*.
- [36] J. W. Harding, C. G. Mackintosh, A. G. Hahn, and D. A. James. Classification of aerial acrobatics in elite half-pipe snowboarding using body mounted inertial sensors (p237). In *The Engineering of Sport 7*, pages 447–456. Springer, 2008.
- [37] J. L. Hernandez-Rebollar, R. W. Lindeman, and N. Kyriakopoulos. A multi-class pattern recognition system for practical finger spelling translation. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, page 185. IEEE Computer Society, 2002.
- [38] G. Hirota, D. T. Chen, W. F. Garrett, M. A. Livingston, et al. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 429–438. ACM, 1996.
- [39] J. Hoff, E. Wagemans, J. Van Hilten, et al. Accelerometric assessment of levodopa-induced dyskinesias in parkinson’s disease. *Movement Disorders*, 16(1):58–61, 2001.
- [40] N. H. Hughes. Quaternion to euler angle conversion for arbitrary rotation sequence using geometric methods. *Accessed online at noelhughes.net/upload-s/quat*, 2, 2008.
- [41] X. Huo and M. Ghovanloo. Using unconstrained tongue motion as an alternative control mechanism for wheeled mobility. *Biomedical Engineering, IEEE Transactions on*, 56(6):1719–1726, 2009.
- [42] P. Jallon, S. Bonnet, M. Antonakios, and R. Guillemaud. Detection system of motor epileptic seizures through motion analysis with 3d accelerometers. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 2466–2469. IEEE, 2009.
- [43] D. A. James. The application of inertial sensors in elite sports monitoring. In *The Engineering of Sport 6*, pages 289–294. Springer, 2006.

- [44] L. Ji, Z. Qi, C. Dixiang, P. Mengchun, and L. Feilu. Integrated compensation method of three-axis magnetometer in geomagnetic navigation. In *Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on*, pages 929–932. IEEE, 2011.
- [45] M. Kangas, A. Konttila, I. Winblad, and T. Jamsa. Determination of simple thresholds for accelerometry-based parameters for fall detection. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1367–1370. IEEE, 2007.
- [46] J.-H. Kim, N. D. Thang, and T.-S. Kim. 3-d hand motion tracking and gesture recognition using a data glove. In *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on*, pages 1013–1018. IEEE, 2009.
- [47] K. King, S. Yoon, N. Perkins, and K. Najafi. Wireless mems inertial sensor system for golf swing dynamics. *Sensors and Actuators A: Physical*, 141(2):619–630, 2008.
- [48] B. Konvalin. Compensating for tilt, hard-iron and soft-iron effects. *Sens. Mag*, pages 1–11, 2009.
- [49] J. B. Kuipers. *Quaternions and rotation sequences*, volume 66. Princeton university press Princeton, 1999.
- [50] K. Kurita. Non-contact and non-attached human hand motion sensing technique for application to the human machine interface. In *SICE Annual Conference 2010, Proceedings of*, pages 3536–3539. IEEE, 2010.
- [51] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.
- [52] C.-F. Lai, S.-Y. Chang, H.-C. Chao, and Y.-M. Huang. Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling. *Sensors Journal, IEEE*, 11(3):763–770, 2011.
- [53] A. Lawrence. *Modern inertial technology: navigation, guidance, and control*. Springer Science & Business Media, 1998.
- [54] Y. Li, L. Shark, S. J. Hobbs, and J. Ingham. Real-time immersive table tennis game for two players with motion tracking. In *Information Visualisation (IV), 2010 14th International Conference*, pages 500–505. IEEE, 2010.

- [55] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–7. IEEE, 2011.
- [56] K. Maenaka. Mems inertial sensors and their applications. In *Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on*, pages 71–73. IEEE, 2008.
- [57] M. Mubashir, L. Shao, and L. Seed. A survey on fall detection: Principles and approaches. *Neurocomputing*, 100:144–152, 2013.
- [58] C. Naranjo. Analysis and modeling of mems based inertial sensors. *Kungliga Tekniska Kgsolan, Stockholm, School of Electrical Engineering*, 2008.
- [59] J. Nikolic and F. Renaut. Mems inertial sensors technology.
- [60] C. Nölker and H. Ritter. Detection of fingertips in human hand movement sequences. In *Gesture and Sign Language in Human-Computer Interaction*, pages 209–218. Springer, 1998.
- [61] E.-J. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 889–894. IEEE, 2004.
- [62] S. Park, S. Yu, J. Kim, S. Kim, and S. Lee. 3d hand tracking using kalman filter in depth space. *EURASIP Journal on Advances in Signal Processing*, 2012(1):1–18, 2012.
- [63] L. Perumal. Quaternion and its application in rotation using sets of regions. *International Journal of Engineering and Technology Innovation*, 1:35–52, 2011.
- [64] H. C. Powell, M. A. Hanson, and J. Lach. On-body inertial sensing and signal processing for clinical assessment of tremor. *Biomedical Circuits and Systems, IEEE Transactions on*, 3(2):108–116, 2009.
- [65] M. Rahman, S. Gustafson, P. Irani, and S. Subramanian. Tilt techniques: investigating the dexterity of wrist-based input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1943–1952. ACM, 2009.

- [66] M. A. Rahman and A. Al-Jumaily. Design and development of a hand exoskeleton for rehabilitation following stroke. *Procedia Engineering*, 41:1028–1034, 2012.
- [67] R. Raya, J. Roa, E. Rocon, R. Ceres, and J. Pons. Wearable inertial mouse for children with physical and cognitive impairments. *Sensors and Actuators A: Physical*, 162(2):248–259, 2010.
- [68] G. Reitmayr and T. W. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, pages 109–118. IEEE, 2006.
- [69] V. Renaudin, M. H. Afzal, and G. Lachapelle. Complete triaxis magnetometer calibration in the magnetic domain. *Journal of sensors*, 2010, 2010.
- [70] O. Rumyantsev, M. Merati, and V. Ramachandran. Hand sign recognition through palm gesture and movement. *Image Processing*, 2012.
- [71] G. Saggio, F. Giannini, M. Todisco, and G. Costantini. A data glove based sensor interface to expressively control musical processes. In *Advances in Sensors and Interfaces (IWASI), 2011 4th IEEE International Workshop on*, pages 192–195. IEEE, 2011.
- [72] A. Salarian, H. Russmann, C. Wider, P. R. Burkhard, F. J. Vingerhoets, and K. Aminian. Quantification of tremor and bradykinesia in parkinson’s disease using a novel ambulatory monitoring system. *Biomedical Engineering, IEEE Transactions on*, 54(2):313–322, 2007.
- [73] J. Säll and J. Merkel. Indoor navigation using accelerometer and magnetometer. 2011.
- [74] T. S. Saponas, D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay. Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 167–176. ACM, 2009.
- [75] N. Sato, T. Takayama, and Y. Murata. Early evaluation of automatic flying distance measurement on ski jumper’s motion monitoring system. In *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, pages 838–845. IEEE, 2013.

- [76] D. K. Shaeffer. Mems inertial sensors: A tutorial overview. *Communications Magazine, IEEE*, 51(4):100–109, 2013.
- [77] D. Simon. Least squares estimation. *Optimal State Estimation: Kalman, H, and Nonlinear Approaches*, pages 79–105.
- [78] H. W. Sorenson. *Kalman filtering: theory and application*. IEEE, 1960.
- [79] P. Strohmeier, R. Vertegaal, and A. Girouard. With a flick of the wrist: stretch sensors as lightweight input for mobile devices. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, pages 307–308. ACM, 2012.
- [80] Y. S. Suh. Orientation estimation using a quaternion-based indirect kalman filter with adaptive estimation of external acceleration. *Instrumentation and Measurement, IEEE Transactions on*, 59(12):3296–3305, 2010.
- [81] Y. Tao, H. Hu, and H. Zhou. Integration of vision and inertial sensors for 3d arm motion tracking in home-based rehabilitation. *The International Journal of Robotics Research*, 26(6):607–624, 2007.
- [82] K. N. Tarchanidis and J. N. Lygouras. Data glove with a force sensor. *Instrumentation and Measurement, IEEE Transactions on*, 52(3):984–989, 2003.
- [83] C. L. Taylor and R. J. Schwarz. The anatomy and mechanics of the human hand. *Artificial limbs*, 2(2):22–35, 1955.
- [84] D. Titterton and J. L. Weston. *Strapdown inertial navigation technology*.
- [85] A. Tognetti, N. Carbonaro, G. Zupone, and D. De Rossi. Characterization of a novel data glove based on textile integrated sensors. In *Engineering in Medicine and Biology Society, 2006. EMBS’06. 28th Annual International Conference of the IEEE*, pages 2510–2513. IEEE, 2006.
- [86] D. T. Wade and B. A. de Jong. Recent advances: Recent advances in rehabilitation. *BMJ: British Medical Journal*, 320(7246):1385, 2000.
- [87] O. J. Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14:15, 2007.

- [88] C.-H. Wu, Y.-T. Chang, and Y.-C. Tseng. Multi-screen cyber-physical video game: An integration with body-area inertial sensor networks. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 832–834. IEEE, 2010.
- [89] G. Wu, F. C. Van Der Helm, H. D. Veeger, M. Makhsous, P. Van Roy, C. Anglin, J. Nagels, A. R. Karduna, K. McQuade, X. Wang, et al. Isb recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motionpart ii: shoulder, elbow, wrist and hand. *Journal of biomechanics*, 38(5):981–992, 2005.
- [90] G. Wu and S. Xue. Portable preimpact fall detector with inertial sensors. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 16(2):178–183, 2008.
- [91] S. You, U. Neumann, and R. Azuma. Hybrid inertial and vision tracking for augmented reality registration. In *Virtual Reality, 1999. Proceedings., IEEE*, pages 260–267. IEEE, 1999.
- [92] Y. Yuksel. Huseyin burak kaygisiz.
- [93] X. Yun, M. Lizarraga, E. R. Bachmann, and R. B. McGhee. An improved quaternion-based kalman filter for real-time tracking of rigid body orientation. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1074–1079. IEEE, 2003.
- [94] Yury. ellipsoid-fit, 2009. [zip file].
- [95] T. E. Zander. *Applied low-cost inertial sensor systems and efficient measurement data processing*.
- [96] X. Zhang, X. Chen, W.-h. Wang, J.-h. Yang, V. Lantz, and K.-q. Wang. Hand gesture recognition and virtual game control based on 3d accelerometer and emg sensors. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 401–406. ACM, 2009.
- [97] H. Zhou, H. Hu, and N. Harris. Application of wearable inertial sensors in stroke rehabilitation. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 6825–6828. IEEE, 2005.

- [98] H. Zhou, T. Stone, H. Hu, and N. Harris. Use of multiple wearable inertial sensors in upper limb motion tracking. *Medical engineering & physics*, 30(1):123–133, 2008.
- [99] C. Zhu, Q. Cheng, and W. Sheng. Human intention recognition in smart assisted living systems using a hierarchical hidden markov model. In *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*, pages 253–258. IEEE, 2008.
- [100] D. G. Zwartjes, T. Heida, J. P. van Vugt, J. A. Geelen, and P. H. Veltink. Ambulatory monitoring of activities and motor symptoms in parkinson’s disease. *Biomedical Engineering, IEEE Transactions on*, 57(11):2778–2786, 2010.

VITA
FATEMEH ABYARJOO

EDUCATION

2011 - 2015 (expected)	Ph.D., Electrical and Computer Engineering
Florida International University	Miami, FL
2003 - 2005	M.S., Mechatronic Engineering
Azad University	Qazvin, Iran
1999 - 2002	B.S., Computer Engineering
Azad University	Qazvin, Iran

PUBLICATIONS AND PRESENTATIONS

Francisco R. Ortega, Fatemeh Abyarjoo, Armando Barreto, Naphthali Rishe, Malek Adjouadi. *“3D User Input Interfaces”*. CRC Press, 2015.

F. Abyarjoo, A. Barreto, N. Rishe, F. Ortega, *“Hand Motion Detection using MEMS IMU/ magnetic sensors”*, (under preparation).

P. Reng, F. Abyarjoo, F. Ortega, X. Ma, , *“Feasible Feature Extraction for Activity Recognition based on the Derivation of Accelerometer Signal of Smartphone”*, submitted to IEEE Journal of Pervasive Computing.

Cofino, J., Barreto, A., Abyarjoo, F., Ortega, F., *“Blind Assistive Spatialized Screen-reading”*, Journal on Technology and Persons with Disabilities (JTPD), Vol. 2, (2014) pp. 46-57

F. Abyarjoo, N. O-larnnarit, S. Tangnimitchok, A. Barreto, *“PostureMonitor: Real-Time IMU Wearable Thecnology to Foster Poise and Health”*, HCI International, (2015).

F. Abyarjoo, A. Barreto, S. Abyarjoo; J. Cofino, F. Ortega, *“Monitoring Human Wrist Rotation in Three Degrees of Freedom”*, IEEE SoutheastCON 2013.

F. Abyarjoo, A. Barreto, J. Cofino, F. Ortega, “*Implementing a Sensor Fusion Algorithm for 3D Orientation Detection with Inertial/ Magnetic Sensors*”, Proceedings of CISSE 2012: The International Joint Conferences on Computer, Information and Systems Sciences and Engineering, December 7–9, 2012, Bridgeport, CT. Springer 2014. LNEE 152, pp.

“3D Human Hand Motion Detection Using MEMS Inertial and Magnetic Sensors”
ACM Richard Tapia Celebration of Diversity in Computing
Poster Presentation - Boston, MA (Feb, 2015)

“Cloud-based Wearable Devices”
Mobile Technology Consortium- Internet of Every Thing
Florida International University - Miami, FL (26 Sep, 2014)

“3D Human Hand Motion Detection Using MEMS Inertial and Magnetic Sensors”
ACM Richard Tapia Celebration of Diversity in Computing
Poster Presentation - Boston, MA (Feb, 2015)

“Monitoring Human Wrist Rotation in Three Degrees of Freedom”
Assistive Technology Industry Association
IEEE Southeast Conference - Jacksonville, FL (April, 2013)