INERTIAL MEASUREMENT UNITS FOR MOTION CONTROL OF SURGICAL ENDOSCOPIC ROBOT

A Thesis Submitted to
The Graduated School of Engineering and Science
Izmir Institute of Technology
In Partial Fulfillment of Requirements for the Degree of

MASTER OF SCIENCE

in Electronics and Communication Engineering

by
Sercan ERAT

June 2018
IZMIR

We approve the thesis of **Sercan ERAT**

Examining Committee Members:

_____

Faculty Member Dr. Barbaros ÖZDEMİREL
Department of Electrical and Electronics Engineering
Izmir Institute of Technology

_____

Assoc. Prof. Dr. Şevket GÜMÜŞTEKİN
Department of Electrical and Electronics Engineering
Izmir Institute of Technology

_____

Prof. Dr. Mehmet ENGİN
Electrical and Electronics Engineering Department
Ege University

29 June 2018

_____

Faculty Member Dr. Barbaros ÖZDEMİREL
Supervisor, Department of Electrical - Electronics Engineering, İzmir Institute of Technology

_____

Prof. Dr. Enver TATLICIOĞLU
Head of The Department of
Electrical and Electronics Engineering

_____

Prof. Dr. Aysun SOFUOĞLU
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGEMENTS

I would like to thank to the thesis supervisor Faculty Member Dr. Barbaros ÖZDEMİREL for his guidance, help, endless support, and trust throughout this thesis. This study would not be possible without my supervisor's help and support. Also, I express my special thanks to my family for their help, motivations and support.

This work was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) (Project No: 115E725 and 115E726). I would like to express a thanks to Assoc. Prof. Dr. Can DEDE for giving me the opportunity to work on this project.

ABSTRACT

INERTIAL MEASUREMENT UNITS FOR MOTION CONTROL OF SURGICAL ENDOSCOPIC ROBOT

Inertial measurement units (IMUs) are preferable over other electromechanical user interfaces for robotics applications that require hands-free operation. In case of a surgical endoscopic robot, one of the requirements is a user interface that gives the surgeon ability to control the camera position while holding other surgical tools. An IMU used for this purpose should provide precise position information in a small motion range restricted by a few degrees of rotation.

In this study, major difficulties in establishing a user interface between an IMU and a surgical robotic arm are identified, and the necessary solutions for implementation of such an interface are presented. The study focuses on the IMU devices that combine accelerometers and gyroscopes based on micro-electromechanical systems (MEMS) technology because of their smaller size and low power requirement. Working principles and the calibration techniques of MEMS IMUs are reviewed. Three fusion algorithms that combine accelerometer and gyroscope data to determine the orientation of an object in three-dimensional space are implemented. Performance of these fusion algorithms and critical specifications of MEMS IMUs from three different manufacturers are evaluated by monitoring sensor data through a wireless interface.

A graphical evaluation software was developed to test position mapping methods by steering field of view of a hypothetical camera towards randomly determined targets on the computer screen. Effectiveness of the position mapping methods was evaluated by several users by comparing the total angular displacement and the elapsed time required to reach a number of simulated targets.

ÖZET

CERRAHİ ENDOSKOPİ ROBOTUNUN HARAKET DENETİMİ İÇİN ATALETSEL ÖLÇÜM BİRİMLERİ

Ataletsel ölçüm birimleri (AÖB) ellerin serbest olarak kullanılmasını gerektiren robotik uygulamalarda diğer elektromekanik kullanıcı arabirimlerine kıyasla tercih edilebilir. Cerrahi endoskopi robotunun kontrolüne yönelik kullanıcı arabiriminin gereklerinden biri, cerrahın diğer cerrahi aletleri tutarken kamera konumunu kontrol edebilmesidir. Bu amaç için kullanılacak bir AÖB birkaç derecelik dönüşle sınırlandırılmış küçük bir hareket aralığında kesin konum bilgisi sağlamalıdır.

Bu çalışmada, AÖB ve cerrahi robot kolu arasında bir kullanıcı arabirimi oluşturulmasındaki güçlükler belirlenmiş ve böyle bir arabirimin uygulanması için gerekli çözümler sunulmuştur. Çalışma kapsamında, daha küçük boyutlu olmaları ve düşük güç gerektirmeleri nedeniyle mikro-elektromekanik sistemler (MEMS) teknolojisine dayalı ivmeölçerler ve jiroskopların bir araya gelmesiyle oluşan AÖB aygıtlarına odaklanılmıştır. Bir nesnenin üç boyutlu uzaydaki konumunu belirlemek için ivmeölçer ve jiroskop verilerini birleştiren üç füzyon algoritması incelenmiştir. Bu füzyon algoritmalarının performansı ve üç farklı üreticiden sağlanan MEMS AÖB'lerinin kritik özellikleri kablosuz bir arayüz aracılığıyla elde edilen algılayıcı verilerini izleyerek değerlendirilmiştir.

Kontrol amaçlı konum eşleme yöntemlerini değerlendirmek için bilgisayar ekranı üzerinde bir varsayımsal kamera görüş alanını rasgele belirlenen hedeflere doğru yönlendiren bir grafiksel değerlendirme uygulaması geliştirilmiştir. Konum eşleme yöntemlerinin etkinliği, farklı kullanıcılar tarafından bir dizi rastgele belirlenmiş hedefe ulaşmak için yapılan denemelerde toplam açısal yer değiştirmeyi ve geçen süreyi karşılaştırarak değerlendirilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations

BLE	Bluetooth low energy
DDB	Discovery development board
DOF	Degree of freedom
DPS	Degree per second
FOG	Fiber optic gyroscope
FOV	Field of view
GAP	Generic access profile
GATT	Generic attribute profile
GPS	Global positioning system
I2C	Inter-integrated circuit
IMU	Inertial measurement unit
MEMS	Microelectromechanical systems
MTC	Motion tracking card
ODR	Output data rate
RLG	Ring laser gyroscope
RPS	Revolution per second
SIG	Special interest group
SPI	Serial peripheral interface
UART	Universal asynchronous receiver-transmitter
UUID	Universally unique identifier

CHAPTER 1

INTRODUCTION

Robotic technology has been increasingly used in surgical procedures for about 20 years. Many successful surgical operations during this period have shown the success and reliability of this technology (*1*). This thesis focuses on the robotic control of the endoscopic camera that monitors the area of surgical operation. Owing to that, the view of the surgical area is a key parameter for surgical success and directly affects the surgeon's performance during surgery (*2*).

The control of the endoscope is a major problem for the surgeon during surgery. The surgeon directly controls the surgical tools while there is an endoscope inside the operation area to monitor the tools and the target tissues. The endoscope is either positioned by the surgeon, or it is controlled by an assistant, following the verbal commands given by the surgeon. In the first case, the surgeon can use only one hand to control the surgical tools, and precise positioning of the endoscope for long periods of time causes fatigue. In the second case, the non-autonomous procedure restricts and slows down the surgeon's control over the surgical area. This non-autonomous control procedure during the surgical operation can be seen in Figure 1.1.

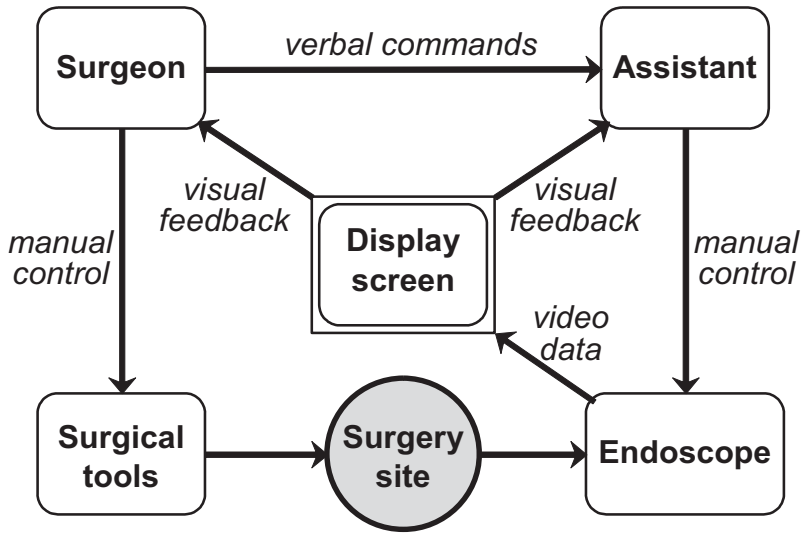


Figure 1.1. Traditional control of the endoscope

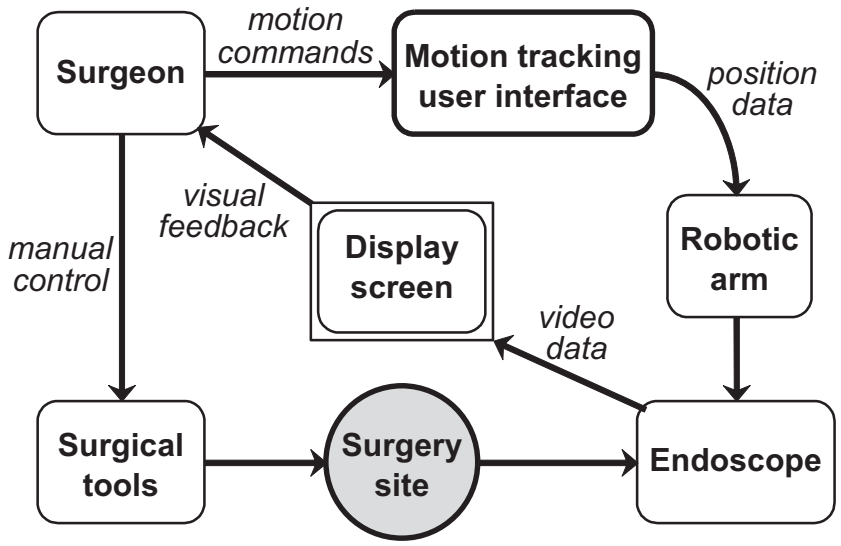


Figure 1.2. Autonomous control of the endoscope with a robotic arm

The autonomous control procedure shown in Figure 1.2 is the basis of a system that will allow the surgeon to directly control the endoscope by using robotic positioning. In this setup, the movement of the surgeon is being detected by using an orientation tracking system. Then, this orientation information is sent wirelessly to the robot control unit. The robot is thus controlled with this orientation information. The following matters are covered in this thesis to achieve this purpose,

- Selection of sensors to detect the orientation of user interface.
- Review of working principles of the sensors used in the study.
- Algorithms for calculating the orientation of the user interface by using various sensors measurements.
- Calibration techniques for the sensors.
- Wireless transmission of sensor data to the robot control unit.
- Evaluation of methods for mapping orientation information to target settings in the robot control process.

The motion tracking user interface contains an inertial measurement unit (IMU) structure that consists of an accelerometer and a gyroscope. A classical IMU has a three-axis accelerometer and a three-axis gyroscope. Each sensor unit can make measurements in the x, y and z axes which are orthogonal to each other. Therefore, the IMU that contains a three-axis accelerometer and three-axis gyroscopes has six degrees

of freedom (DOF). The gyroscope senses the change in orientation and the accelerometer measures the linear acceleration. The accelerometers and gyroscopes have both drawbacks and advantages due to the nature of the sensors and the measurement conditions. For instance, the accelerometer is sensitive to vibration, and it is constantly affected by gravity, whereas angular orientation data obtained by integration of the gyroscope output drifts continuously. For this reason, outputs of the accelerometer and the gyroscope are combined to compensate the disadvantages of each other and to estimate the orientation of IMU by using sensor fusion algorithms. In addition, a magnetometer sensor can also be used in the sensor fusion algorithms. However, the magnetometer sensor is not considered in this thesis because there are several ferromagnetic metal objects in the operating room that affect the reliability of the magnetometer measurements.

The surgeon controls a robotic arm that holds the endoscope by changing the orientation of the motion tracking device while he is working with other surgical tools. Therefore, the surgeon has to make slight movements in a hazardous environment. Micro-electro-mechanical systems (MEMS) based IMUs are used to detect these minimal movements of the surgeon in three-dimensional space. MEMS IMUs are selected for this purpose because of their small structure, easy communication with a microprocessor and low power consumption. The accuracy and reliability of the MEMS IMUs are also critical parameters, since changes in the orientation should be detected with better than one-degree accuracy. The MEMS IMUs from different manufacturers are evaluated to choose the best among them. The key parameters in this comparison are the stability, resolution and noise. The power consumption of the MEMS IMU is another critical parameter, because the MEMS IMU data are sent wirelessly (i.e. Bluetooth) and the motion tracking card (MTC) works with limited battery power.

Bluetooth communication is used to send IMU data wirelessly to the robot control unit. The surgeon must push the activation button on the MTC whenever he wants to control the endoscope. A mechanical vibration is generated when the activation button is pressed, which affects the accelerometer measurements. This mechanical vibration is another problem that the thesis will tackle on. Also, the motion sensor is activated for brief periods of time during a surgical operation, and the motion sensor should determine its orientation after every activation, and the system must adopt the motion mapping dictated by the operator.

A typical motion tracking system is obtained by combining low-cost IMU modules with a microprocessor supporting wireless communication that are available commercially. All microcontrollers used in this thesis have ARM-based processors, which is another requirement in order to have an onboard low-energy Bluetooth module for wireless communication.

A motion monitoring system was built to evaluate tracking efficiency of the motion sensor card with an evaluation software running on a PC. The evaluation software was developed by using ***Processing*** utility which is a flexible and open source software. Sensor fusion algorithms for sensor data processing, sensor data collection for sensor evaluation, and targeting algorithms for simulation of robot control were performed on this software.

To sum up the goals and challenges in this thesis; firstly, the motion tracking system with IMU is built. Firmware development covers the configuration and the calibration of the accelerometer and gyroscope sensors. The accuracy for small angular motion range and short integration time is evaluated considering resolution, stability, and noise as the critical factors. The second aim is to analyze the calibration techniques. The calibration technique must be quick and reliable because the sensors should be calibrated according to the operation room environment before every surgical operation. Calibration techniques have been examined, and the most appropriate firmware for auto-calibration has been created. Furthermore, the practical issues are solved such as power consumption, battery life, yaw angle orientation and vibration caused by the activation button on the MTC. The third aim is to transmit sensor data wirelessly by using Bluetooth communication. The last aim is to develop an evaluation software with a graphical user interface to measure and display the speed and efficiency in targeting experiments on a computer monitor.

1.1. Thesis Structure

Throughout this sub-section, the structure of the thesis is presented with brief information about chapters. Chapter 2 provides the background information about accelerometers and gyroscopes. Accelerometer and gyroscope sensors' working principles are briefly explained, including general information about the common

accelerometer and gyroscope types. A review of control applications utilizing MEMS IMUs and various user interface mechanisms for surgical robotic instruments are given.

Chapter 3 provides information about sensor fusion algorithms. Firstly, the information about coordinate system, Euler angles, and quaternion are given. Secondly, the complementary filter, Madgwick filter and Mahony filter algorithms are formulized in this section. Lastly, calculation of gravity and linear acceleration by using IMU measurements is addressed.

Chapter 4 describes the hardware components used in this study. Information is given about the microprocessors that are used in this thesis, namely STM32F407 from STMicroelectronics and NRF51822 from Nordic Semiconductor. Two cards are used to implement an experimental system to transfer IMU data to the evaluation software wirelessly, which are a motion tracking card (MTC) and a scanner card. These cards establish a Bluetooth Low Energy (BLE) connection to transfer motion data from the MTC to the scanner card. The scanner card sends the data received from the MTC to the PC (i.e. evaluation software) through USB interface.

Chapter 5 contains a comparison of technical specifications given for three different MEMS IMUs, namely MPU6050 from Inversense, BMX055 from Bosch, and LSM6DS3 from STMicroelectronics. Also, the terminology of MEMS accelerometer and gyroscope specifications in sensor datasheets are explained. Calibration techniques for accelerometers and gyroscopes starting with offset error calibration are described. Output stability and thermal stability of the accelerometers and gyroscopes are evaluated. Lastly, an automatic calibration algorithm is introduced.

Chapter 6 provides information about evaluation software and position mapping methods to control the surgical endoscopic robot. There are two mapping methods that were evaluated while controlling the robot arm according to the MTC orientation. In the first method, the position of the MTC controls the robot's position. In the second method, the MTC's position controls the robot's velocity.

Chapter 7 gives a summary of the conclusion of the thesis and an overview of the future work.

CHAPTER 2

INERTIAL MOTION SENSORS

Throughout this chapter, principles of accelerometer and gyroscope sensors are presented. Specifically, brief information about sensor types, details of MEMS accelerometers, MEMS gyroscopes and terminology of these sensors are given in separate sub-sections.

2.1. Accelerometer

The accelerometer is a device which can detect and measure linear acceleration. By using acceleration a, the velocity v is calculated with single integration and position x is calculated with double integration, as shown below:

$$v = \int a \ dt \tag{2.1}$$

$$x = \iint a \ dt^2 \tag{2.2}$$

The output of accelerometer is in m/s^2 or 'g' term equivalent to the earth's gravity, which is roughly equal to 9.81 m/s^2. The accelerometer sensor measures 1 g constant gravity on earth that slightly varies according to the elevation and location around the world. An accelerometer also senses the forces resulting from acceleration and vibration. In brief, the source of the measured acceleration may be the static force of gravity, or it may be dynamic, caused by accelerometer motion or vibration (*3*). The 'g' term is also used to describe the measurement range of an accelerometer. While some accelerometers can operate in a single measurement range, it is possible to change parameters such as, measurement range, resolution, and sampling rate on most of the digital accelerometer sensors.

Newton's second law of motion expresses the connection between the motion of a body and the forces acting on it (*4 p. 153*). The acceleration a of a body of mass m resulting from the applied force F is given by:

$$F = ma \tag{2.3}$$

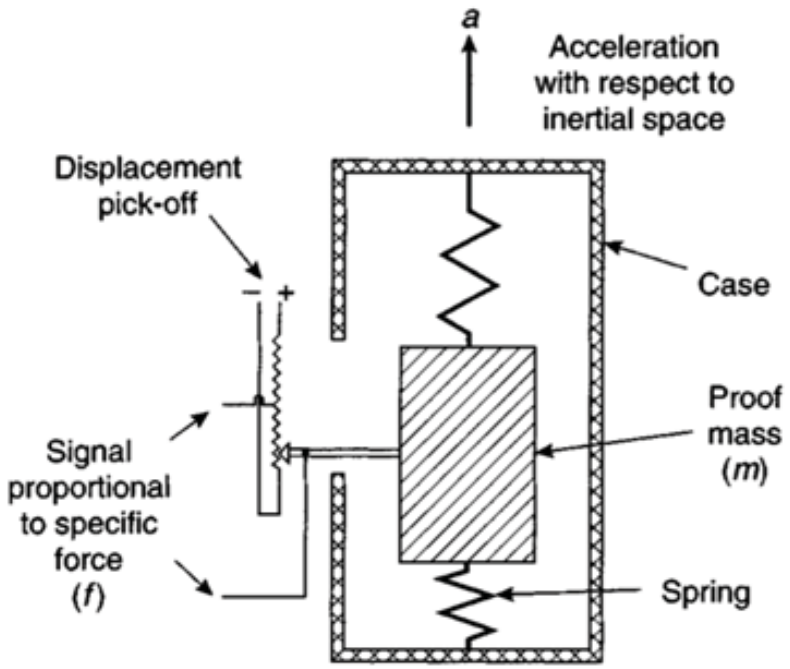


Figure 2.1. Accelerometer mass-spring structure (*4 p. 154*)

The accelerometer can be simply described as a mass-string structure and it consists of a mass attached to a frame through couple of springs, as shown in Figure 2.1. If an external force is applied to this accelerometer structure, the proof mass tends to remain motionless, and resists to this force due to its inertia. Consequently, the relative displacement of the mass due to this force causes tension on the springs, and the accelerometer detects the force. The displacement resulting from the force and acceleration are proportional to each other, and the displacement of the mass is used to calculate the acceleration. The total force F_{tot} acting on the mass is calculated as:

$$F_{tot} = ma_{tot} = mg + ma_{mov} \tag{2.4}$$

$$a_{tot} = g + a_{mov} \tag{2.5}$$

where, g and a_{mov} correspond to the gravity and the acceleration resulting from movement respectively. If the structure of an accelerometer is aligned with z-axis and there is no additional force caused by movement, then it senses +1 g acceleration due to

gravity. It is considered that it must measure -1 g acceleration, since the gravity is acting on the mass towards the negative z-axis. However, it actually measures the force experienced by the proof mass against gravity.

The accelerometer can also measure sideways movement along the other sensitive axes as represented by a_{mov}. The detection of the acceleration in the sideway movements of the accelerometer sensors are seen in Figure 2.2. The proof mass is held in balance between the four force transducers. If the box starts to accelerate at 1 m/s^2 in positive X direction as shown in Figure 2.2, then the transducer in the negative X direction senses 1 N force. As seen in this example, an accelerometer indirectly measures the force resulting from the sideway acceleration. Additionally, the output of the transducer in the negative Z direction is 9.81 N due to gravity. The accelerometer gives an output of 1 g constant acceleration in the positive Z direction as a result of the gravity in the negative Z direction.

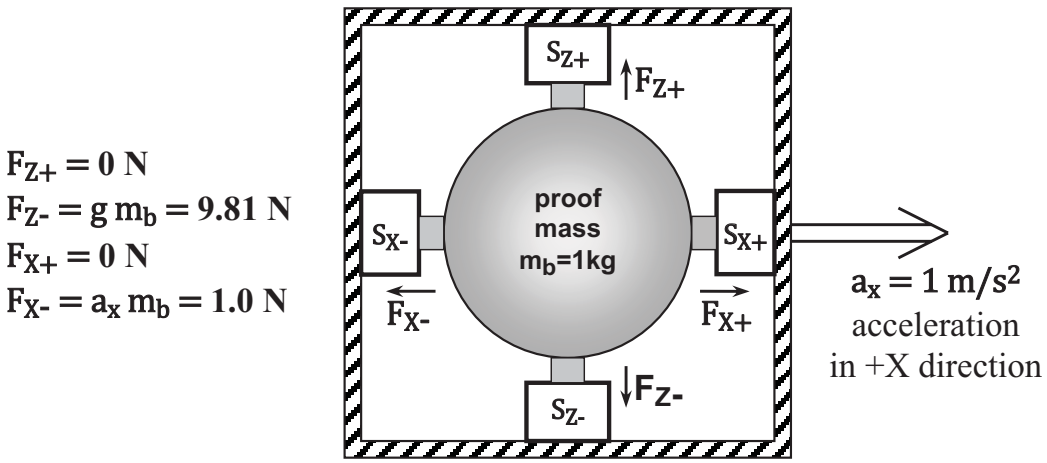


Figure 2.2. The proof mass system of the accelerometer

If an accelerometer is dropped, it provides 0 g measurement in all of its axes during free fall, because the gravity drags both the mass and the frame of the accelerometer. In this way, an accelerometer can detect the free fall condition. As an example of the practical use of free fall detection, some disk manufacturers use an accelerometer to protect hard disk drives in portable computers. If the user suddenly drops the computer, the accelerometer detects this free fall, and secures the disk drive of the computer (*5*).

2.1.1. Accelerometer Types

The accelerometers can be divided into three groups according to their manufacturing technology: mechanical, solid state, and MEMS. In this section, only the basic principles about the most common sensor types are given although there are many different types of accelerometers in addition to those covered in this section. More information about accelerometers can be found in (*6*).

Mechanical Accelerometer

A mechanical accelerometer contains a mass-spring structure and a frame, as seen in Figure 2.1. As mentioned before, the displacement of mass is used to measure the force resulting from acceleration, and it is detected by a detector. The output signal of the detector is proportional to the force F acting on the mass along its sensitive axis. Consequently, the acceleration of a mechanical accelerometer can be calculated by using Newton's second law of motion (*6*).

Solid State Accelerometer

Solid-state accelerometers have several types, such as, surface acoustic wave, vibratory or quartz devices (*4 p. 161*). The surface acoustic wave (SAW) accelerometer can be given as an example of a solid-state accelerometer. SAW accelerometer is made up of a cantilever beam which is attached to the frame and carries a proof mass on its other end, as shown in Figure 2.3. The SAW transducers on the cantilever beam resonate at a specific frequency. When there is a force due to acceleration acting on the beam along its sensitive axis, the beam bends and changes the resonance frequency. As a result, the acceleration is calculated by sensing the frequency change (*6*).

MEMS Accelerometer

MEMS accelerometers have the same physical principle as mechanical or solid-state sensors (*6*). The structures of these sensors are etched into silicon by using MEMS manufacturing techniques. The accuracy levels of solid-state accelerometers cannot be reached by using MEMS accelerometer, however, MEMS accelerometers have many advantages. The main advantage of MEMS accelerometer is its easy connection to electronic circuit elements near the sensor and simple communication with a

microprocessor through I2C or SPI interface. Also, MEMS accelerometers have many other advantages (*6*), such as:

- small size,
- low weight,
- low power consumption,
- short start-up time,
- inexpensive to produce,
- compatible with operations in hostile environments.

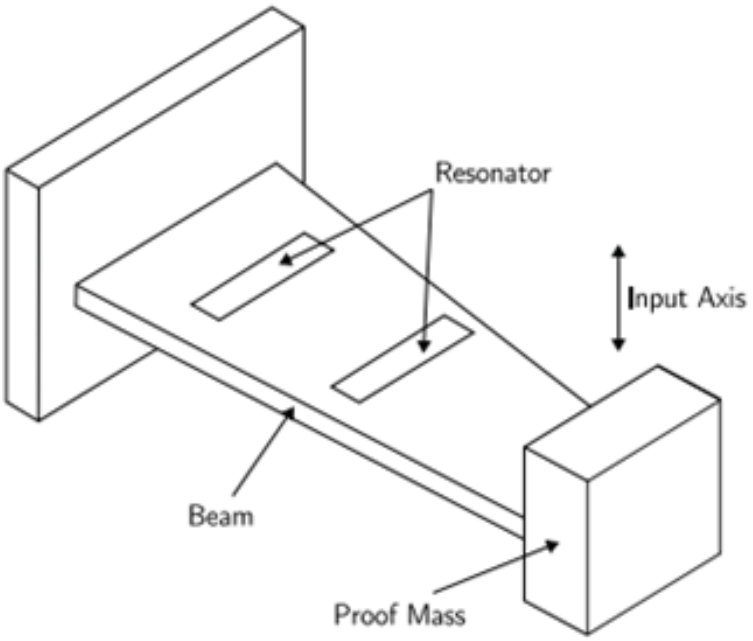


Figure 2.3. A surface acoustic wave accelerometer (*6*)

2.2. Gyroscope

A gyroscope is a device that measures angular velocity without using any external reference such as the earth's magnetic field or gravity. The output ω of a gyroscope corresponds to the time derivative of angular position ϕ.

$$\omega = \frac{\partial \phi}{\partial t} \tag{2.6}$$

When gyroscope sensor is rotated in clockwise direction, it gives a positive output proportional to the angular velocity. If the gyroscope is rotated counter clockwise, it provides negative output signal. The output signal of the gyroscope is related to the angular velocity in degree per second (º/s) or revolution per second (RPS) (*7*). The gyroscope's output signal must be integrated to find angular position change around an axis. However, the initial position of an object relative to the reference earth frame cannot be determined by using the gyroscope alone.

The use of gyroscope alone in the calculation of rotation angle causes inaccurate results over time. The most important reason for this error is the integration required for calculation of the rotation angle. This error is named as gyroscopic drift. As the measurement time increases, the drift error in the calculated rotation angle increases as well, unless the drift error is not eliminated. In this work, this drift of the gyroscope is corrected around x and y axes by using gravity component of the accelerometer output. Additionally, a magnetometer that measures the earth's magnetic field, can be used to compensate gyroscopic drift around z-axis.

2.2.1. Gyroscope Types

The common gyroscope sensors can be divided into three groups, mechanical (spinning disk), optical and MEMS. There are many different types of gyroscopes in addition to those mentioned in this section. More information about these sensors can be found in (*4*) and (*8*)**.**

Mechanical Gyroscope

A mechanical gyroscope is also named as spinning wheel gyroscope since this gyroscope is consists of a spinning wheel attached to its two gimbals which provide rotation in two axes as seen in Figure 2.4.

Angular changes are detected by angle pick-offs in mechanical gyroscopes. The angle between the wheel and gimbals will change since the spinning wheel resists orientation changes, and the mechanical gyroscope detects the angular position based on this difference. The main problem of the mechanical gyroscopes is the presence of moving elements that cause friction. The friction of the moving elements causes drift in

angle measurements. Also, the mechanical gyroscope needs a long start-up time to get the system ready (*6*).

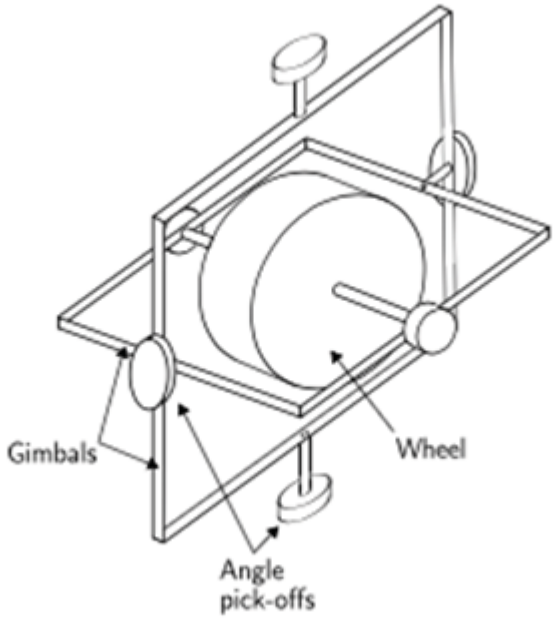


Figure 2.4. Mechanical Gyroscope (*6*)

Optical Gyroscope

Optical gyroscopes mainly include fiber optic gyroscopes and ring laser gyroscopes. A fiber optic gyroscope (FOG) contains a coil of optical fiber, and it uses the interference of beams. Two beams are sent into a coil in the opposite directions to detect any rotational changes. If the FOG sensor rotates in one direction, the beam in the same direction has to take a longer distance which is known as Sagnac effect.

As seen in Figure 2.5, ω refers to the angular velocity and rotation direction. The dashed line refers to the path that a beam travels in the direction of rotation, and the solid line refers to the path that the other beam travels in the opposite direction. The rotation angle ϕ is expressed as the difference between the two beam paths due to the rotation of the gyroscope (*6*).

In order to detect the Sagnac effect, an interferometric method is used, where the two light beams are combined with each other when they come out of the fiber. The intensity of the new combined beam changes as a function of the angular velocity, and FOG uses the intensity of the combined beam to sense the angular rotation.

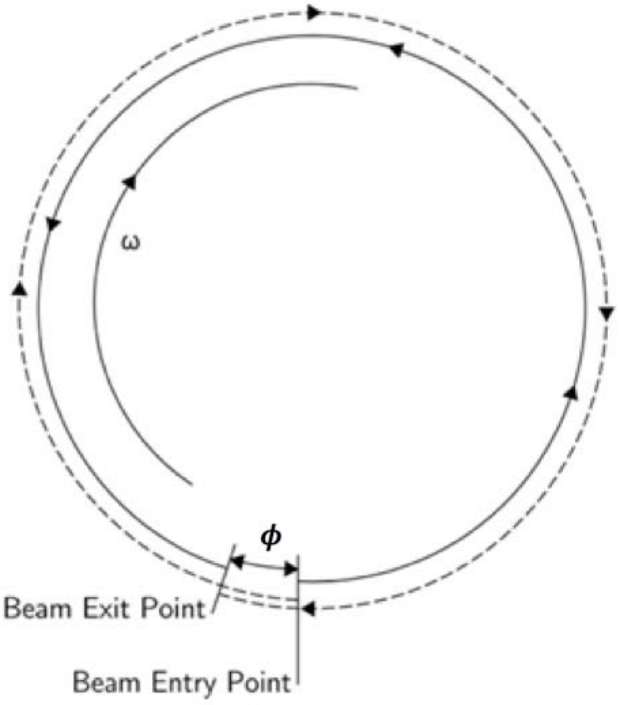


Figure 2.5. Sagnac effect in optical gyroscope (*6*)

The ring laser gyroscope (RLG) is another example of the optical gyroscopes, and they also use the Sagnac effect phenomenon. RLG consists of mirrors instead of optical fibers and the laser beams travel around a closed path determined by the mirrors.

The main advantage of optical gyroscopes is their operation without any moving parts that provides a much shorter start-up time. The optical gyroscope provides more accurate output compared to other gyroscope types. Additionally, the sensitivity of an optical gyroscope increases if the beams travel a longer path, however, this situation affects the size of an optical gyroscope.

MEMS Gyroscope

Mechanical and optical gyroscopes are still complex and expensive devices despite the improvements in recent years. MEMS gyroscopes are cheaper to manufacture due to their silicon structure and fewer parts. The MEMS gyroscopes have oscillating components to detect Coriolis Effect, as shown in Figure 2.6. When the gyroscope rotates, the angular velocity can be calculated by measuring the resultant Coriolis force F_c which appears orthogonal to the drive axis of the MEMS gyroscope.

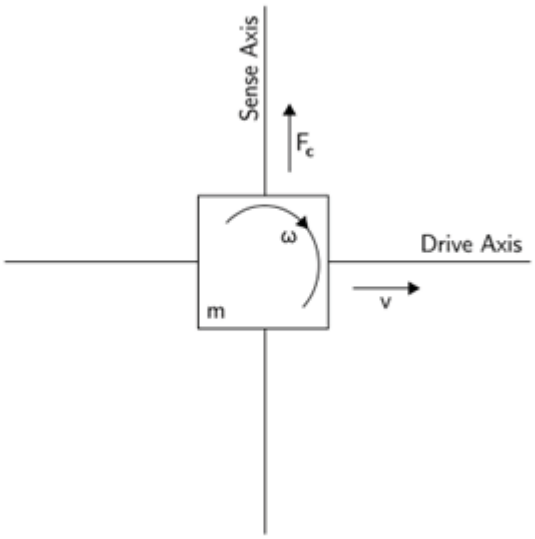


Figure 2.6. An oscillating mass gyroscope (*6*)

When a frame of reference is rotating with an angular velocity ω, a mass m moving with velocity v experiences a Coriolis force F_c as seen in equation 2.7.

$$F_c = -2\, m\, (\omega \times v) \tag{2.7}$$

The advantages described in MEMS accelerometer section are also valid for MEMS gyroscope. The main disadvantage of MEMS gyroscope is its limited accuracy which is worse than accuracy of an optical gyroscope.

2.3. MEMS Inertial Measurement Units Applications

The complex accelerometer and gyroscope structures can be fabricated in micro-size by using MEMS technology. The MEMS accelerometer and gyroscope are manufactured on silicon material, and they can be placed in a single package with low power consumption, reliable outputs and a very small sensor area. This single package structure that consists of a MEMS accelerometer and a MEMS gyroscope is named as MEMS Inertial Measurement Unit (IMU). MEMS based IMUs are constantly evolving day by day, and have a dominant role with a wide range of applications, such as automotive industry, virtual reality glasses, positioning without GPS satellites, smart telephones, stabilization of aerial vehicles, video-camera stabilization, and biomedical

applications (*9*). The MEMS IMUs have been the major parts of these applications during recent years. MEMS IMUs are selected in this thesis because of their small structure, easy communication with a microprocessor and low power consumption. It has become essential for the sensors to perform accurate measurements and to reduce the total energy consumption, size and weight.

MEMS accelerometers are one of the most widely used MEMS sensors in the market (*9*). These accelerometers are extensively used in the automotive industry, and they have the second largest market share among all MEMS sensors. The main usage areas in vehicles are safety systems, electronic suspension control, and airbag trigger circuits. On the gyroscope side, the mechanical, fiber optic, and ring laser gyroscopes are very bulky and expensive. Therefore, MEMS gyroscopes are preferred because of their small size and very advantageous price. The combination of these sensors is also used in orientation calculation or positioning applications.

Steven M. LaValle et al. (*10*) reported methods for human head tracking by using MEMS IMUs. Oculus Rift virtual reality headset monitors the head orientation of the user to present the correct image to the user's eyes, as seen in Figure 2.7

Figure 2.7. Oculus Rift virtual reality glass (*10*)

Oculus Rift uses a board as seen in Figure 2.8 that consists of STM32F103 microprocessor, Invensense MPU-6000 MEMS IMU and Honeywell HCM5983 magnetometer. STM32F103 microprocessor (red square) communicates with MEMS IMU (orange square) and magnetometer sensor (yellow square). The IMU is working at 1000 Hz sampling frequency. The IMU and magnetometer measurements are sent to the PC via USB interface. In this study, gyroscope drift is shown and compensation of this

drift with the accelerometer at the x and y axes and with the magnetometer at z-axis is described.

Figure 2.8. The sensor board of the Oculus Rift for head tracking (*10*)

Steven M. LaValle, who is a robotic professor at University of Illinois has worked with Oculus Rift virtual reality team and he had an Internet blog to describe sensor fusion algorithms (*11, 12*). In these blog posts, tracking method of the Oculus headset is explained and the challenging situations in this project are mentioned. In this study, the orientation of the human head can be calculated by using standard methods such as the Kalman filter (*13*), however, the use of the Kalman filter demands a lot of modeling assumptions and adjustments to achieve its theoretical benefits. Kalman filter is the most suitable method for linear systems and linear measurements with gaussian noise. However, the success of the Kalman filter is not certain out of these criteria. Also, Kalman filter is suitable to work at lower sampling rates. As a result, they are using the quaternions to track user's head orientation.

MEMS IMUs and magnetometers are used for position calculation without GPS satellite. The usage area of the positioning without GPS can be divided into two parts. The first usage is the uninterrupted navigation in the absence of GPS signal, and the second is to perform indoor navigation operations without using the GPS signal at all. For example, Hide and Moore (*14*) showed the performance of the GPS and IMU measurements integration. Kim and Kim in (*15*) reported how the indoor navigation system is implemented by using only MEMS IMUs. In this study, a smartphone that contains its own MEMS IMU module is used to calculate position information of a pedestrian. Step detection and step distance are estimated with the proposed scheme.

MEMS IMU modules are extensively used in smart phones. For example, the Samsung Galaxy S7 smart phone has STMicroelectronics LSM6DS3 IMU. Figure 2.9 shows the board of Samsung Galaxy S7 smart phones and green square indicates the LSM6DS3 IMU. When the user holds the phone in vertical or horizontal position, the screen orientation of the phone changes according to IMU measurements. Also, IMU modules can be used as motion detectors. For example, when the motion detectors of some gaming products such as Nintendo Wii, and Microsoft Kinects are moved, the characters in the game make similar moves. In addition, smart watches use the same IMU modules to recognize body movements, and thus, more accurately identify number of steps in walking.

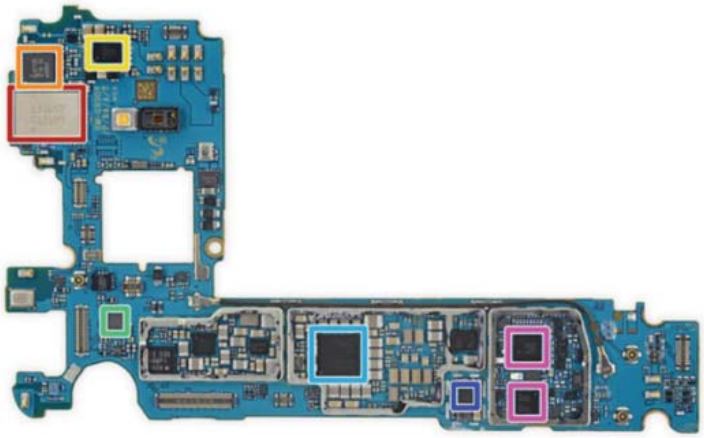


Figure 2.9. Samsung Galaxy S7 smart phone main board (*16*)

The IMU modules are used for a wide range of stabilization procedures, such as aerial vehicles, camera signal stabilization or camera gimbals. Karpenko et al. (*17*) reported about digital camera stabilization and shutter correction by using a MEMS gyroscope. In this paper, they propose an algorithm to correct rolling shutter warping and to stabilize the video signal. At the end, they use the phone's MEMS gyroscope to show how this algorithm remove the camera shake and correct rolling shutter warping in real-time. Also, MEMS IMUs are used to stabilize aerial vehicles. Orientation of the aerial vehicles are calculated by using MEMS IMU measurements and aerial vehicles are stabilized by using this orientation information. IMUs also keep the aerial vehicles stable against the wind or any other surprising turns.

In this application, a MEMS IMU is used to recognize the movement of the surgeon in three-dimensional space. The surgical endoscopic robot is directly controlled by the surgeon with this orientation information. The similar applications of endoscopic camera control with MEMS IMUs in the literature are reviewed in the following paragraphs.

Reilink et al. (*18*) reported about the endoscopic camera control by using a MEMS IMU during thoracic surgery. According to this study, endoscope device had been controlled by surgeon's assistant during surgery and this situation had several disadvantages. For instance, the surgeon and the assistant have to stand in the same area and this situation restricted the surgeon's work space. The surgeon was stressed and tired during surgery, and the camera position was not adjusted exactly the way the surgeon wants. As a solution, this study aims to control the endoscope device according to the head movement of the surgeon. An IMU is placed over the surgeon's head and IMU data are sent to the motor controller of the endoscope robot through RS232 connection. The actuator motors are controlled based on the orientation of IMU, and the surgeon controls the start and end of the endoscopy control process with a clutch. In addition, the surgeon can monitor the endoscopy image on a monitor or a display that can be attached to the surgeon head.

Lin et al. (*19*) focus on the usage of endoscopic camera during sinus surgery. In traditional sinus surgery, the endoscope and other surgical tools are directly controlled by the surgeon. In this study, the endoscope is mounted on a robot manipulator and a MEMS IMU is attached to the surgeon's foot. The surgeon controls the endoscope device with foot movements, and pan, tilt, and zoom motions of the endoscopy device are controlled according to the foot orientation. Pitch angle of the IMU is used to enable or disable the system. Roll angle is used to select one of the motors that control pan, tilt, or zoom of the endoscopy device, and yaw angle is used to control the selected motor movement. Also, detailed information about the robot manipulator which controls the endoscope device is provided in this study.

Apart from these studies, endoscopy robots are controlled with various methods in the literature such as foot pedals, joystick, voice control, and eye tracking. A detailed list of applications found in the literature related to the endoscopy robots and automated control can be found in (*20*).

CHAPTER 3

SENSOR FUSION ALGORITHMS

Throughout this chapter, coordinate frames, rotation matrices, Euler angles and quaternion representation are presented. Specifically, the information about sensor fusion algorithms which are the complementary filter, Madgwick filter and Mahony filter, is given in separate sub-sections.

3.1. Coordinate Frames

The position of an object in space is expressed by using coordinate frames. As seen in Figure 3.1, there are two coordinate frames that are used to describe the position of an object (*21*).

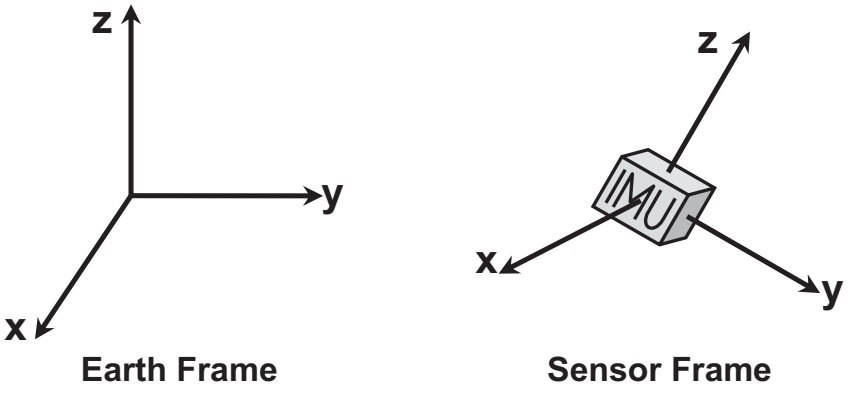


Figure 3.1. Earth frame and sensor frame

The earth frame is also called the world or inertial frame. As discussed in Chapter 2, an accelerometer senses the gravity and the dynamic motion it is exposed to. The earth frame is defined by the gravity and the magnetic north, that can be determined by the accelerometer itself and a magnetometer, respectively. The second coordinate frame is the sensor frame, which is also named as body frame. The three axes accelerometer and the three axes gyroscope in an IMU make measurements along or around their three sensitive axes in the sensor frame. Sensor frame and earth frame are related to each other,

and earth frame is used as a reference to express the position, orientation and motion of the object. Additionally, different transformation techniques are used to switch between these two frames that can be summarized as follows.

- Earth Frame: It is determined by constant forces like gravity and magnetic north. So, accelerometer or magnetometer senses forces that are stationary in the earth frame. The origin of this frame is located at a fixed point on the earth and the axes are fixed with respect to earth (*22*).
- Sensor Frame: It is the coordinate system of a moving object which is the IMU in this thesis. The origin of this frame is at the center of the IMU, and the axes are aligned with the body of the IMU (*22*).

3.2. Rotation Matrix

Rotation matrix is used to describe rotation of an object around one of its axes. Rotations around x-axis, y-axis and z-axis are expressed by $R_x(\psi)$, $R_y(\theta)$ and $R_z(\varphi)$ matrices, where, ψ, θ and φ are the Euler angles that define the rotation angle around the x, y and z axes, respectively (*23*). Accordingly, the rotation matrices in a three-dimensional coordinate system can be written as follows:

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\psi) & -sin(\psi) \\ 0 & sin(\psi) & cos(\psi) \end{bmatrix} \tag{3.1}$$

$$R_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \tag{3.2}$$

$$R_z(\varphi) = \begin{bmatrix} cos(\varphi) & -sin(\varphi) & 0 \\ sin(\varphi) & cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

A reference frame is defined for the object at the initial point, and subsequent calculations are performed according to this frame while calculating the orientation of an object. For instance, frame A is rotated 90° around z-axis as seen in Figure 3.2, and the rotation result is obtained as frame B, where both of them are configurations of the body frame.

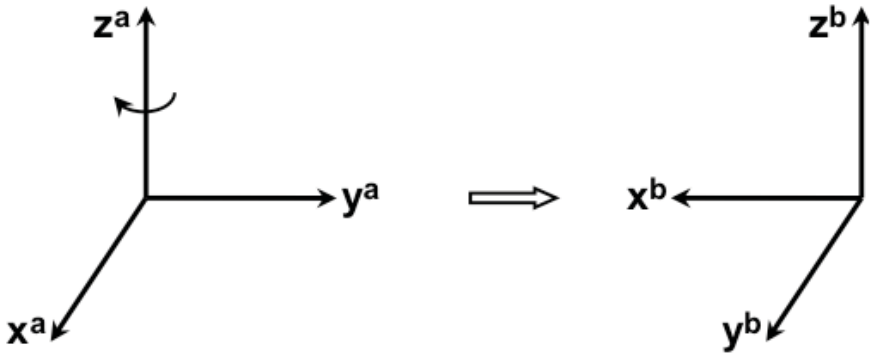


Figure 3.2. The rotation around frame A

If the p_a vector is equal to (1, 0, 0) in frame A that represents both body frame and earth frame initially, and this vector's frame is rotated 90°around z-axis, then the p_b vector in the rotated frame B is calculated as follows.

$$p_b = R_z(\varphi)\, p_a \tag{3.4}$$

$$p_b = \begin{bmatrix} cos\,(90°) & -sin\,(90°) & 0 \\ sin\,(90°) & cos(90°) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{3.5}$$

As another example, if the rotation is made around the x-axis by ψ angle, then the y^a-axis moves to the y^b-axis and the z^a-axis moves to the z^b-axis. Since the rotation is made around x^a-axis, the orientation of x^a-axis and x^b-axis remains the same and this rotation is calculated as seen in equation 3.6.

$$p_b = R_x(\psi)\, p_a \tag{3.6}$$

Afterwards, if a second rotation is made around the new y^b-axis by θ angle in frame B, then as a result of this rotation, the x^b-axis moves to the x^c-axis position and the z^b-axis moves to z^c-axis position. The orientation of y^b-axis and y^c-axis is equal to each other. The vector p_c in the new frame C is calculated as follows.

$$p_c = R_y(\theta)\, p_b \tag{3.7}$$

Lastly, a rotational movement is performed around z^c-axis in frame C by an angle φ, and the x^c-axis moves to x^d-axis position and the y^c-axis moves to the y^d-axis position. The orientation of z^c-axis and z^d-axis is equal to each other. The final orientation of the vector is represented by using x^d, y^d, and z^d axes in frame D.

$$p_d = R_z(\varphi)\, p_c \tag{3.8}$$

The frames from A to D are configurations of the body frame (*21; 22*). The all three rotations from frame A to frame D can be expressed with a single transformation matrix as $R_{z,y,x}(\psi, \theta, \varphi)$, and the final vector p_d is calculated as follows.

$$p_d = R_z(\varphi)\, R_y(\theta)\, R_x(\psi)\, p_a \tag{3.9}$$

$$R_{z,y,x}(\psi, \theta, \varphi) = R_z(\varphi)\, R_y(\theta)\, R_x(\psi) \tag{3.10}$$

$$p_d = R_{z,y,x}(\psi, \theta, \varphi)\, p_a \tag{3.11}$$

3.3. Euler Angles

Any rotation can be represented by three angles according to Euler's rotation theorem. These three angles are named as roll, pitch, and yaw angles in aviation. Roll refers to a rotation around nose to tail axis of the aircraft with ψ angle around the x-axis. Pitch is a rotation around wing to wing axis with θ angle around the y-axis, and yaw is rotation of the nose towards left or right with φ angle around the z-axis (*21*).

$$R_{z,y,x}(\psi, \theta, \varphi) = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{3.12}$$

$$R_{z,y,x}(\psi, \theta, \varphi) = \begin{bmatrix} c\theta c\varphi & s\psi s\theta c\varphi - c\psi s\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ c\theta s\varphi & s\psi s\theta s\varphi + c\psi c\varphi & c\psi s\theta s\varphi - s\psi c\varphi \\ -s\theta & s\psi c\theta & c\psi c\theta \end{bmatrix} \tag{3.13}$$

These three angles can be calculated by using the rotation matrix that is defined in equation 3.12 and 3.13 (*23*). In equation 3.13, the letters c and s refer to cosine and sine functions, respectively.

To determine the orientation of an object in earth frame, the roll, pitch, and yaw angles are calculated using equations 3.14, 3.15 and 3.16 respectively.

$$\psi = tan^{-1}\left(\frac{R_{32}}{R_{33}}\right) \tag{3.14}$$

$$\theta = -sin^{-1}(R_{31}) \tag{3.15}$$

$$\varphi = tan^{-1}\left(\frac{R_{21}}{R_{11}}\right) \tag{3.16}$$

However, $sin^{-1}(x)$ and $tan^{-1}(y/x)$ functions are defined between -90° and +90° and it is not feasible to calculate all possible rotations with these functions. For example, if x and y values are smaller than zero, then both of the $tan^{-1}\left(\frac{-1}{-1}\right)$ and $tan^{-1}\left(\frac{+1}{+1}\right)$ produce the same result, and there is no unique solution for these cases. This problem can be solved by replacing $tan^{-1}(y/x)$ function with $atan2(y,x)$ function that is defined between -180° and +180°. Many programming languages support the usage of function $atan2(y,x)$ (*23*) and it is defined as follows.

$$atan2(y,x) = \begin{cases} tan^{-1}\left(\frac{y}{x}\right) & if\ x > 0 \\ tan^{-1}\left(\frac{y}{x}\right) + \pi & if\ x < 0\ and\ y \geq 0 \\ tan^{-1}\left(\frac{y}{x}\right) - \pi & if\ x < 0\ and\ y < 0 \\ \frac{\pi}{2} & if\ x = 0\ and\ y > 0 \\ -\frac{\pi}{2} & if\ x = 0\ and\ y < 0 \\ undifened & if\ x = 0\ and\ y = 0 \end{cases} \tag{3.17}$$

As a result, the roll, pitch, and yaw angles are calculated with $atan2(y,x)$ function by using equations 3.18, 3.19 and 3.20.

$$\psi = atan2(R_{32}, R_{33}) \tag{3.18}$$

$$\theta = -sin^{-1}(R_{31}) \tag{3.19}$$

$$\varphi = atan2(R_{21}, R_{11}) \tag{3.20}$$

Different angular rotations are represented by various rotational combinations. The complete rotational motion is obtained by a multiplication of rotation matrices that are obtained for each axis as explained in section 3.2. However, If the pitch angle θ is equal to ±90°, then the Euler angles cannot be calculated from the rotation matrix, because R_{11}, R_{21}, R_{32} and R_{33} will all be equal to zero. As a result, it will not be possible to distinguish angular positions, such as (90°, 90°, 0°), (0°, 90°, 90°), and (180°, 90°, 90°), from each other under this condition. This phenomenon is named as *Gimbal lock* (*22, 23*), and quaternions can be used to prevent this singularity problem.

3.3.1. Angular Position Tracking with Accelerometer

The accelerometer senses linear acceleration and earth's gravity. The gravity is utilized to calculate rotation angles around x and y axis. The angular position around z-axis cannot be calculated by using an accelerometer, because the measured gravity does not provide any information for this axis.

If a 3-axis accelerometer is used, roll ψ and pitch θ angles can be calculated by using sampled accelerometer measurements $a = \left[a_x, a_y, a_z\right]^T$, as seen in equation 3.21 and equation 3.22.

$$\psi[k] = arctan\left(\frac{a_y[k]}{\sqrt{a_x^2[k] + a_z^2[k]}}\right) \tag{3.21}$$

$$\theta[k] = arctan\left(\frac{a_x[k]}{\sqrt{a_y^2[k] + a_z^2[k]}}\right) \tag{3.22}$$

There, a_x, a_y and a_z refer to the measurements made by the accelerometer on x, y and z axes and k indicates the sample index of the accelerometer measurements. Additionally, roll and pitch angles together are named as inclination (*24*).

The data collected from the accelerometer of MPU6050 at ±2 g measurement range setting and 100 Hz sampling rate are used to evaluate angle calculation. The accelerometer is calibrated during data collection. The figures given below are the results generated by MATLAB using the collected data.

A mechanism was designed to evaluate IMU response to as small as one-degree angular changes. This mechanism consists of a platform where the IMU can be mounted on, and two arms that control orientation of the platform around the x and y axes, as seen in Figure 3.3. Two pairs of screws mounted at 114 mm distance from the pivot point of the platform set the limits of angular motion. The 3 mm metric screws have 0.5 mm thread pitch, and the resultant rotation step corresponding to one turn of the screws is 0.25° = asin(0.5/114). The range of the platform's movement is controlled by adjusting these screws. It is possible to check how fast the IMU responds to small step changes and its tracking accuracy after application of several steps by using this mechanism.

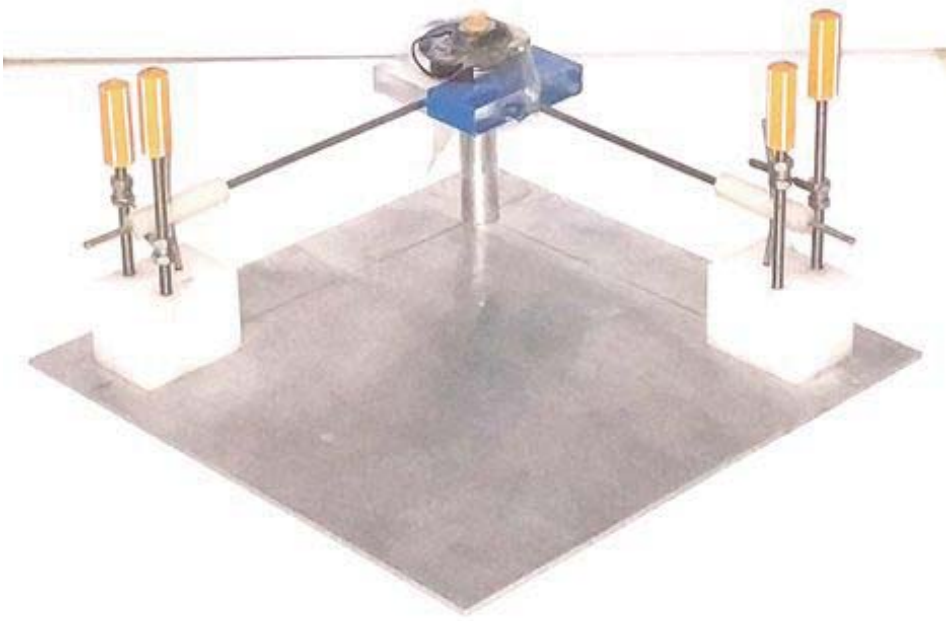


Figure 3.3. The IMU testing platform

In Figure 3.4, the angles of x and y axes are calculated by using the accelerometer data in stationary condition. Range of fluctuations were approximately 0.25° in pitch and roll angles. As seen in Figure 3.5, one-degree rotation can be detected with sufficient resolution by using the accelerometer. The large spikes that are seen during the step changes correspond to the short vibrations caused by the platform positioning arms hitting the limiting screws of the test mechanism and these spikes should be filtered out to prevent errors in a control process.

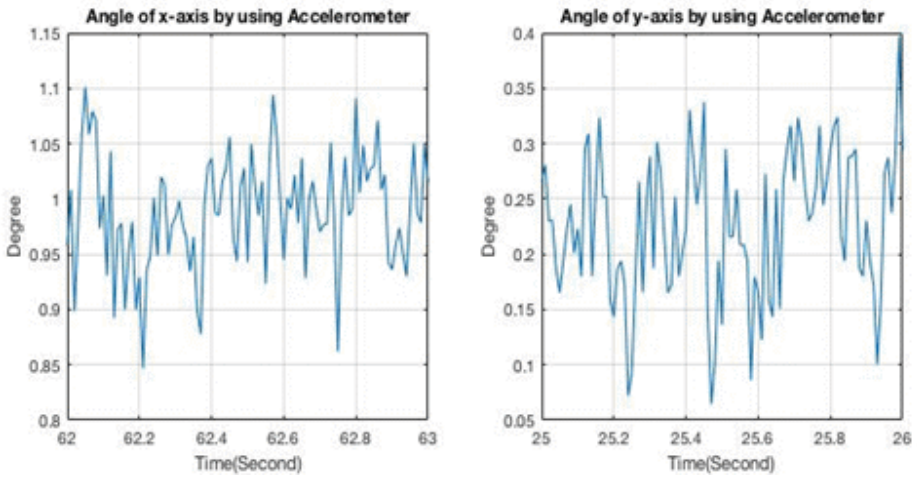


Figure 3.4. Angle calculation by using accelerometer at stationary condition

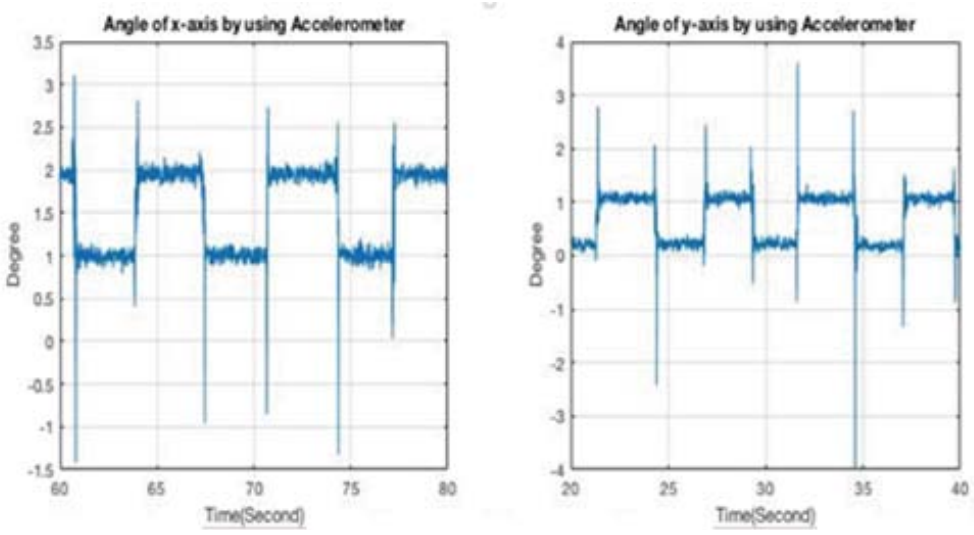


Figure 3.5. Angle calculation by using accelerometer in motion

3.3.2. Angular Position Tracking with Gyroscope

Gyroscope senses angular velocity in degree per second, and its output is equal to the time derivative of angular position $\phi = [\psi, \theta, \varphi]^T$. In this experiment, roll ψ, pitch θ, and yaw φ angles were calculated simply by integrating the angular velocity $\omega = [\omega_x, \omega_y, \omega_z]^T$ measured by the gyroscope, since the changes in IMU position are restricted to rotation around a single axis.

$$\omega(t) = \frac{\partial \phi(t)}{\partial t} \tag{3.23}$$

The discrete time form of the equation 3.23 is represented as seen in equation 3.24, and it is approximated by a summation of discrete gyroscope samples $\omega[k]$ multiplied by the sampling time interval Δt.

$$\phi(t) = \int_0^t \omega(t)dt \approx \sum_{k=1}^{Ns} \omega[k]\,\Delta t \tag{3.24}$$

where, k is the sample index and $N_s \approx t/\Delta t$ is the number of samples acquired during the integration period. The rotation around any of the three axes is given by

$$\psi[k] = \psi[k-1] + \omega_x[k]\,\Delta t \tag{3.25}$$

$$\theta[k] = \theta[k-1] + \omega_y[k]\,\Delta t \tag{3.26}$$

$$\varphi[k] = \varphi[k-1] + \omega_z[k]\,\Delta t \tag{3.27}$$

where, ω_x, ω_y and ω_z represent the output signal of the gyroscope around x, y and z-axis, and ψ, θ, and φ refer to roll, pitch and yaw angles, respectively. Δt represents the elapsed time between two measurements given by inverse of the output data rate of the gyroscope.

The equations 3.25, 3.26 and 3.27 give the final angular position when the IMU motion is restricted to rotation around a single axis of the sensor frame. In case of a combined rotation around multiple axes, the sampled gyroscope outputs in the sensor frame should be transformed by using rotation matrices or quaternions according to the current IMU orientation at every step.

In Figure 3.6, the angles of x and y axes are calculated by using the gyroscope measurements in stationary condition. The data were collected from gyroscope of MPU6050 at ±2000 dps measurement range and 100 Hz sampling rate. The gyroscope is calibrated, and the collected data were processed by MATLAB program. The angle results are almost noiseless compared to the accelerometer readings. The expected angular position is 0° in all axes, because there is no angular velocity that affects the gyroscope. However, there was nearly 0.5° shift on x-axis and nearly 1.4° shift on y-axis

after 150 seconds due to gyroscopic drift.

As seen in Figure 3.7, one degree of change can be detected by using the gyroscope with much better precision compared to the accelerometer. There are no spikes in the angle measurements compared to the accelerometer while the gyroscope is in motion. However, the calculated angular position around x and y axes shift while the gyroscope is at stationary state.

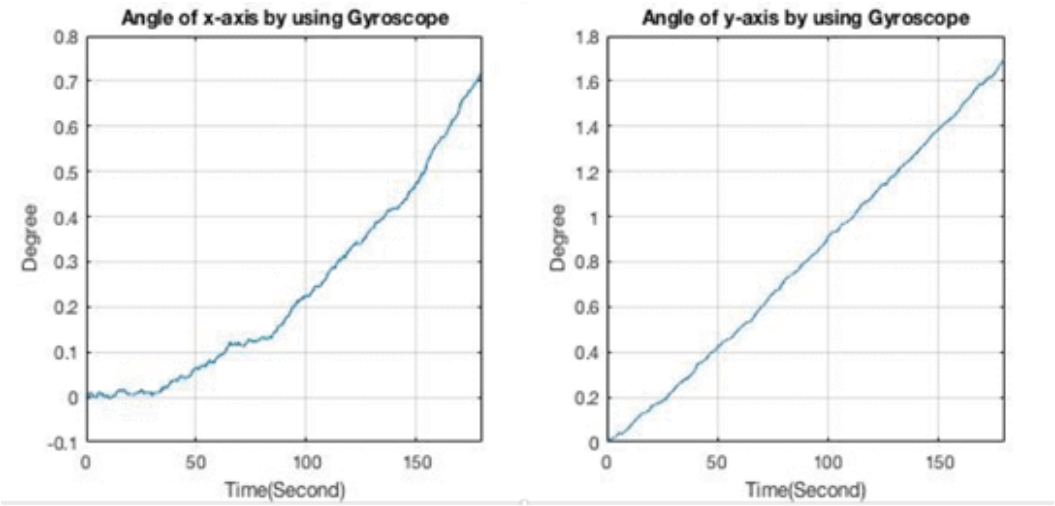


Figure 3.6. Angle calculation by using gyroscope in stationary condition

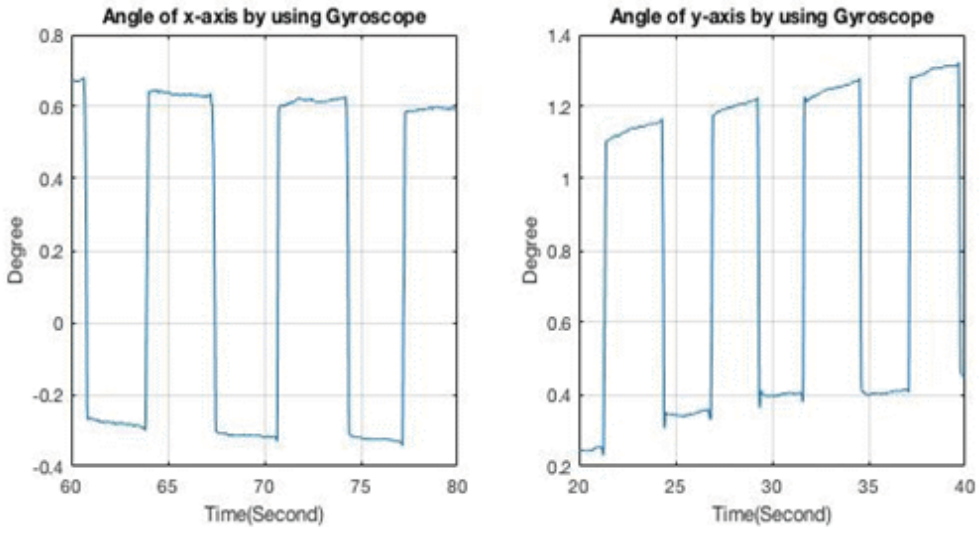


Figure 3.7. Angle calculation by using gyroscope in motion

3.3.3. Complementary Filter

The complementary filter is used to combine accelerometer and gyroscope measurements as seen in Figure 3.8 (*25*). The purpose of this filter is to eliminate the negative aspects of the accelerometer and the gyroscope. The gyroscope gives a slowly changing angle of rotation in stationary condition because of the zero-rate output level and its temperature dependency. In this case, the angular positon calculated by gyroscope drifts continuously. The other problem is that, the gyroscope cannot estimate the initial position by itself. On the other hand, the accelerometer has noisy output due to the nature of the accelerometer, and the measurements are not reliable because of accelerometer sensitivity to vibration. As far as advantages are concerned, the gyroscope is reliable while monitoring fast motion, and the accelerometer can be used to calculate the reference angle values based on gravity. As a result, both of the sensors have deficiencies when they are used as standalone devices, but accurate angular calculations can be made by combining their advantages.

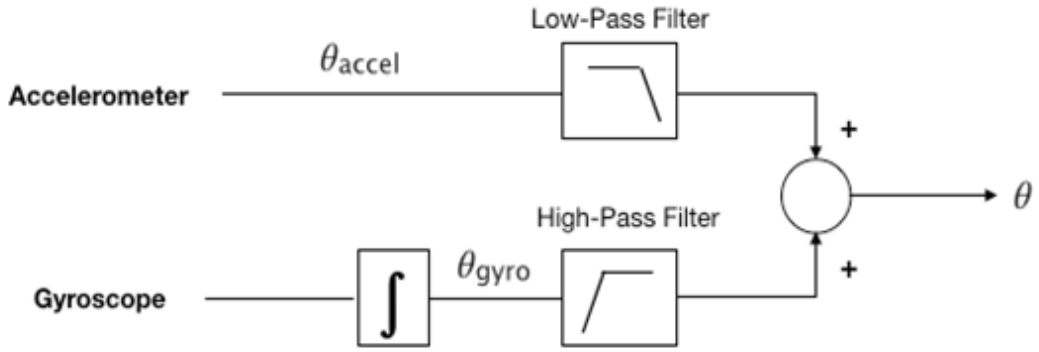


Figure 3.8. The complementary filter

The accelerometer output must be filtered with a low pass filter as seen in Figure 3.8, because the accelerometer is reliable in stationary position. The gyroscope output, on the other hand, must be filtered with a high pass filter because the gyroscope makes measurements more efficiently during motion.

$$\psi[k] = (a)\,(\,\psi[k-1] \;+\; \omega_x[k]\,\Delta t\,) \;+\; (1-a)\,(\,\psi_{accel}[k]\,) \qquad (3.28)$$

$$\theta[k] = (a)\left(\theta[k-1] + \omega_y[k]\,\Delta t\right) + (1-a)\left(\theta_{accel}[k]\right) \quad (3.29)$$

$$a = \frac{\tau}{\tau + \Delta t} \quad (3.30)$$

In equation 3.28 and 3.29, ω_x and ω_y represent output signal of the gyroscope around x and y axis respectively. Δt represents the sampling time of the gyroscope and k is the sample index of the accelerometer and gyroscope measurements. ψ_{accel} and θ_{accel} refer to roll and pitch angles, respectively, that are calculated according to the accelerometer output. The determination of the filter constant a is an important task. If the filter constant increases, then the noise in angle calculation is reduced, however, the response time for calculation of the initial reference position will be longer. If the filter constant decreases, then the initial response time will be shorter, but the calculated angle will be more sensitive to noise in the accelerometer output. Filter constant a can be calculated as seen in equation 3.30, where, τ refers to the time constant of the complementary filter.

As seen in Figure 3.9, the noise and spikes caused by the accelerometer and the angular shift due to the gyroscope are reduced by using a complementary filter. The time constant of the complementary filter is 0.1 seconds. As a result, the filter coefficient a is equal to 0.909 with 100 Hz output data rate of the accelerometer and gyroscope.

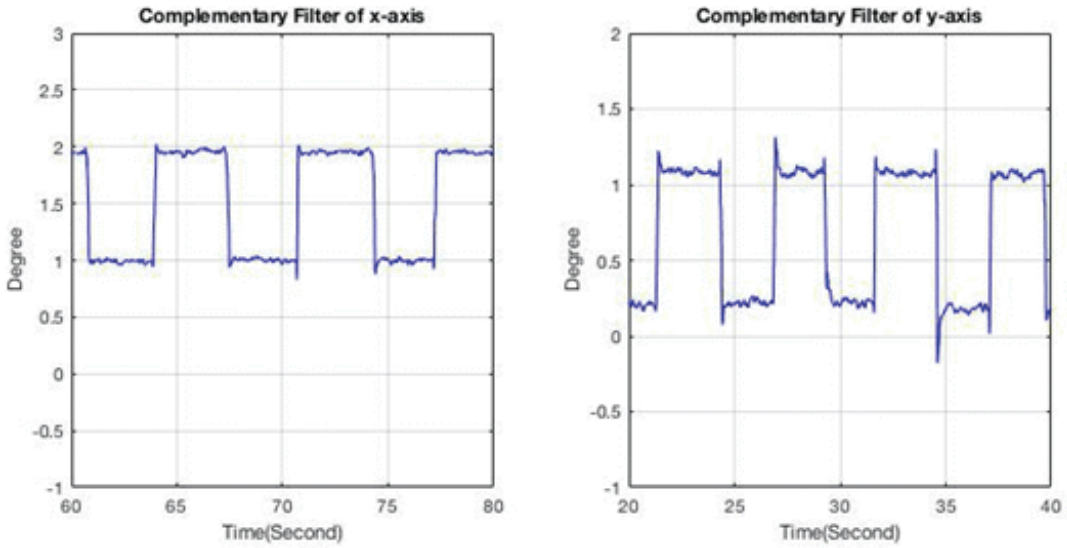


Figure 3.9. Angle calculation by using complementary filter

As seen in Figure 3.10, results obtained with the complementary filter contain less noise compared to the angle calculation with the accelerometer alone. One degree of angular motion can be detected with sufficient accuracy by using the complementary filter. As seen in Figure 3.11, the dynamic response of the complementary filter is almost same as the output of the gyroscope. The advantage of the complementary filter is that, the gyroscopic drift is eliminated and the initial position of the IMU is obtained from the accelerometer measurements.

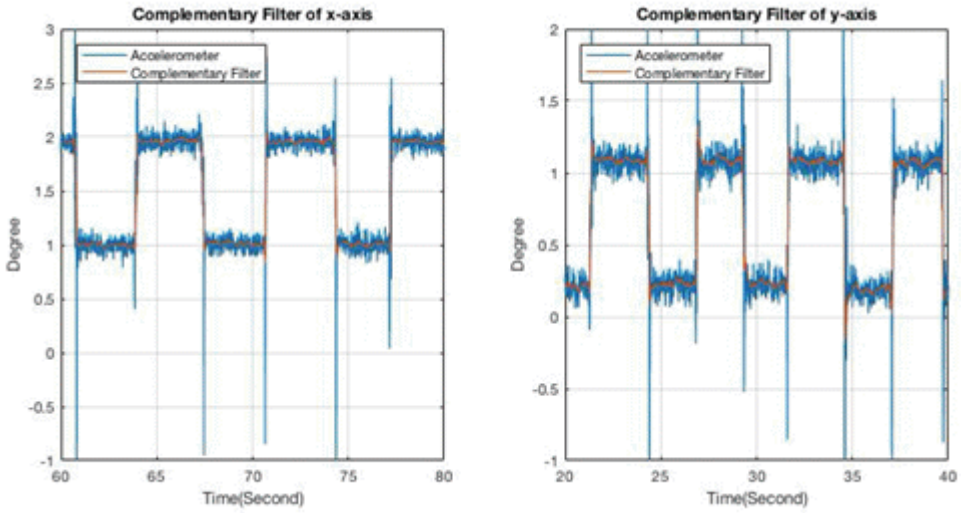


Figure 3.10. Angle calculation by using accelerometer and complementary filter

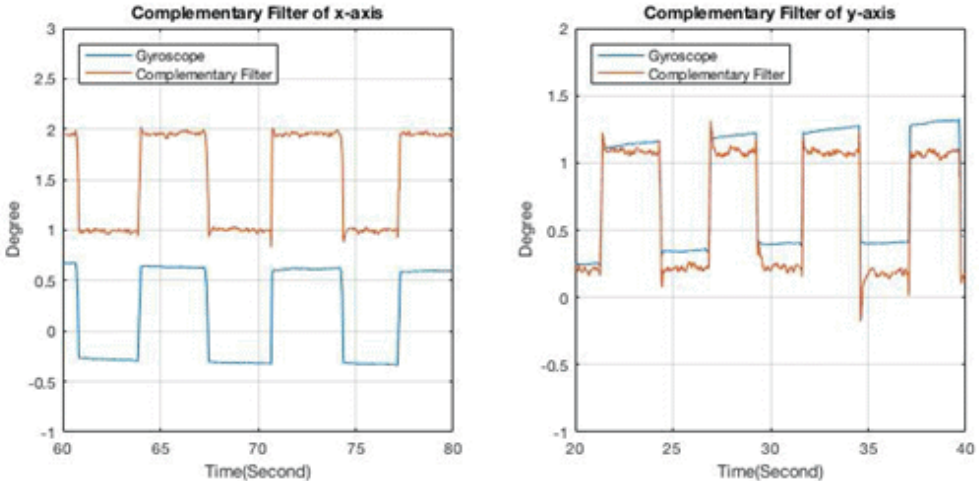


Figure 3.11. Angle calculation by using gyroscope and complementary filter

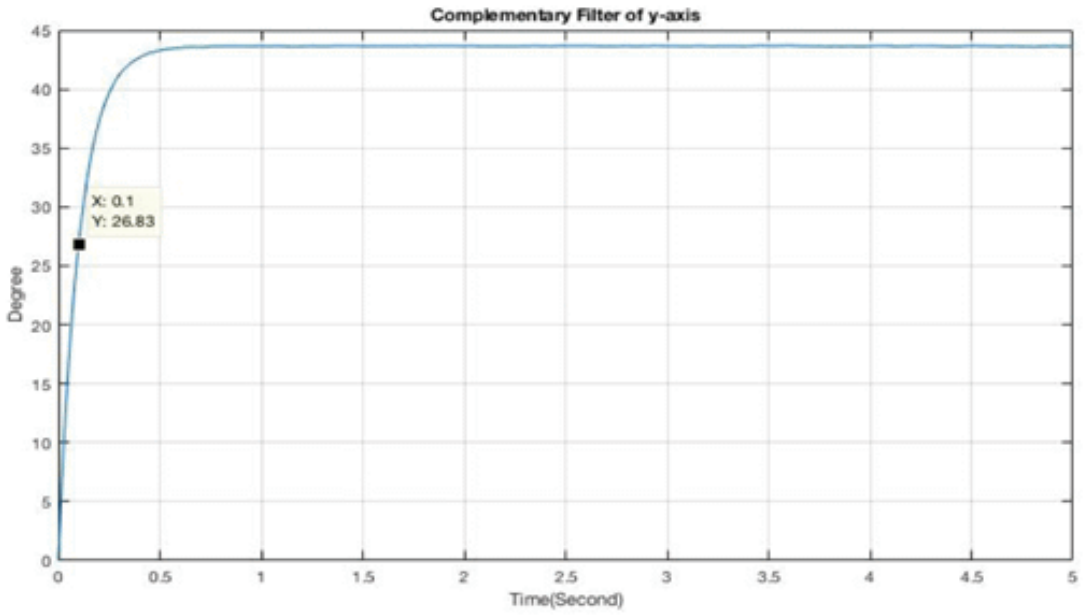


Figure 3.12. Calculation of initial position by using complementary filter (τ=0.1 s)

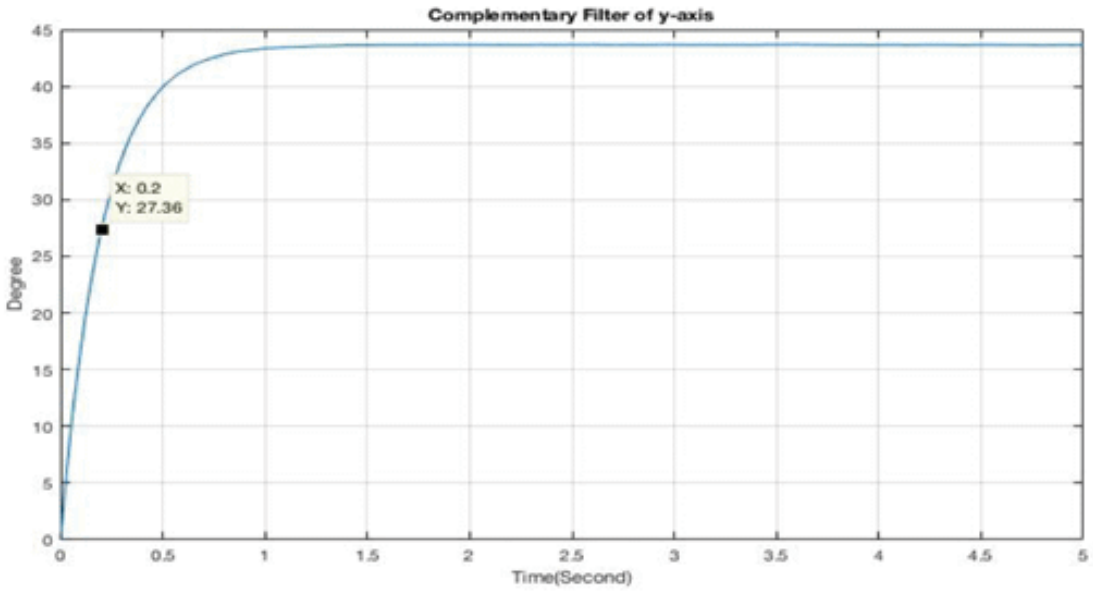


Figure 3.13. Calculation of initial position by using complementary filter (τ=0.2 s)

Figure 3.12 shows how long it takes for the complementary filter to find the initial position of IMU. As it is known, the output of the filter reaches 63.2% $(1 - e^{-1})$ of the full response when time is equal to the time constant. The initial position of y-axis angle is equal to 43.87° and the value obtained for y-axis is equal to 26.83° at 0.1 seconds. Thus, the requirement of the filter has been verified for the complementary filter with 0.1

second time constant. Figure 3.13 shows a similar complementary filter response when the time constant is equal to 0.2 seconds. The starting point of y-axis is equal to 43.87° and the filter output for y-axis is equal to 27.36° at 0.2 seconds.

3.4. Quaternion

Quaternions have been used as an alternative method for expressing the orientation of an object in three-dimensional space since their introduction by William Roman Hamilton in 1843 (*26*). A quaternion q is defined as the combination of a scalar part $q_0 \in \mathrm{R}$, and a vector part $v(q_1\ q_2\ q_3) \in R^3$. Three-unit vectors $\vec{\imath} = (1,0,0)$, $\vec{j} = (0,1,0)$ and $\vec{\mathrm{k}} = (0,0,1)$, cover the standard orthonormal basis for R^3. As a result, $q = [\ q_0\ q_1\ q_2\ q_3]^T$ is defined in R^4 as follows.

$$q = s + \vec{v} = q_0 + q_1\vec{\imath} + q_2\vec{j} + q_3\vec{k} = [\ q_0\ q_1\ q_2\ q_3]^T \qquad q \in R^4 \tag{3.31}$$

There are some basic mathematical rules while working with quaternions (*26, 27*). The following rules must be used while multiplying the vector parts of the quaternions.

$$i^2 = j^2 = k^2 = ijk = -1 \tag{3.32}$$

$$ij = -ji = k \tag{3.33}$$

$$jk = -kj = i \tag{3.34}$$

$$ki = -ik = j \tag{3.35}$$

The sum of two quaternions q and q' is calculated as follows.

$$q + q' = (q_0 + q_0') + (q_1 + q_1')\vec{\imath} + (q_2 + q_2')\vec{j} + (q_3 + q_3')\vec{k} \tag{3.36}$$

The multiplication of the quaternions q and q' is calculated as follows.

$$q \otimes q' = [\ q_0\ q_1\ q_2\ q_3]^T \otimes [q_o'\ q_1'\ q_2'\ q_3']^T \tag{3.37}$$

$$q \otimes q' = \begin{bmatrix} q_0 q'_o - q_1 q'_1 - q_2 q'_2 - q_3 q'_3 \\ q_0 q'_1 + q_1 q'_o + q_2 q'_3 - q_3 q'_2 \\ q_0 q'_2 - q_1 q'_3 + q_2 q'_o + q_3 q'_1 \\ q_0 q'_3 + q_1 q'_2 - q_2 q'_1 + q_3 q'_o \end{bmatrix} \tag{3.38}$$

The multiplication of quaternion q with scaler r is calculated as follows.

$$rq = (r\ q_0) + (r\ q_1)\vec{\imath} + (r\ q_2)\vec{\jmath} + (r\ q_3)\vec{k} \tag{3.39}$$

q^* is named as the conjugate of q and it can be calculated as seen below,

$$\begin{aligned} q &= q_0 + q_1\vec{\imath} + q_2\vec{\jmath} + q_3\vec{k} = [\,q_0\ q_1\ q_2\ q_3]^T \\ q^* &= q_0 - q_1\vec{\imath} - q_2\vec{\jmath} - q_3\vec{k} = [\,q_0\ -q_1\ -q_2\ -q_3]^T \end{aligned} \tag{3.40}$$

The norm of q is equal to $\|q\|$ and it is defined as,

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{3.41}$$

The inverse of q is defined as;

$$q^{-1} = \frac{q^*}{\|q\|^2} \tag{3.42}$$

The unit quaternion $\hat{q}$ is defined as seen in equation 3.43 and the norm of the unit quaternion is equal to one ($\|\hat{q}\| = 1$).

$$\hat{q} = \frac{q}{\|q\|} \tag{3.43}$$

A quaternion is a four-dimensional complex number given by q_0, q_1, q_2 and q_3, where q_0 refers to the rotation around the quaternion vector and q_1, q_2 and q_3 are x, y, and z axes components in three-dimensional coordinate system, respectively.

Orientation of an object can be described by using quaternion in three-dimensional coordinate system. Consider two vectors v^A and v^B that are the same vectors expressed in the frames A and B, with the components x_A, y_A, z_A and x_B, y_B, z_B, respectively. The representation ${}^A_B q$ expresses the orientation of an object in frame B with respect to frame A (*28*).

$$ {}^A_B q = {}^B_A q^* \tag{3.44} $$

$$ v^B = {}^A_B q \otimes v^A \otimes {}^A_B q^* \tag{3.45} $$

The rotation matrix R can be obtained by using the elements of a unit quaternion as seen in equation 3.46.

$$ R = \begin{bmatrix} 2q_0^2 - 1 + 2q_1^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & 2q_0^2 - 1 + 2q_2^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 - 2q_0q_1 & 2q_0^2 - 1 + 2q_3^2 \end{bmatrix} \tag{3.46} $$

The Euler angles are calculated by using quaternion representation as seen in equations 3.47, 3.48 and 3.49.

$$ \psi = atan2(2q_2q_3 - 2q_0q_1, 2q_0^2 - 1 + 2q_3^2) \tag{3.47} $$

$$ \theta = -sin^{-1}(2q_1q_3 + 2q_0q_2) \tag{3.48} $$

$$ \varphi = atan2(2q_1q_2 - 2q_0q_3, 2q_0^2 - 1 + 2q_1^2) \tag{3.49} $$

Also, the quaternions can be calculated by using Euler angles as seen in equations 3.50, 3.51, 3.52 and 3.53.

$$ q_0 = cos\left(\frac{\varphi}{2}\right) cos\left(\frac{\psi}{2}\right) cos\left(\frac{\theta}{2}\right) - sin\left(\frac{\varphi}{2}\right) sin\left(\frac{\psi}{2}\right) sin\left(\frac{\theta}{2}\right) \tag{3.50} $$

$$ q_1 = sin\left(\frac{\varphi}{2}\right) sin\left(\frac{\psi}{2}\right) cos\left(\frac{\theta}{2}\right) + cos\left(\frac{\varphi}{2}\right) cos\left(\frac{\psi}{2}\right) sin\left(\frac{\theta}{2}\right) \tag{3.51} $$

$$ q_2 = cos\left(\frac{\varphi}{2}\right) sin\left(\frac{\psi}{2}\right) cos\left(\frac{\theta}{2}\right) - sin\left(\frac{\varphi}{2}\right) cos\left(\frac{\psi}{2}\right) sin\left(\frac{\theta}{2}\right) \tag{3.52} $$

$$ q_3 = sin\left(\frac{\varphi}{2}\right) cos\left(\frac{\psi}{2}\right) cos\left(\frac{\theta}{2}\right) + cos\left(\frac{\varphi}{2}\right) sin\left(\frac{\psi}{2}\right) sin\left(\frac{\theta}{2}\right) \tag{3.53} $$

3.4.1. Quaternion Representation for Gyroscope Output

The angular velocities measured by a gyroscope around x, y, and z axes are shown as ω_x, ω_y and ω_z respectively, in equation 3.54. The unit of these variables is radian per second (rad/s) for quaternion calculations. The vector ω^s refers to the measurements of gyroscope in the sensor frame.

$$\omega^s = \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \end{bmatrix}^T \tag{3.54}$$

The rate of change of the sensor frame relative to the earth frame can be represented by using derivative of the quaternion ${}^S_E q(t)$ as in equation 3.55.

$$\frac{\partial {}^S_E q(t)}{\partial t} = \frac{1}{2} {}^S_E \hat{q}(t - \Delta t) \otimes \omega^s(t) \tag{3.55}$$

The orientation of an object can be calculated by integrating the quaternion derivative.

$${}^S_E q(t) = \int \left(\frac{1}{2} {}^S_E \hat{q}(t - \Delta t) \otimes \omega^s(t) \right) dt \tag{3.56}$$

The discrete time form of the equation 3.56 is given as equation 3.57, where $\omega^s[k]$ is described as sampled gyroscope measurements at sample index k, and Δt is the sampling time. ${}^S_E \hat{q}[k-1]$ refers to the previous estimation of the quaternion value, and ${}^S_E q[0]$ is the initial condition assumed to be $[1\ 0\ 0\ 0]^T$.

$${}^S_E q[k] = {}^S_E \hat{q}[k-1] + \left(\frac{1}{2} {}^S_E \hat{q}[k-1] \otimes \omega^s[k] \right) \Delta t \tag{3.57}$$

$$\begin{bmatrix} q_{0,k} \\ q_{1,k} \\ q_{2,k} \\ q_{3,k} \end{bmatrix} = \begin{bmatrix} \hat{q}_{0,k-1} \\ \hat{q}_{1,k-1} \\ \hat{q}_{2,k-1} \\ \hat{q}_{3,k-1} \end{bmatrix} + \frac{1}{2} \left(\begin{bmatrix} \hat{q}_{0,k-1} \\ \hat{q}_{1,k-1} \\ \hat{q}_{2,k-1} \\ \hat{q}_{3,k-1} \end{bmatrix} \otimes \begin{bmatrix} 0 \\ \omega_{x,k} \\ \omega_{y,k} \\ \omega_{z,k} \end{bmatrix} \right) \Delta t \tag{3.58}$$

$$\begin{bmatrix} q_{0,k} \\ q_{1,k} \\ q_{2,k} \\ q_{3,k} \end{bmatrix} = \begin{bmatrix} \hat{q}_{0,k-1} \\ \hat{q}_{1,k-1} \\ \hat{q}_{2,k-1} \\ \hat{q}_{3,k-1} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} -\hat{q}_{1,k-1}\omega_{x,k} - \hat{q}_{2,k-1}\omega_{y,k} - \hat{q}_{3,k-1}\omega_{z,k} \\ \hat{q}_{0,k-1}\omega_{x,k} + \hat{q}_{2,k-1}\omega_{z,k} - \hat{q}_{3,k-1}\omega_{y,k} \\ \hat{q}_{0,k-1}\omega_{y,k} - \hat{q}_{1,k-1}\omega_{z,k} + \hat{q}_{3,k-1}\omega_{x,k} \\ \hat{q}_{0,k-1}\omega_{z,k} - \hat{q}_{1,k-1}\omega_{y,k} + \hat{q}_{2,k-1}\omega_{x,k} \end{bmatrix} \Delta t \qquad (3.59)$$

As a result, the orientation of an object is described by unit quaternion, as seen in equation 3.60.

$$\overset{S}{E}\hat{q}[k] = \frac{\overset{S}{E}q[k]}{\| \overset{S}{E}q[k] \|} \qquad (3.60)$$

In Figure 3.14, the angles of x and y axes are calculated by using only the gyroscope data at stationary state. The blue and red lines refer to the previously calculated Euler angles and the quaternion representation respectively. There was almost the same shift around x and y-axes for both Euler angle and quaternion calculations while the IMU is not in motion. As seen in Figure 3.15, angular changes smaller than one degree can be detected with gyroscope by quaternion representation, and the results of Euler angle and quaternion representation are the same.

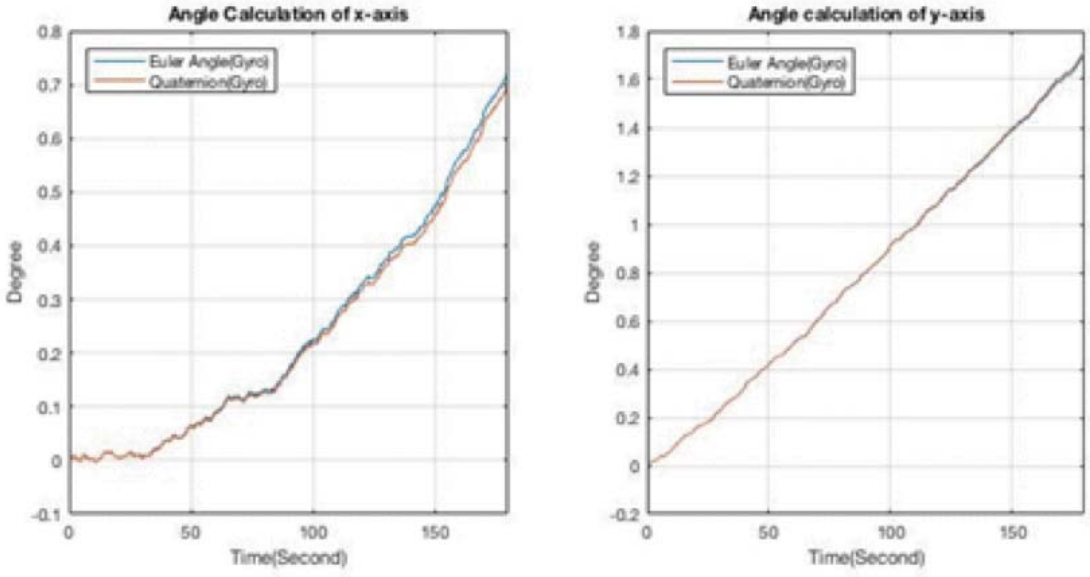


Figure 3.14. Angle calculation by using gyroscope at stationary condition

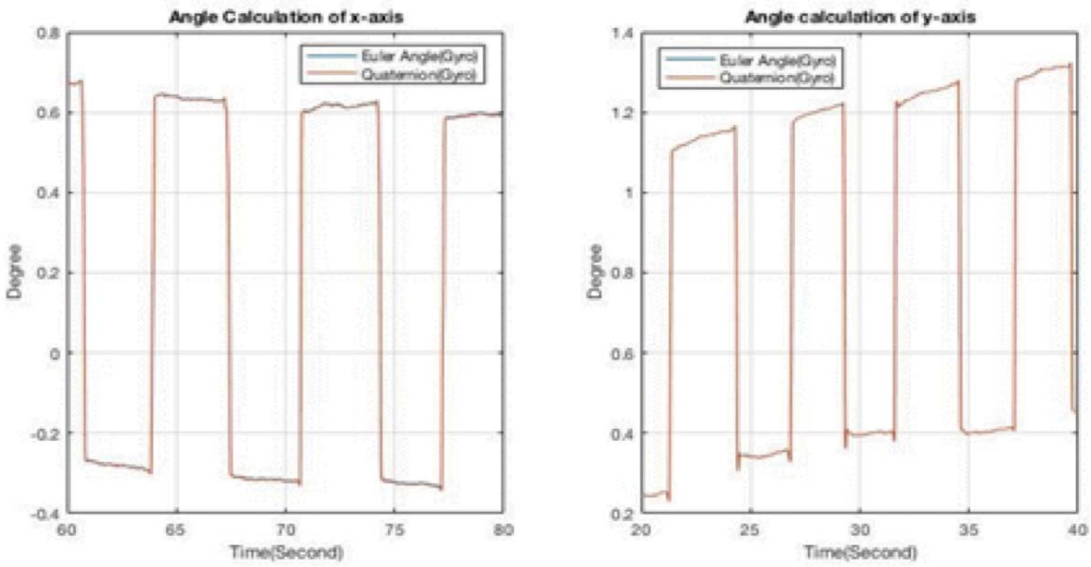


Figure 3.15. Angle calculation by using gyroscope in motion

3.4.2. Madgwick Filter

Madgwick filter was developed by Sebastian O. H. Madgwick as part of his doctoral study (*28, 29*). As mentioned before, when the orientation of an object is calculated by using gyroscope only, the gyroscopic drift may cause large deviations in the orientation of an object in all three axes. An accelerometer and a magnetometer were used in the Madgwick filter to correct the gyroscopic drift. An accelerometer is used to correct the drifts around x and y axes, and for outdoor applications, a magnetometer that senses magnetic north is used to correct the gyroscopic drift around z-axis. However, magnetometer measurements are not reliable in operating room conditions because of ferromagnetic building materials and other objects. In the scope of this thesis, the most practical solution is to accept the present positon of the IMU as the z-axis reference after every activation of the user interface unit. Further information on how to include magnetometer measurements in Madgwick filter can be found in (*28, 29*).

If a vector is known in the earth frame, it can be mapped to the sensor frame knowing that the gravity is in z direction and its amplitude is 1 g. The gravity vector in the earth frame is defined as g^E in equation 3.61.

$$g^E = [0\ 0\ 0\ 1]^T \tag{3.61}$$

The gravity vector can be mapped to the sensor frame by using equation 3.62.

$$g^s = {}^E_S q \otimes g^E \otimes {}^E_S q^* \tag{3.62}$$

$$g^s = {}^S_E q^* \otimes g^E \otimes {}^S_E q \tag{3.63}$$

$$g^s = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{3.64}$$

$$g^s = \begin{bmatrix} 0 \\ 2(q_1 q_3 - q_0 q_2) \\ 2(q_0 q_1 + q_2 q_3) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{3.65}$$

Accordingly, the sensor frame gravity vector g^s can be calculated as in equation 3.66.

$$g^s = \begin{bmatrix} 0 \\ 2(q_1 q_3 - q_0 q_2) \\ 2(q_0 q_1 + q_2 q_3) \\ 1 - 2q_1^2 - 2q_2^2 \end{bmatrix} \tag{3.66}$$

Accelerometer measures the gravity in the sensor frame, and the resultant measurements are represented as a^s. If the a^s vector is subtracted from g^s, then the function $f({}^S_E q,\ a^s)$ is found:

$$f({}^S_E q,\ a^s) = g^s - a^s = \begin{bmatrix} 0 \\ 2(q_1 q_3 - q_0 q_2) \\ 2(q_0 q_1 + q_2 q_3) \\ \left(1 - 2q_1^2 - 2q_2^2\right) \end{bmatrix} - \begin{bmatrix} 0 \\ a_x \\ a_y \\ a_z \end{bmatrix} \tag{3.67}$$

The function $f({}^S_E q,\ a^s)$ is reorganized as in equation 3.68.

$$f({}^S_E q,\ a^s) = \begin{bmatrix} 0 \\ 2(q_1 q_3 - q_0 q_2) - a_x \\ 2(q_0 q_1 + q_2 q_3) - a_y \\ (1 - 2q_1^2 - 2q_2^2) - a_z \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix} \tag{3.68}$$

If $f(\,{}^{S}_{E}q,\ a^{s})$ function is minimized, the gyroscopic drift in the x and y axes will be eliminated. The gradient descent algorithm is used to find the minimum of $f(\,{}^{S}_{E}q,\ a^{s})$.

$$ {}^{S}_{E}q[k] = {}^{S}_{E}\hat{q}[k-1] - \beta \frac{\nabla f\left({}^{S}_{E}\hat{q}[k-1], a^{s}[k]\right)}{\left\|\nabla f({}^{S}_{E}\hat{q}[k-1], a^{s}[k])\right\|} \qquad k = 1,2,3,\ldots. \tag{3.69} $$

If the gradient descent algorithm is repeated k times with step size β and the first estimated direction of ${}^{S}_{E}q[0]$ is known, then ${}^{S}_{E}q[k]$ is found by using equation 3.69. The gradient of the $f\left({}^{S}_{E}\hat{q}[k-1], a^{s}[k]\right)$ function is calculated as in equation 3.70.

$$ \nabla f({}^{S}_{E}\hat{q}[k-1], a^{s}[k]) \ = J({}^{S}_{E}\hat{q}[k-1])^{T} f({}^{S}_{E}\hat{q}[k-1], a^{s}[k]) \tag{3.70} $$

where, $J({}^{S}_{E}\hat{q}[k-1])$ refers to the Jacobian matrix that can be calculated as follows.

$$ J({}^{S}_{E}\hat{q}[k-1]) = \begin{bmatrix} \frac{\partial f_0}{\partial q_0} & \frac{\partial f_0}{\partial q_1} & \frac{\partial f_0}{\partial q_2} & \frac{\partial f_0}{\partial q_3} \\ \frac{\partial f_1}{\partial q_0} & \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial q_3} \\ \frac{\partial f_2}{\partial q_0} & \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_2}{\partial q_3} \\ \frac{\partial f_3}{\partial q_0} & \frac{\partial f_3}{\partial q_1} & \frac{\partial f_3}{\partial q_2} & \frac{\partial f_3}{\partial q_3} \end{bmatrix} \tag{3.71} $$

$$ J({}^{S}_{E}\hat{q}[k-1]) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -2q_2 & 2q_3 & -2q_0 & 2q_1 \\ 2q_1 & 2q_0 & 2q_3 & 2q_2 \\ 0 & -4q_1 & -4q_2 & 0 \end{bmatrix} \tag{3.72} $$

In equation 3.73, the estimated quaternion at any sample index k is obtained as ${}^{S}_{E}q[k]$. The previous estimated quaternion value is ${}^{S}_{E}\hat{q}[k-1]$ and the initial quaternion value is ${}^{S}_{E}q[0] = [1\ 0\ 0\ 0]^{T}$. The derivative of $f({}^{S}_{E}\hat{q}[k-1], a^{s}[k])$ is calculated by $\nabla f({}^{S}_{E}\hat{q}[k-1], a^{s}[k])$.

$$ {}^{S}_{E}q[k] = {}^{S}_{E}\hat{q}[k-1] + \left(\frac{1}{2}\,{}^{S}_{E}\hat{q}[k-1] \otimes \omega^{s}[k] - \beta \frac{\nabla f({}^{S}_{E}\hat{q}[k-1], a^{s}[k])}{\left\|\nabla f({}^{S}_{E}\hat{q}[k-1], a^{s}[k])\right\|}\right)\Delta t \tag{3.73} $$

where, β and Δt are the step size and the sampling period, respectively. As a result, the orientation of an object is described by a unit quaternion given by equation 3.74.

$$ {}^{S}_{E}\hat{q}[k] = \frac{{}^{S}_{E}q[k]}{\|{}^{S}_{E}q[k]\|} \tag{3.74} $$

Figure 3.16 shows the effect of the β parameter on the error in x and y axes calculations performed on the data collected while the IMU is stationary. The minimum RMS error is obtained when β is equal to 0.03 for x and y axes. As seen in Figure 3.17, spikes in the angle measurements caused by accelerometer and the shift due to gyroscope have been eliminated by using Madgwick filter with β = 0.03. Figure 3.18 compares the results of Madgwick filter with the angle measurements obtained by using the accelerometer alone.

The effect of wrong β value selection on angle calculations at the output of the Madgwick filter can be seen in Figure 3.19. The orientation estimation obtained when β is equal to 0.4 is noisier than the result obtained when β is equal to 0.03.

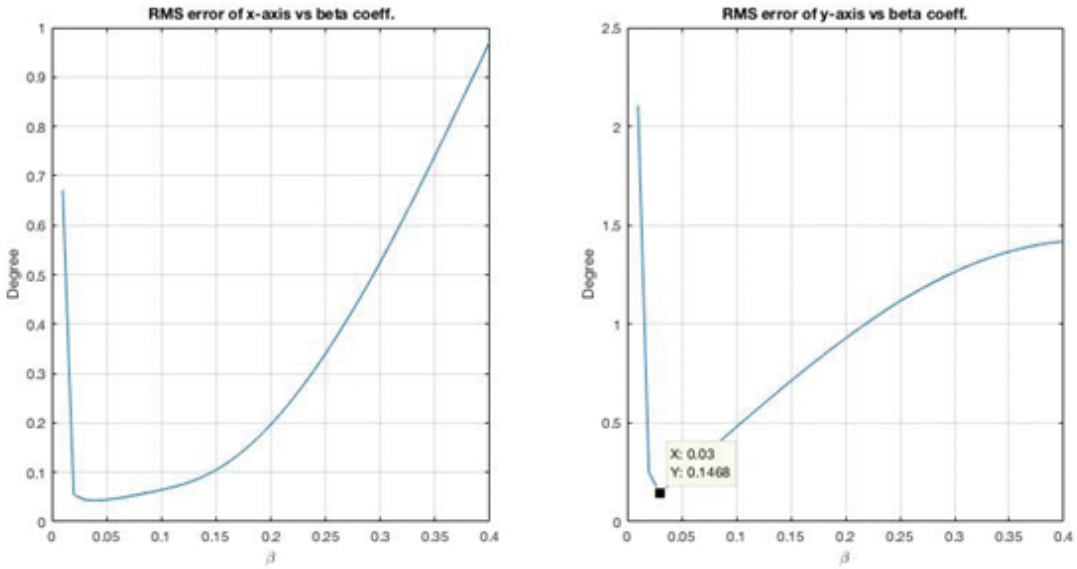


Figure 3.16. The effect of β parameter

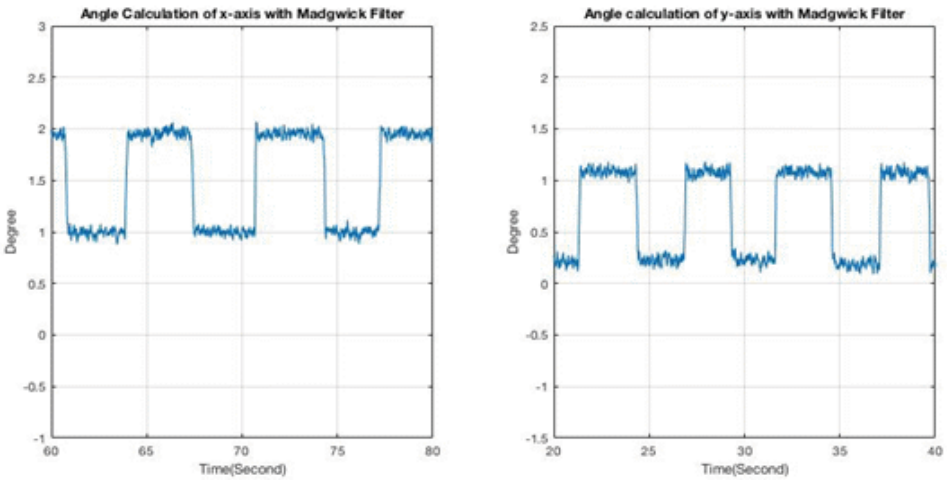


Figure 3.17. Angle calculation by using madgwick filter ($\beta = 0.03$)

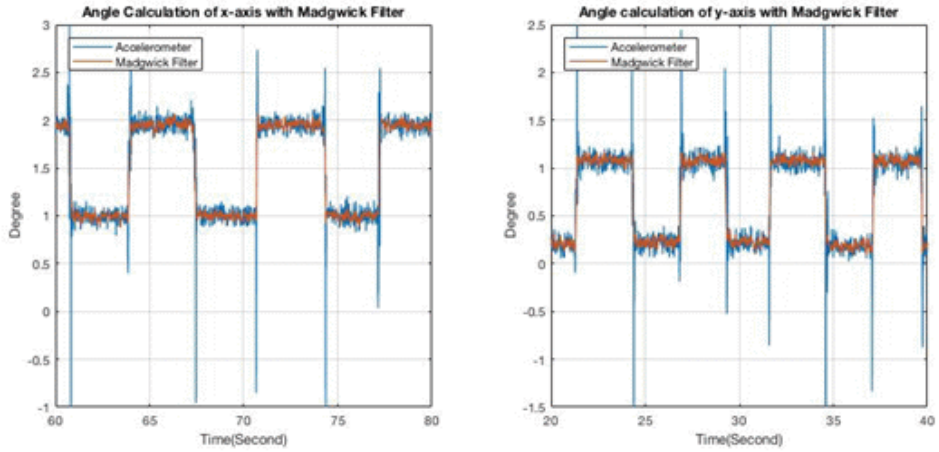


Figure 3.18. Angle calculation by using madgwick filter ($\beta = 0.03$) and accelerometer

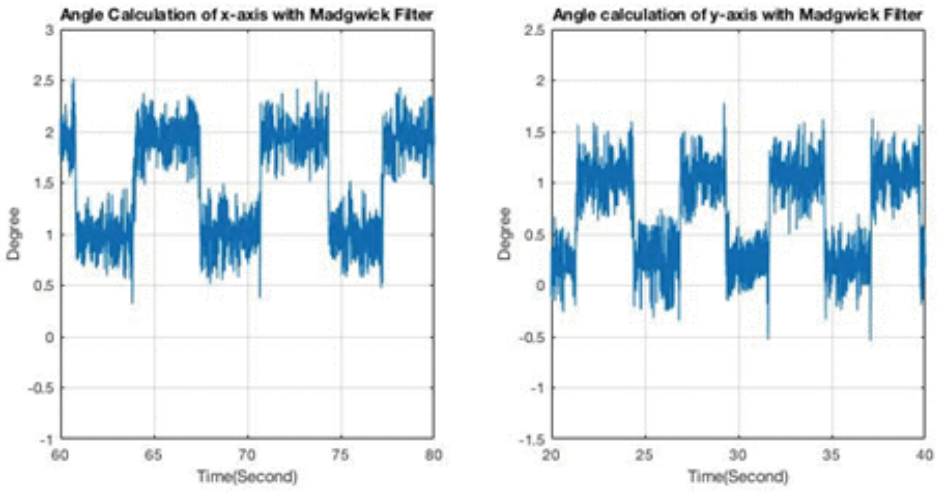


Figure 3.19. Angle calculation by using the Madgwick filter for $\beta = 0.4$

In another test, the IMU was held at nearly 45 degrees angle with respect to the x and y axes of the earth frame, and the time required for the Madgwick filter to reach the initial position is measured. Figure 3.20 shows the Madgwick filter response for $\beta = 0.03$, where the elapsed time for finding initial orientation of IMU is 13.72 seconds. Figure 3.21 shows the result of a similar test when $\beta = 0.4$. In this case, the elapsed time to find initial orientation is 1.02 seconds.

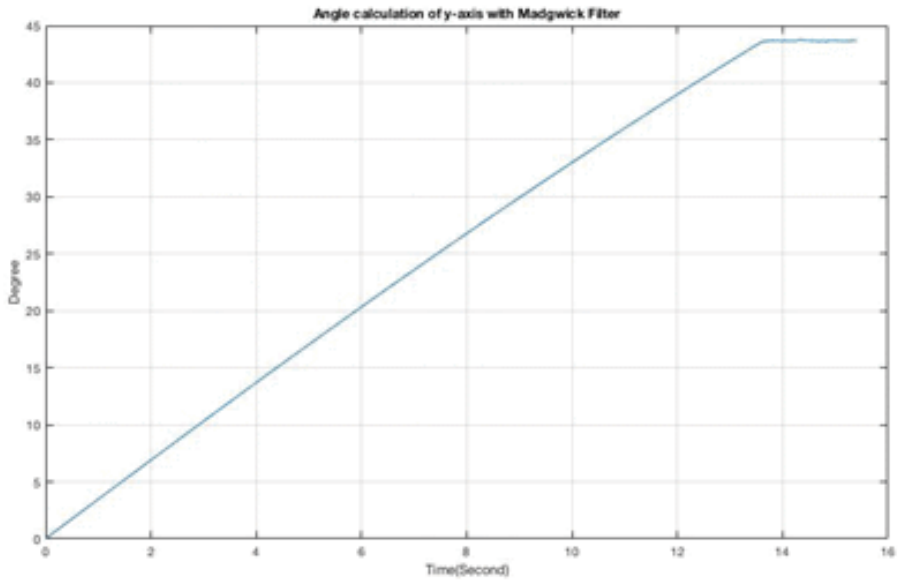


Figure 3.20. Initial position calculation by using Madgwick filter for $\beta = 0.03$

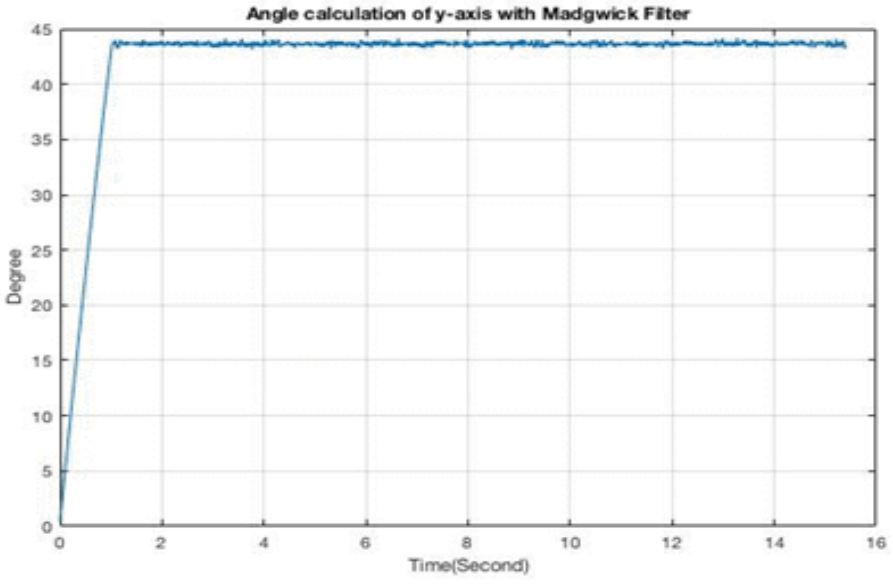


Figure 3.21. Angle calculation by using Madgwick filter for β = 0.4

It takes a shorter time to find the initial position with a larger β setting, but the filter output becomes noisy. Output of the Madgwick filter is as noisy as the accelerometer measurements when β is set to 0.1 as seen in Figure 3.22. The response time is 4.12 seconds with this β setting as seen in Figure 3.23.

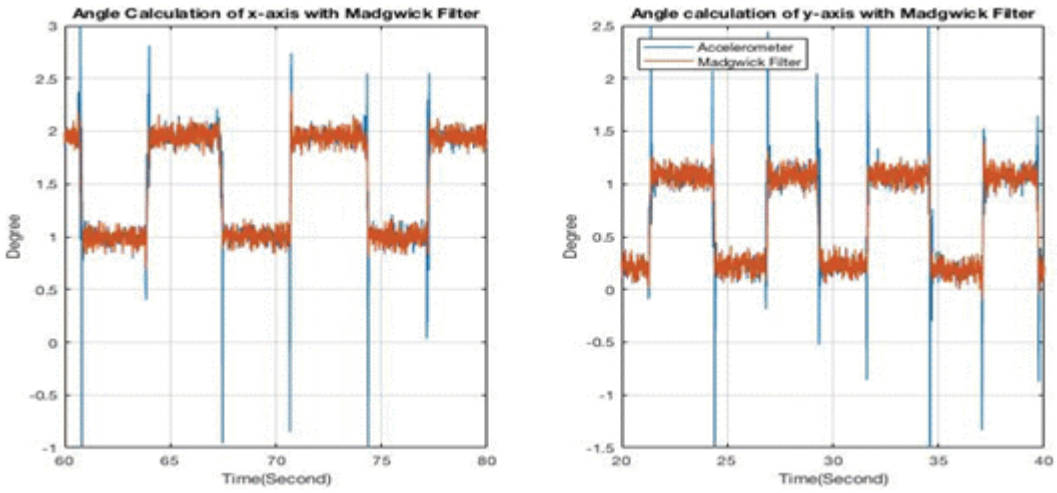


Figure 3.22. Angle calculation by using madgwick filter (β = 0.1) and accelerometer

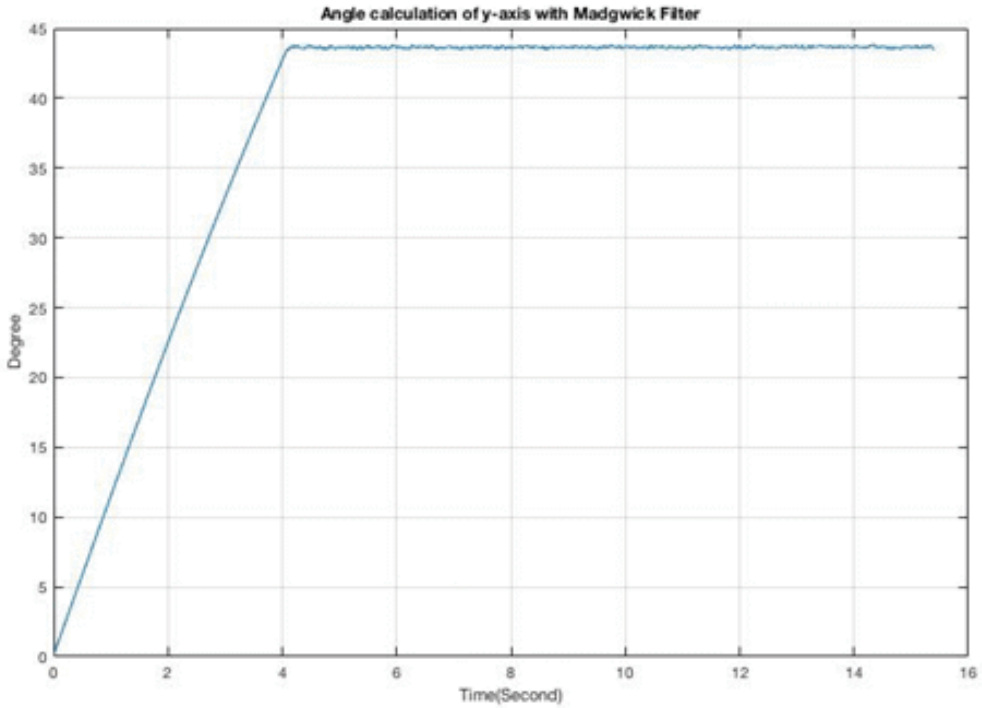


Figure 3.23. Angle calculation by using Madgwick filter (β = 0.1)

3.4.3. Mahony Filter

Mahony filter is a modified version of the complementary filter (*30*). Mahony approach corrects the gyroscope measurements of ω by using a correction vector e. A PI controller is used to eliminate the gyroscopic drift error in this filter (*31*). Firstly, the estimated gravity vector is calculated, as in equation 3.75.

$$g^s = \begin{bmatrix} 0 \\ 2(q_1q_3 - q_0q_2) \\ 2(q_0q_1 + q_2q_3) \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{3.75}$$

Secondly, an error vector (i.e. correction vector) is calculated by using cross product of the accelerometer measurements and the estimated gravity vector.

$$e = a^s \otimes g^s \tag{3.76}$$

$$e = \begin{bmatrix} 0 \\ a_x \\ a_y \\ a_z \end{bmatrix} \otimes \begin{bmatrix} 0 \\ g_x \\ g_y \\ g_z \end{bmatrix} \tag{3.77}$$

As a result, the correction of the gyroscope measurement can be done as follows.

$$\omega_c^s = \omega^s + K_P\, e + K_I \int e\ dt \tag{3.78}$$

where, K_P and K_I are the PI filter proportional and integral gain coefficients, respectively. The orientation of the object is calculated by using corrected gyroscope measurements ω_c^s as in equation 3.79 and 3.80. The quaternion ${}^S_E q[0]$ in the initial condition is equal to $[1\ 0\ 0\ 0]^T$.

$${}^S_E q[k] = {}^S_E \hat{q}[k-1] + \left(\frac{1}{2}\, {}^S_E \hat{q}[k-1] \otimes \omega_c^s[k]\right) \Delta t \tag{3.79}$$

$${}^S_E \hat{q}[k] = \frac{{}^S_E q[k]}{\| {}^S_E q[k] \|} \tag{3.80}$$

In Figure 3.24, the collected data is evaluated by using Mahony filter with the PI filter gain coefficients K_P equal to 1.4 and K_I equal to 0.002. As seen in Figure 3.25, the output of the Mahony filter is less noisy compared to the Madgwick filter with β equal to 0.03. The elapsed time to find the initial IMU position with less than 1º error is approximately 3 seconds with Mahony filter as seen in Figure 3.26.

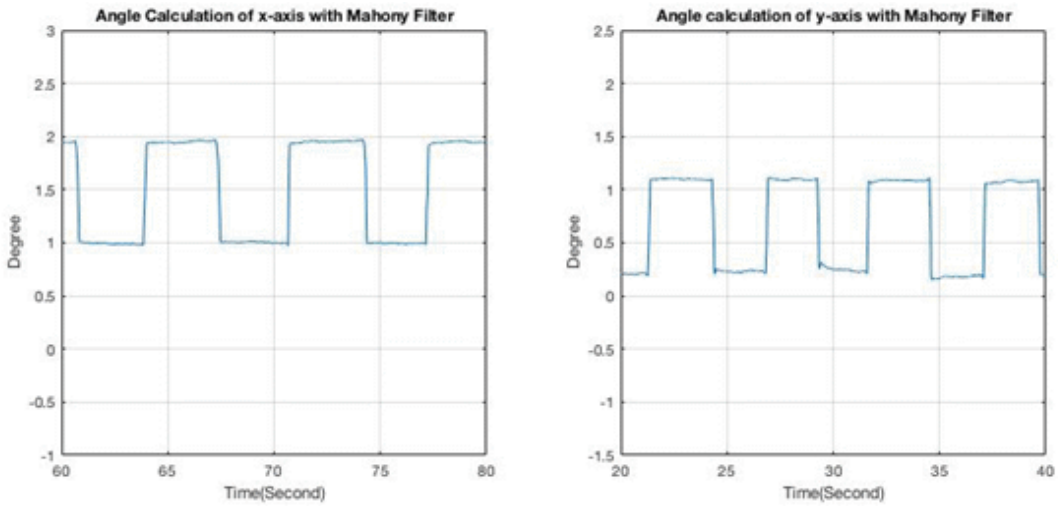


Figure 3.24. The output of the Mahony filter

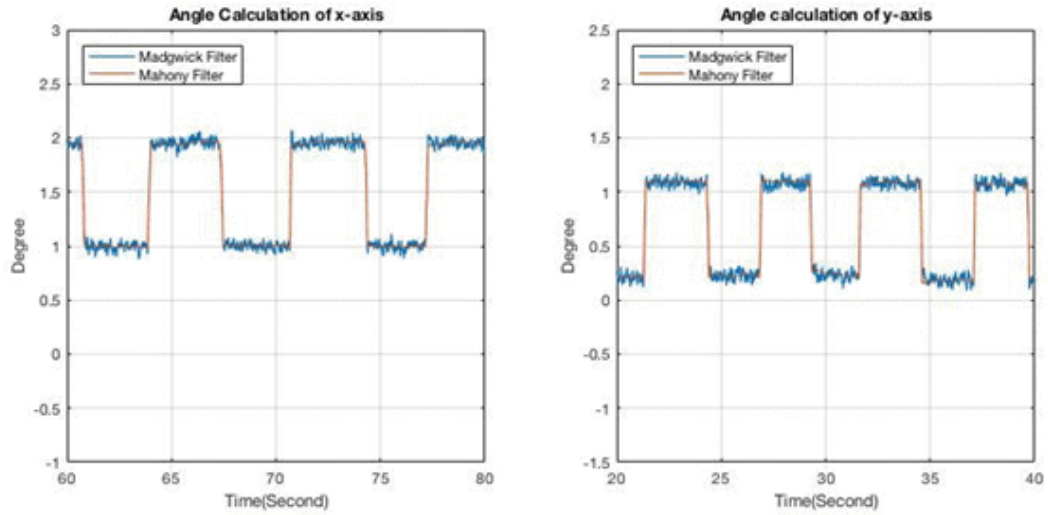


Figure 3.25. The comparison of the Madgwick filter and Mahony filter

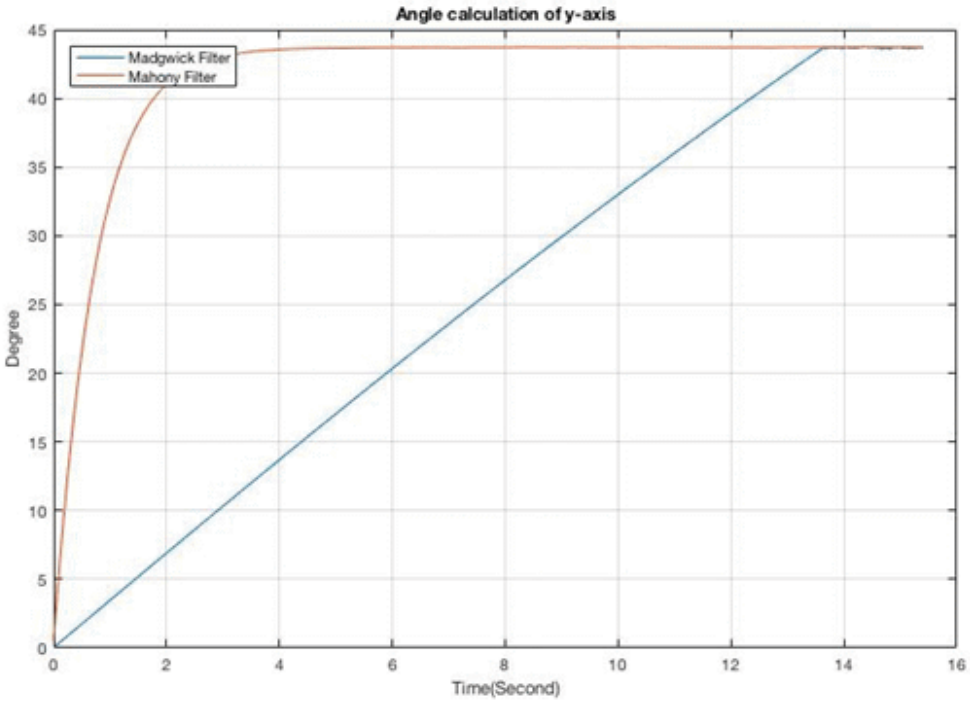


Figure 3.26. The comparison of the Madgwick filter and Mahony filter

3.5. Linear Acceleration

As mentioned earlier, accelerometer can be used to measure both gravity and linear acceleration. Speed and position calculations based on linear acceleration requires elimination of the effect of gravity. If the effect of gravity is subtracted from the accelerometer measurements, then the linear acceleration can be calculated as follows.

$$a^L = a^s - g^s = \begin{bmatrix} 0 \\ a_x - 2(q_1 q_3 - q_0 q_2) \\ a_y - 2(q_0 q_1 + q_2 q_3) \\ a_z - (1 - 2q_1^2 - 2q_2^2) \end{bmatrix}^T \tag{3.81}$$

CHAPTER 4

EVALUATION HARDWARE

The hardware components used in this study consists of the IMU modules for evaluation and the interface boards that transfer the acquired IMU data to the evaluation software running on a personal computer. All components were purchased as pre-assembled boards that are available in the market, and the necessary firmware codes were written to establish the wireless interface. Two different hardware configurations were arranged by using these interface boards according to the requirements of experiments.

Main features of the interface boards are described in this section. Also, an overview of Bluetooth Low Energy (BLE) wireless technology and the important steps of the communication protocol are given.

4.1. Hardware Setup for IMU Evaluation

The first configuration shown in Figure 4.1 uses a wired link between the interface board that is STM32F4 Discovery Development Board (DDB) and the IMU modules in order to support evaluation of a variety of IMU modules. STM32F4 DDB communicates with the IMU modules through I2C interface and HC-05 Bluetooth 2.0 module transfers the received IMU data to the evaluation software running on the PC through Bluetooth communication. The evaluation software receives the incoming IMU data and stores them for further processing. The collected data was used for evaluation of LSM6DS3, MPU6050 and BMX055 IMUs, as seen in Figure 4.2, and the results of evaluation are presented in Chapter 5.

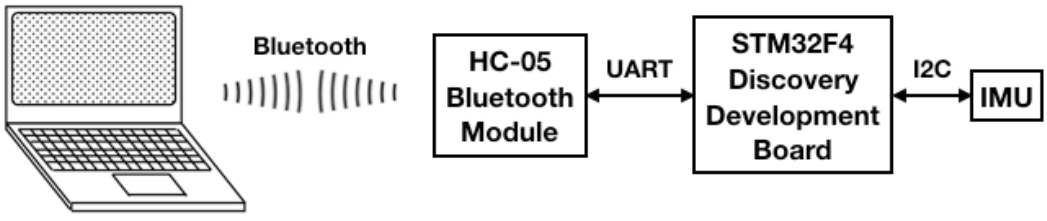


Figure 4.1. Hardware configuration for evaluation of IMUs

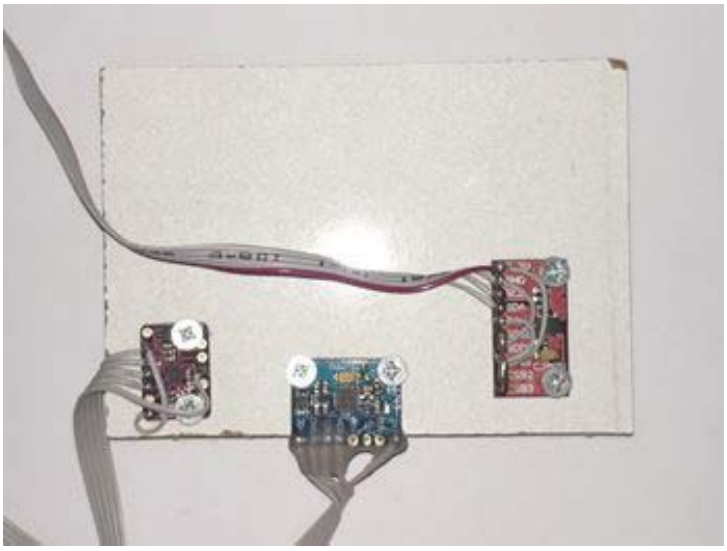


Figure 4.2. IMU modules used for evaluation

The interface board for evaluation of the IMUs is STM32F4 Discovery Development Board (DDB) shown in Figure 4.3, that contains a STM32F407VGT6 microcontroller manufactured by STMicroelectronics. STM32F407VGT6 is based on ARM Cortex-M4 32-bit processor with the maximum clock frequency of 168 MHz, and this microcontroller is highly preferred for digital signal processing projects due to its powerful processor.

Figure 4.3. STM32F4 Discovery Development Board and HC-05 Bluetooth module

4.2. Hardware Setup for Simulation

The second configuration in Figure 4.4 uses a low-power motion tracking card (MTC) that contains its own wireless interface as well as the IMU device. This configuration is used for testing motion mapping methods in a simulation environment since the MTC can easily operate on battery power for long periods of time.

The MTC that uses NRF51822QFAA microprocessor from Nordic semiconductor is selected for testing motion mapping methods due to its low power consumption. The MTC tracks the user's motion by using MPU6050 IMU, and the resultant information is transferred to the scanner card via Bluetooth low energy (BLE) communication. After that, the scanner card sends this incoming motion information to the evaluation software via USB by using CP2102 TTL to USB interface module.

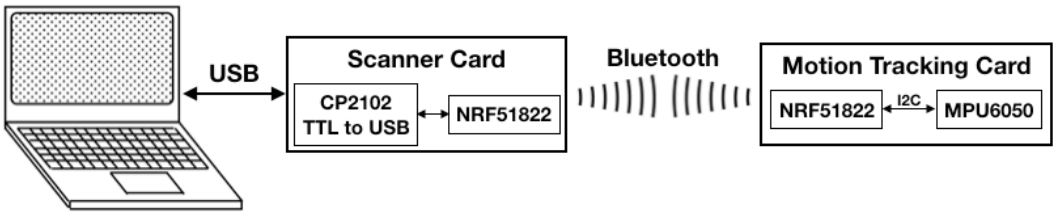


Figure 4.4. Hardware configuration for controller simulation studies

4.2.1. Motion Tracking Card

The MTC shown in Figure 4.5 is used to sense the movements of the surgeon by using NRF51822QFAA microprocessor and MPU6050 IMU. NRF51822QFAA is an ARM Cortex-M0 based 32-bit microprocessor that runs at 16 MHz clock frequency. This microprocessor has a built-in Bluetooth Low Energy (BLE) module, and it communicates with MPU6050 IMU through I2C interface at 400 kHz. The default measurement range of MPU6050 accelerometer is 2 g, gyroscope measurement range is 2000 dps, and the sampling rate is 100 Hz. MPU6050 has a data ready interrupt pin activated after every IMU measurement. After that, the MTC reads the IMU measurements and sends them to the scanner card over BLE communication. Additionally, MTC uses Nordic's Direct Firmware Update (DFU) feature to update the microprocessor wirelessly. The

microprocessor firmware can be updated wirelessly by using DFU feature when specific modifications are necessary. MTC works with a single CR2032 battery that allows longer than 24 hours of operation without battery replacement.

Figure 4.5. Motion tracking card with MPU6050 IMU

MTC performs the following operations:

- MPU6050 IMU on the MTC is calibrated automatically, while the MTC is in stationary condition on a flat surface. The card automatically calculates the offset values, and these offset values are written into the specific registers of the MPU6050.
- It broadcasts advertisement every second and connects to the scanner card.
- The beginning and end of user's movements are determined by using the activation button on the MTC. The control process of the robot starts after the user pushes on the activation button, and it stops when the button is pressed again. However, each button press generates a mechanical noise, which affects the IMU measurements. The MTC eliminates this mechanical noise by using a filter.
- After Bluetooth connection is established with the scanner card, the MTC transfers 15-byte data packets over the Bluetooth service. Each data packet contains 6 bytes for 3-axis accelerometer, 6 bytes for 3-axis gyroscope, 2 bytes for temperature, and 1 byte for MTC status. Also, the battery level of CR2032 is updated every 30 minutes by using battery service.

- The minimum and maximum connection intervals of the MTC are 7.5 ms and 9 ms respectively. In this case, 100 Hz data stream can be provided between MTC and scanner card.

Operation of the MTC is described briefly in Figure 4.6. The MTC starts to advertise its Bluetooth ID following initialization and offset calibration of the MPU6050 IMU. After establishing a connection with the scanner card, the MTC reads the necessary measurements and activation button state. IMU measurements are filtered by button spike filter to eliminate the mechanical noise of the activation button, and the filtered data packets are sent to the scanner card via BLE together with the MTC status.

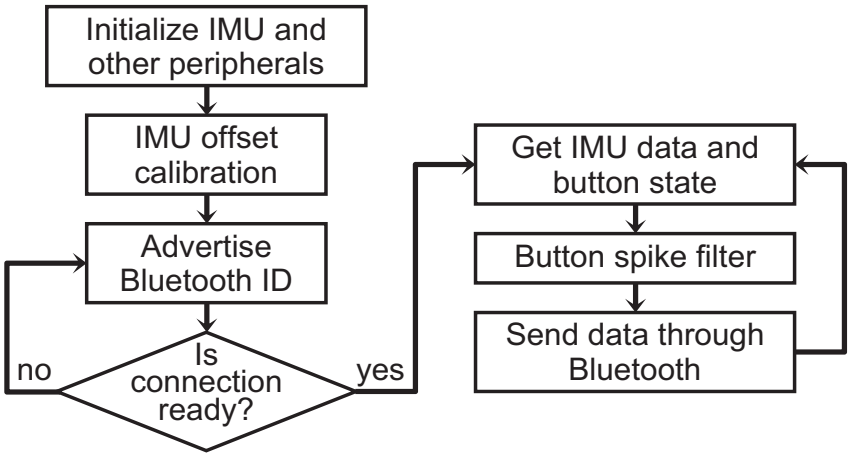


Figure 4.6. Operation of the MTC

The average current consumption of MTC is 6.184 mA while there is a connection with the scanner card. A typical CR2032 battery has 220 mAh energy capacity, and the MTC can work continuously for over 24 hours with a single CR2032 battery.

4.2.2. Scanner Card

The scanner card also has a NRF51822 microprocessor, which scans the MTC's specific Bluetooth advertisement. When it finds the correct advertisement data, it establishes a connection with the MTC. After that, scanner card processes the data coming from MTC and transfers it to the evaluation software. The scanner card uses an onboard

CP2102 USB to TTL interface module from Silicon Labs, to communicate with the evaluation software.

Figure 4.7. Scanner card with NRF51822

Scanner card monitors the broadcast of MTC, and when it captures the correct broadcast structure, it establishes the connection with the MTC. The scanner card receives 15-byte data packets from the MTC, organizes a specific string structure, and sends it to the evaluation software (i.e. PC). The evaluation software receives the IMU data through a USB driver that provides a virtual UART interface through the USB connection. The format of the string is "#:a_x:a_y:a_z:w_x:w_y:w_z:temperature:MTC_status:\r\n" where, a_x, a_y, and a_z are the accelerometer measurements, and w_x, w_y, and w_z are the gyroscope measurements of the x-axis, y-axis and z-axis, respectively. In a real application, this string would be sent to the robot controller hardware via UART. The speed of the connection between scanner card and evaluation software is 115200 baud-rate.

The operations performed by the scanner card can be described as seen in Figure 4.8. Scanner card searches for Bluetooth advertisement every 250 ms. If scanner card catches the correct broadcast of MTC, it establishes a connection with MTC. The correct advertisement of MTC is "CONTROL.CMR". After the connection is established, the measurements and MTC status are received from MTC. These data are translated into a

specific string format and scanner card transfers this string to the evaluation software via USB. Also, the battery level of MTC is read by battery service every 30 minutes.

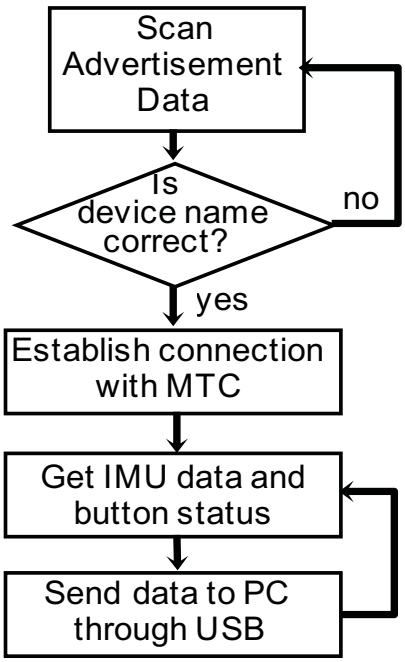


Figure 4.8. Operations of the scanner card

The SDK 12.3 software development kit provided by Nordic Semiconductor was used to program the NRF51822 processors. The softdevice version 2.0.1 that is produced by Nordic semiconductor is loaded to NRF51822 processor to enable its embedded Bluetooth module.

4.3. Bluetooth Low Energy

In this section, an overview of Bluetooth Low Energy (BLE) is given. BLE is the easiest method to establish a wireless connection and to transfer data between devices. BLE uses 2.4 GHz radio frequency and it is compatible with iOS, Android, and several other embedded systems. The devices designed with BLE technology can have over 1-year battery life with limited battery power (*32, 33*).

4.3.1. Generic Access Profile

Generic access profile is abbreviated as GAP in the BLE literature. GAP controls advertisements and connections of Bluetooth devices. There are four key roles for Bluetooth devices which are, peripheral, central, broadcaster and observer. Peripheral devices broadcast advertisement packets. Other devices recognize the presence of peripheral devices with these advertisements. Peripheral devices are generally required to have low power consumption, so they can work with limited battery power. Central devices can establish connections with peripheral devices and they are responsible for scanning of the peripheral devices' advertisements. If the central device needs to get more information without establishing a connection, it can send a scan response request to the peripheral device. The peripheral device replies with the scan response data. A central device is more complex and needs more processing power. Third key role is the broadcaster devices that cannot be connected to any device. Broadcasters are non-connectable and they only broadcast advertisement data. The last key role is for observer devices that receive advertisement packets from broadcasters and they are also non-connectable.

In this application, the peripheral device is the MTC and the central device is the scanner card. The procedure of advertisement between central and peripheral devices can be seen in Figure 4.9. Advertising data is used to determine whether the peripheral device is present or not. The scan response data generally includes name of the peripheral device, but also any information can be added into the scan response data. The data length of the advertising payload and scan response payload can be up to 31 bytes. Advertisement interval, refers to time between two advertisements of peripheral devices. This value can range from 20 ms to 10.24 s. The battery power will be adversely affected if shorter advertisement intervals are used.

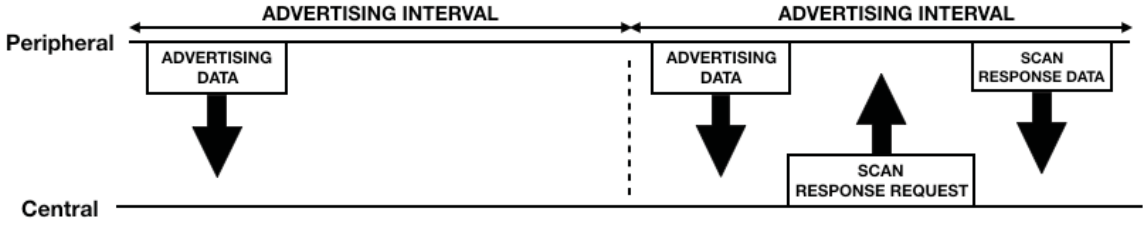


Figure 4.9. Advertisement procedure of Bluetooth Low Energy

If the central device wants to establish a connection, it must send a connection request to the peripheral. Afterwards, the peripheral device responds to the connection request and a connection is established between the two devices. In our application, if a scan response data is equal to "CONTROL.CMR", the central device (scanner card) establishes a connection with the peripheral device (MTC).

4.3.2. Generic Attribute Profile

The generic attribute profile named as GATT in the Bluetooth literature establishes server/client relationship and controls how Bluetooth devices interact with one another while they are connected. After the connection is established between central and peripheral devices, the GATT services and characteristics are used to transfer data between devices. A peripheral device establishes a connection with only one central device at a time, and it stops advertising until the end of the connection. However, the central device can establish connections with various peripheral devices. GATT is activated when an individual connection is established between two Bluetooth devices. The peripheral device is named as GATT server and the central device is named as GATT client in the GATT terminology.

As seen in Figure 4.10, a GATT client (central) sends a request to a GATT server (peripheral). After that, the GATT server sends the response to GATT client. All of the operations are started by GATT client. In this application, the connection interval ranges between 7.5 ms and 9 ms, because sampling rate of the IMU on the MTC is 100 Hz. This sampling rate requires a connection interval less than 10 ms to get all IMU data without any loss.

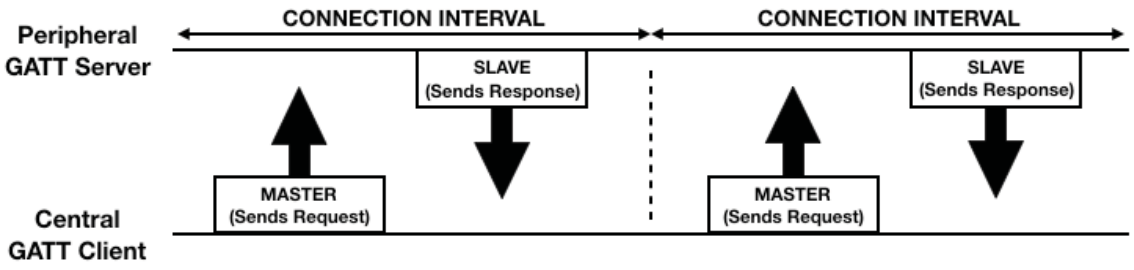


Figure 4.10. Communication procedure after Bluetooth connection

4.3.2.1. Services and Characteristics

Overall functionality of the BLE applications is represented as a profile that consists of all services and characteristics of the peripheral device. Each service has a universally unique identifier (UUID). If the service is officially defined by the Bluetooth special interest group (Bluetooth SIG), the length of UUID is 16 bit (i.e. battery service 0x180F, device information service 0x180A). If the service is manually defined by a programmer, then the length of UUID is 128 bit.

A service can consist of one or more characteristics to interact with the peripheral device, and these characteristics are also identified by 16-bit or 128-bit UUIDs. There are three types of characteristics which are, read, write and notify. The read characteristic represents that the central device can read the value of the characteristic. However, this value cannot be updated by central or peripheral device. The write characteristic represents that the central device can write specific values to the characteristic of the peripheral device. The notify characteristic represents that, the peripheral device can continuously send data packets to the central device. Initially, notification of the characteristic is disabled and there is no data transfer. The notify characteristic must be enabled by the central device to start the data transfer. The notification of the characteristic can be enabled by using client characteristic configuration descriptor (CCCD). Additionally, a characteristic can support multiple roles (i.e. write and read).

Figure 4.11. Advertisements of motion tracking card

As seen in Figure 4.11, the MTC is a peripheral device that has a 48-bit MAC ID given as "D9:D2:38:9A:7E:21". The scan response of the MTC is called "CONTROL.CMR". 1003 ms and -49 dBm refer to the advertisement interval and output power of the MTC. As seen in Figure 4.12, the profile of the MTC consists of six services. The two most important services are battery service and IMU data transfer service.

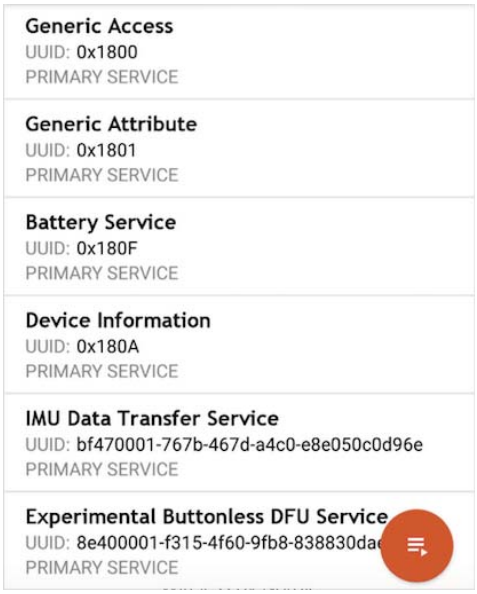


Figure 4.12. The profile of a motion tracking card

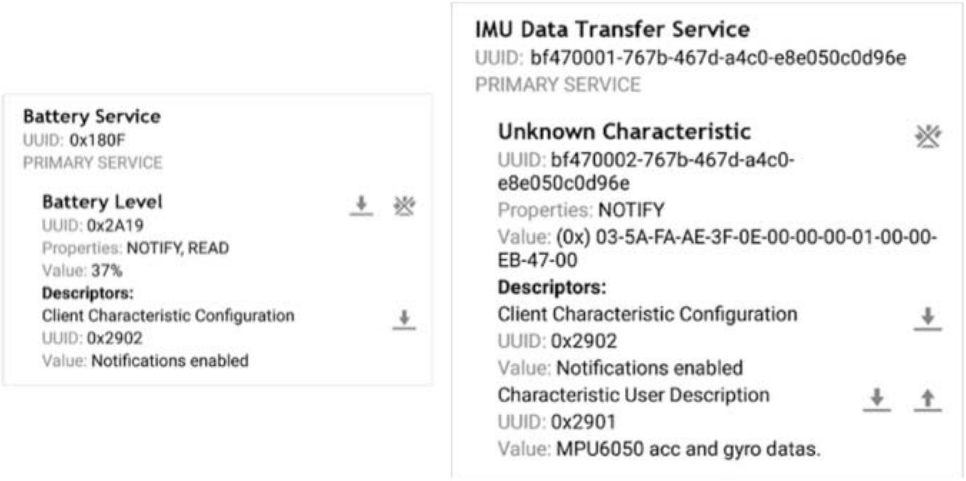


Figure 4.13. The battery and IMU data transfer services

As seen in Figure 4.13, the battery service has the battery level characteristic which is used to get battery level value of the MTC. IMU data transfer characteristic is used to transfer MPU6050's accelerometer data, gyroscope data, MTC status, and temperature data. Client characteristic configuration descriptor of these characteristics must be enabled to get notifications.

CHAPTER 5

EVALUATION OF INERTIAL MEASUREMENT UNITS

Three IMUs based on MEMS technology, LSM6DS3 from ST Microelectronics, MPU6050 from Inversense, and BMX055 from Bosch were evaluated in this study. All of these IMUs contain a three-axis accelerometer, a three-axis gyroscope, and a temperature sensor in a single device package suitable for design of motion tracking devices. The significant parameters specified in the datasheets of these IMUs are compared, and some of the critical specifications were analyzed by using the evaluation hardware.

5.1. MEMS Inertial Measurement Units

Three MEMS IMU modules, BMX055 from Bosch, MPU6050 from Inversense and LSM6DS3 from STMicroelectronics are shown in Figure 5.1, respectively. This evaluation provides the necessary reference information for comparison of accuracy and capabilities of MEMS IMU modules that are actively sold in the market.

Figure 5.1. MEMS Inertial Measurement Units

The measurement range of the accelerometers is selectable between ±2 g, ±4 g, ±8 g, and ±16 g in all of the three IMUs. The available measurement ranges in MPU6050's gyroscope are ±250 dps, ±500 dps, ±1000 dps, and ±2000 dps. BMX055's and LSM6DS3's gyroscopes support the same measurement ranges and an additional

range of ±125 dps. These digital sensors can be connected to a microprocessor through I2C or SPI interface (*34, 35, 36*).

Familiarity with the terminology used in the MEMS accelerometer and gyroscope datasheets is essential to understand their specifications, and to make a correct selection among these sensors. The critical specifications required for MEMS accelerometers are measurement range, sensitivity, sensitivity variation vs temperature, nonlinearity, zero-g output, zero-g offset vs temperature, and noise density. These parameters are explained in the following paragraphs (*37, 38*).

Measurement range is specified as a multiple of the g term for accelerometers. It is also named as full-scale range. The selected measurement range determines the maximum and minimum acceleration values an accelerometer can sense. For instance, if the accelerometer is in 2 g range the output signal of the accelerometer saturates at 2 g during 3 g acceleration. It is also noted that the breaking point of the accelerometer is specified as the absolute maximum acceleration, and it is much higher than the measurement range.

Accelerometer sensitivity is specified in unit of LSB/g for digital output of a MEMS accelerometer. The sensitivity defines the minimum incremental acceleration that an accelerometer can detect. Accelerometer sensitivity is affected by temperature, and its dependency is specified as percent sensitivity change per unit temperature variation in %/°C or %/K. The temperature dependency can also be specified as a total percent change for a specific temperature range (for example, from -40 °C to 85 °C) in some of the sensor datasheets.

Nonlinearity is defined as the deviation from the ideal linear relation between the actual acceleration and output signal of the accelerometer. Nonlinearity is specified as a percentage of full-scale range (%FS), and generally it can be ignored because its effect is lower compared to the other sources of error.

Cross-axis sensitivity expresses how much the acceleration in one axis is imposed on other axes of the accelerometer. This is defined as a percentage.

Zero-g offset level represents the output signal level when the accelerometer is motionless. The accelerometer on a flat surface should ideally produce 0g, 0g, and 1g output signals for x, y and z axes, respectively. However, the accelerometer output signals for x and y axes have non-zero values specified in mg for MEMS accelerometers in real applications.

Zero-g offset level versus temperature variation describes how much the output signal of the motionless accelerometer shifts for 1 °C temperature change. This is also named as bias temperature coefficient, and it is specified in mg/°C or mg/K for the accelerometer.

The accelerometer noise density specifies noise fluctuation with the inverse square of the bandwidth. If noise density is multiplied by square root of the measurement bandwidth, the RMS noise value of the accelerometer is obtained. This parameter is important because the noise of the accelerometer becomes more significant when detecting low acceleration changes.

Output data rate (ODR) parameter specifies how many independent acceleration data per second can be sampled by the accelerometer. System frequency bandwidth is half of the ODR value, and high-frequency movements beyond the system bandwidth cannot be detected by the accelerometer.

Table 5.1. Comparison of MEMS accelerometer parameters

	MPU6050	**LSM6DS3**	**BMX055**
Measurement Range (g)	±2, ±4, ±8, ±16	±2, ±4, ±8, ±16	±2, ±4, ±8, ±16
Sensitivity (LSB/g)	16384, 8192, 4096, 2048	16384, 8192, 4096, 2048	1024, 512, 256, 128
Sensitivity Change vs Temperature	±0.02 (%/°C)	±1(%) (from -40 °C to 85 °C)	±0.03 (%/K)
Zero-g Offset (mg)	±80	±40	±80
Zero-g offset vs Temperature	60 mg (from 0 °C to 70 °C)	±0.5 (mg/°C)	±1 (mg/K)
Nonlinearity (%FS)	0.5	-	±0.5
Cross Axis Sensitivity (%)	±2	-	1
Output Noise Density ($\mu g/\sqrt{Hz}$)	400	90	150
Output Data Rate	4Hz to 1000Hz	12.5Hz to 6664 Hz	8Hz to 1000Hz

Several specifications of the three accelerometers are nearly identical as seen in Table 5.1. For example, the measurement ranges of the accelerometers are equal to each other. Non-linearity and cross-axis sensitivity parameters have nearly the same value. Also, these two parameters are too small to be taken into account. The advantages and disadvantages of these sensors can be summarized as follows. The sensitivity values of MPU6050's and LSM6DS3's accelerometers are equal to each other, but BMX055 has a lower sensitivity value compared to the other two competitors because of 12-bit ADC resolution of BMX055 accelerometer, while LSM6DS3 and MPU6050 accelerometers have 16-bit ADC resolution. LSM6DS3 accelerometer stands out with a noise density of 90 $\mu g/\sqrt{Hz}$. This value is almost half the BMX055's noise density and almost one-fourth of MPU6050's.

The most important specifications on the datasheets of MEMS gyroscopes are, measurement range, sensitivity, sensitivity change over temperature, zero rate output, zero rate output over temperature, nonlinearity, cross-axis sensitivity and noise density. These parameters are summarized in the following paragraphs (*39*).

The unit of the measurement range is degree per second (dps). This parameter defines the maximum and minimum angular velocity that a gyroscope can sense. For example, if a gyroscope is used at ±250 dps range, the maximum and minimum values that can be measured with this gyroscope are +250 dps and -250 dps respectively. When a gyroscope experiences angular velocity beyond its measurement range, the gyroscope output will be saturated. This saturation will not have a permanent effect on its performance and structure.

The unit of gyroscope sensitivity is LSB/mdps for a gyroscope with digital output. The sensitivity of a gyroscope expresses the minimum angular velocity that a gyroscope can detect. The output of the gyroscope can easily be calculated by using this parameter.

Gyroscope sensitivity is affected by temperature and this is represented as sensitivity change over temperature in %/°C or %/K. The temperature dependency can also be specified as a total percent change for a specific temperature range (for example, from 0 °C to 70 °C) in some of the sensor datasheets.

Zero-rate output defines the output signal for all sensitive axes of the gyroscope when it is motionless. Ideally, this specification should be 0 dps, but it varies between ±1 dps and ±20 dps for MEMS gyroscopes. Temperature dependency of zero-rate level

can be specified in dps/°C as an incremental variation per unit temperature, or it can be specified in ±dps as the maximum deviation in a specific temperature range.

Non-linearity parameter defines the maximum difference in percentage between the output signal of the gyroscope and the best fit straight-line with respect to the measurement range.

Bandwidth parameter defines the mechanical frequency range of the gyroscopes. This parameter is specified separately for x, y, and z axes.

Table 5.2. Comparison of the MEMS Gyroscope Parameters

	MPU6050	**LSM6DS3**	**BMX055**
Measurement Range (dps)	±250, ±500, ±1000, ±2000	±125, ±250, ±500, ±1000, ±2000	±125, ±250, ±500, ±1000, ±2000
Sensitivity (LSB/dps)	131, 65.5, 32.8, 16.4	228.5, 114.2, 57.1, 28.5, 14.2	262.4, 131.2, 65.6, 32.8, 16.4
Sensitivity Change vs Temperature	±2% (from -40 °C to 85 °C)	±1.5% (from -40 °C to 85 °C)	±0.03(%/K)
Zero-rate Level (dps)	±20	±10	±1
Zero-rate level vs. Temperature	±20 dps (from -40 °C to 85 °C)	0.05 (dps/°C)	0.015 (dps/°K)
Nonlinearity (%)	0.2%	-	0.05%
Cross Axis Sensitivity (%)	±2%	-	±1%
Output Noise Density ($mdps/\sqrt{Hz}$)	5	7	14
Output Data Rate	4Hz to 8000Hz	12.5Hz to 1666Hz	100Hz to 2000Hz

As seen in Table 5.2, measurement ranges and sensitivity values are almost the same for all three gyroscopes with 16-bit outputs. Additionally, LSM6DS3 and BMX055 sensors can make measurements in ±125 dps range. While BMX055's gyroscope has 1 dps as the zero-rate level, this value is 10 times higher for LSM6DS3's gyroscope and 20 times higher for MPU6050's gyroscope. In this case, the gyroscope of BMX055 sensor

can be used without offset calibration, but the other two sensors must be calibrated at the beginning. While the noise densities of the MPU6050 and LSM6DS3 sensors are similar, the noise density of the BMX055 sensor is almost three times higher than this value. BMX055 is the least affected by temperature among all three sensors. This sensor is preferable in an environment with high temperature changes.

Table 5.3. Comparison of the MEMS IMUs Operating Condition

	MPU6050	**LSM6DS3**	**BMX055**
Operating Voltage	2.375V to 3.46V	1.71V to 3.6V	2.4V to 3.6V
Operating Temperature	-40 °C to 85 °C	-40 °C to 85 °C	-40 °C to 85 °C
Normal Operating Current for Accelerometer (μA)	500	240	130
Normal Operating Current for Gyroscope (mA)	3.6	-	5
Normal Operating Current for Accelerometer and Gyroscope(mA)	3.8	1.25	-
Power Down Current (μA)	5	6	5
Sensor Area	4 x 4 x0.9 mm	2.5 x 3 x0.83 mm	4.5 x 3 x 0.82 mm

As seen in Table 5.3, while MPU6050 and BMX055 sensors are working with almost the same supply voltage range, LSM6DS3 sensor can work with supply voltages as low as 1.71 V. This is an advantage for LSM6DS3, which is more beneficial in case of the systems operating with battery power. Considering the total current consumption of the sensors, the LSM6DS3 is obviously advantageous. The LSM6DS3 draws an average current value of 1.25 mA while MPU6050's average current value is 3.8 mA and BMX055 has an average current value above 5 mA. The operating temperature range is the same for all three sensors. LSM6DS3 has the smallest sensor area, which can be an advantage for a controller application that needs to be designed in small size.

5.2. Accelerometer Calibration

In this section, the calibration techniques of MEMS accelerometers are described, and results of some calibration procedures on three IMU models are presented. The measurement range is set to ±2 g and sampling rate is set to 100 Hz for all accelerometers. STM32F4 DDB and HC-05 Bluetooth 2.0 module are used for the data collection, as seen in Figure 4.1.

5.2.1. Accelerometer Offset Error

The output offset of an accelerometer can be calculated by taking the difference between the actual gravity components on the accelerometer axes and the output signals of the accelerometer at a particular position. The linear position is obtained by calculating double integral of accelerometer output signal that consists of acceleration due to physical effect a_p and accelerometer offset error a_{oe}. As a result, if there is a constant offset error a_{oe} in the measured acceleration, then the error in calculated position increases dramatically in time due to double integration of the constant offset error (*6*).

$$x = \iint a\, dt^2 = \iint (a_p + a_{oe}) dt^2 = \iint a_p\, dt^2 + \iint a_{oe}\, dt^2 \qquad (5.1)$$

Calculation of the accelerometer offset value is difficult, because the acceleration reading is continuously affected by gravity. The effect of gravity on the device in a particular position, must be known exactly to be able to calculate the offset value of the accelerometer in a predefined time interval. The most common calibration algorithms for accelerometer offset, manual calibration, freefall calibration, and simple calibration are discussed in the following sections.

5.2.1.1. Accelerometer Manual Calibration

It was stated that to calculate the offset value of the accelerometer reliably, it is essential to precisely know the gravitational forces acting on the accelerometer axes (*40*). The accelerometer to be calibrated can be placed on a flat surface, however, there is no

guarantee that the accelerometer senses 0 g in both x and y-axes, and 1 g in z-axis at this position. The offset value can be calculated without relying on positioning of the accelerometer by using the following manual calibration algorithm:

- Collect data from accelerometer while rotating the accelerometer 360° around all three axes.
- The maximum and the minimum values of the accelerometer output must be equal to 1 g and -1 g respectively because of the gravity.
- The differences between zero to 1 g and zero to -1 g changes give the offset value of the accelerometer.

5.2.1.2. Accelerometer Freefall Calibration

The freefall calibration algorithm is an alternative method for calibration of the offset values of an accelerometer (*40*). As mentioned before, all three axes of the accelerometer must give 0-g reading during free fall, since there is no gravitational force acting on the accelerometer sensor. Based on this condition, the offset value can be calculated with the following algorithm.

- Collect data from accelerometer during freefall. The free fall condition can be recognized by monitoring z-axis because the output of the accelerometer is 1 g at z-axis at the starting point and it suddenly drops to 0 g level when the free-fall begins.
- The offset value of each accelerometer axis is the average output value during free fall.

The advantage of this method is the fact that all three axes reach 0-g level at the same time. It is also noted that because of the air friction while the sensor is falling, offset value of the sensor axes found in this approach might not be as accurate as the results of manual calibration.

5.2.1.3. Accelerometer Simple Calibration

Simple calibration method does not give an accurate result, but it is the easiest approach for offset calculation that can be performed at the beginning of an operation

without any specialized equipment. In this method, the accelerometer is placed on a flat surface, and it is assumed that accelerometer must measure 0 g, 0 g and 1 g on x, y and z-axes, respectively. The following algorithm summarizes simple calibration approach to calculate an offset value of the accelerometer (*40*).

- Collect data from accelerometer while it is motionless for a certain time duration.
- Calculate the mean value of this dataset which gives the offset value of the accelerometer sensor.

The offset error can be eliminated by subtracting the calculated offset value from the accelerometer output signal. In the digital accelerometer, offset correction can be achieved by writing offset value to a specific register of the sensor. MPU6050's, BMX055's and LSM6DS3's accelerometers used in this study are calibrated by using the simple calibration method. All sensors are positioned on the same surface against gravitational force as accurately as possible while collecting data.

It is also known that a constant 1 g gravity is acting on the accelerometer while it is stationary, and each axis of the accelerometer is separately affected by this gravitational field. The gravity can be calculated by using accelerometer data as seen in equation 5.2. Additionally, by using the known gravity value g, it can be checked to what extent the offset calibration affects the accelerometer output.

$$g = \sqrt{{a_x}^2 + {a_y}^2 + {a_z}^2} \tag{5.2}$$

As seen in Figure 5.2, the gravity value measured from MPU6050's accelerometer is roughly equal to 0.91 g. This value indicates that the accelerometer needs calibration. After offset calibration, the gravity value measured from MPU6050's accelerometer is equal to 1 g as seen in Figure 5.3.

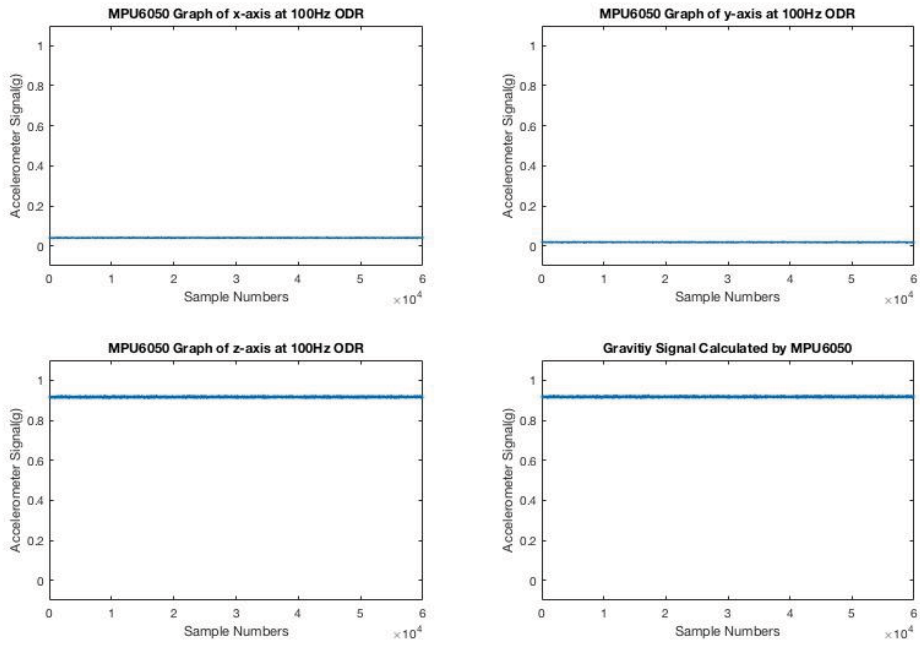


Figure 5.2. Output signal of the MPU6050's accelerometer before offset calibration

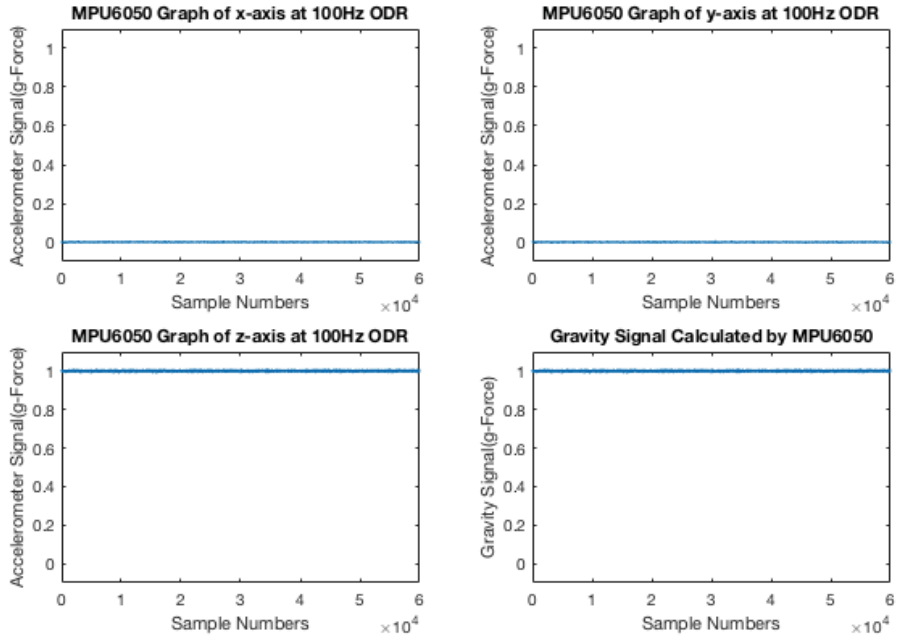


Figure 5.3. Output signal of the MPU6050's accelerometer after offset calibration

As seen in Figure 5.4 and Figure 5.5, BMX055 and LSM6DS3 accelerometer sensors are almost calibrated. Gravity values measured by the accelerometers of BMX055 and LSM6DS3 are nearly equal to 1 g. As seen in Figure 5.3, Figure 5.6 and Figure 5.7,

all sensors give the same results after offset calibration, and they measure gravity correctly after eliminating offset errors.

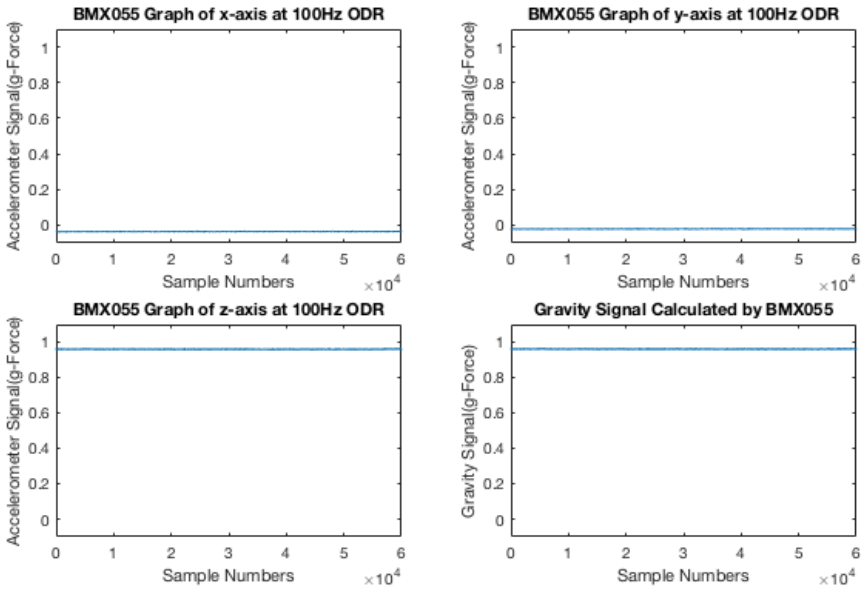


Figure 5.4. Output signal of the BMX055's accelerometer before offset calibration

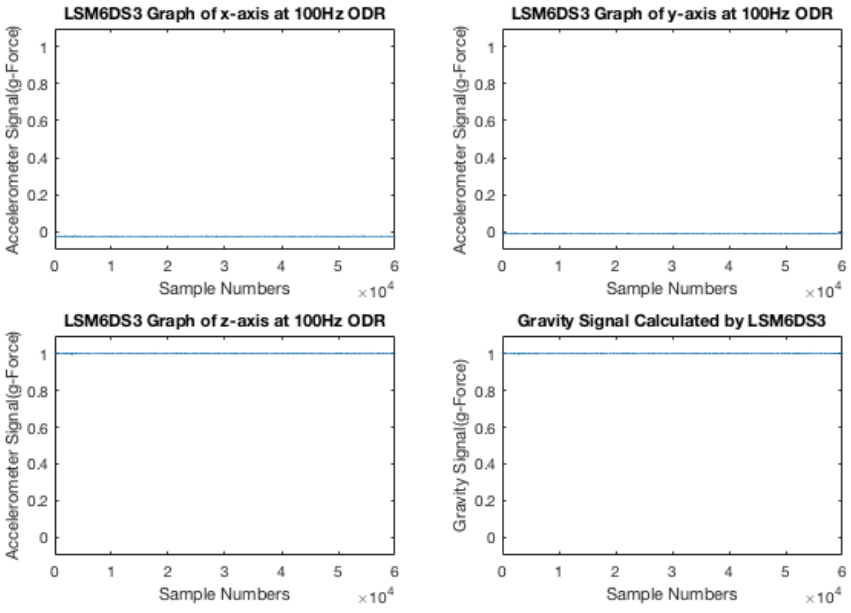


Figure 5.5. Output signal of the LSM6DS3's accelerometer before offset calibration

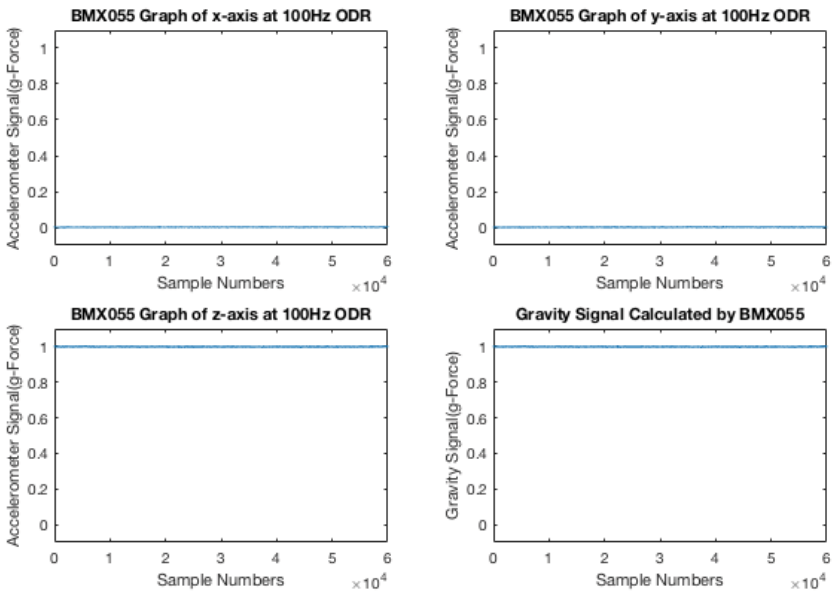


Figure 5.6. Output signal of the BMX055's accelerometer after offset calibration

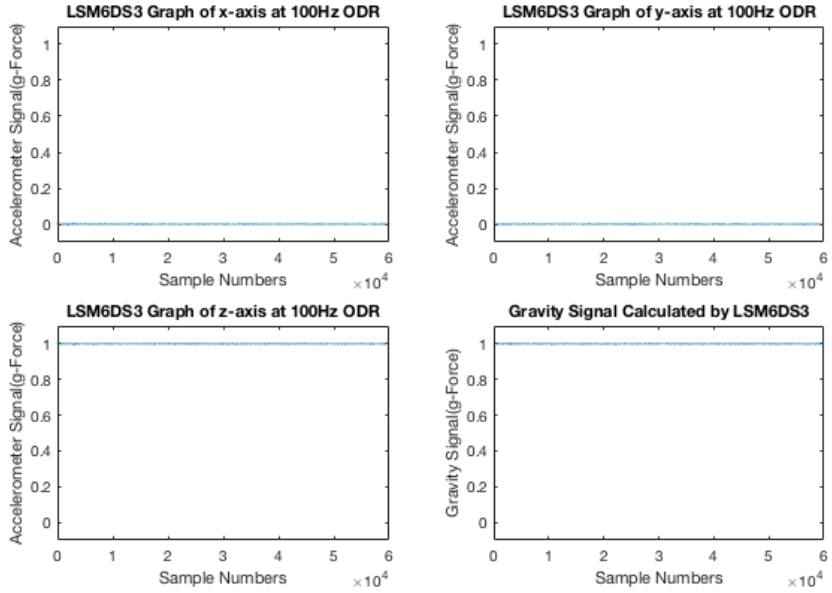


Figure 5.7. Output signal of the LSM6DS3's accelerometer after offset calibration

5.2.2. Accelerometer Output Stability

After offset calibration, data are collected from all of the accelerometers to determine the stability of accelerometer output signal. In this application, instantaneous

variations in the accelerometer output are more critical to assure a smooth motion tracking response. Because of that, it is better to check peak-to-peak variation to evaluate the accelerometer output stability. The following algorithm is used to evaluate the accelerometer output.

- Wait until the IMU temperature stabilizes and eliminate offset error.
- Collect data from accelerometers for ten minutes and calculate the maximum and minimum value of this dataset while accelerometers are motionless.
- Calculate the difference between maximum and minimum values. Smaller difference means better stability.

Table 5.4. Instability of the accelerometer sensors

	MPU6050	**BMX055**	**LSM6DS3**
Max value of x-axis (g)	0.0078	0.0059	0.0037
Min value of x-axis (g)	-0.0076	-0.0058	-0.0041
Difference between max and min value for x-axis (g)	0.0154	0.0117	0.0079
Max value of y-axis (g)	0.0082	0.0055	0.0018
Min value of y-axis (g)	-0.0077	-0.0052	-0.0019
Difference between max and min value for y-axis (g)	0.0159	0.0107	0.0037
Max value of z-axis (g)	1.0139	1.0074	1.0034
Min value of z-axis (g)	0.9875	0.9927	0.9925
Difference between max and min value for z-axis (g)	0.0264	0.0146	0.0109

As seen in Table 5.4, accelerometer of LSM6DS3 gives the most stable output signal. The worst one is MPU6050 and it is almost twice as noisy compared to LSM6DS3 for all axes.

5.2.3. Accelerometer Thermal Stability

The sensor temperature changes due to the heat generated by the sensor itself or environmental conditions. This affects the offset value of the sensor and causes a slowly changing error on angle calculations over time. Because of this, every accelerometer sensor contains an internal temperature sensor. It should also be noted that, temperature changes also cause fluctuations on the sensitivity of the accelerometer (6). However, this situation is not considered in this thesis, because the sensor used in the operating room is expected to perform well because the room temperature is not changing dramatically.

Temperature dependency of the accelerometers are determined by evaluating the data collected while sensor temperature is changing. The sensors are first warmed up until their temperatures reach roughly 40 ºC, and then they are placed in a refrigerator to drop sensor temperature to nearly 10 ºC.

As seen in Figure 5.8 and Figure 5.9, BMX055 and LSM6DS3 sensors were not affected by temperature changes. However, z-axis output of MPU6050 varies linearly with temperature as seen in Figure 5.10. Obviously, the temperature dependency varies from one accelerometer to another.

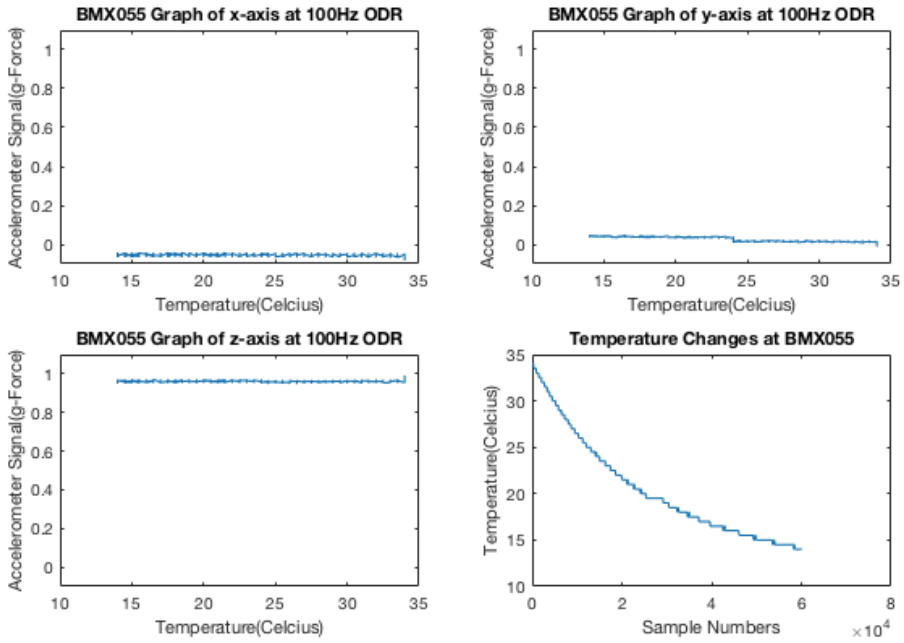


Figure 5.8. Temperature dependencies of the BMX055's accelerometer

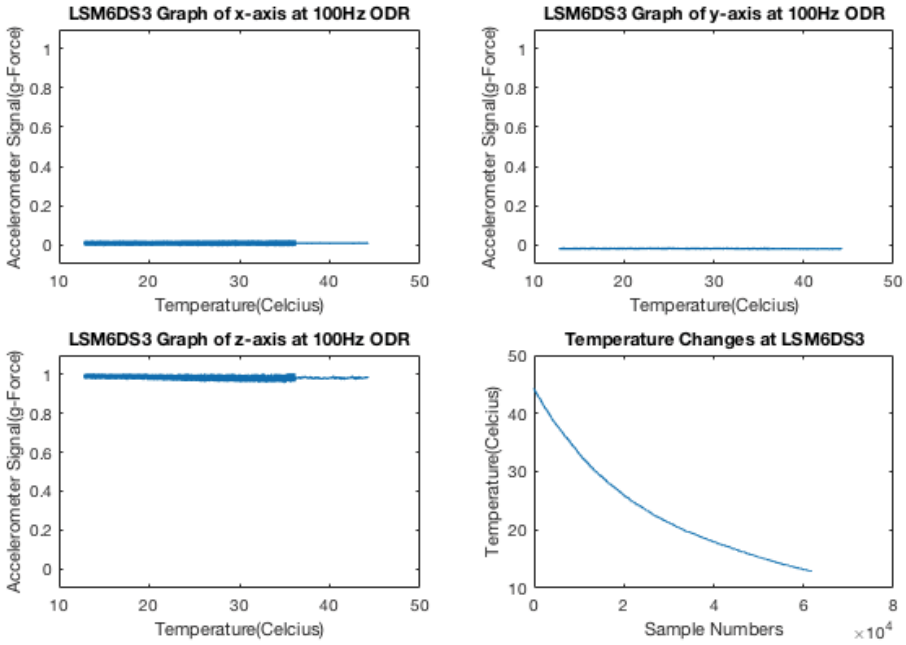


Figure 5.9. Temperature dependencies of the LSM6DS3's accelerometer

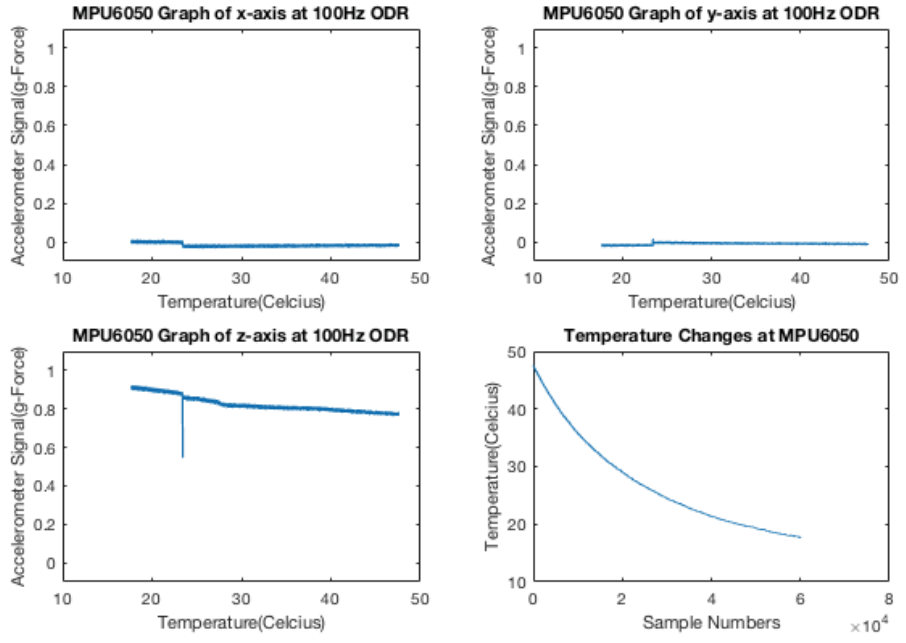


Figure 5.10. Temperature dependencies of the MPU6050's accelerometer

5.3. Gyroscope Error Calibration

In this section, error calibration methods for the gyroscope are examined, and the impact of calibration on the output signal and angle calculation results are presented. The measurement range is set to ±2000 dps for all sensors used in this work to make a fair comparison.

5.3.1. Gyroscope Offset Error

Offset error can be expressed as the average value of the gyroscope output signal while it is motionless (*41*). A gyroscope is expected to give zero output unless it is under a physical effect. It is known that angular position around an axis is calculated by taking the integral of gyroscope output ω (*6*).

$$\phi = \int_0^t \omega \, dt \tag{5.3}$$

The gyroscope output signal can be divided into two parts, gyroscope angular motion due to physical effect ω_p and gyroscope offset error ω_{oe}.

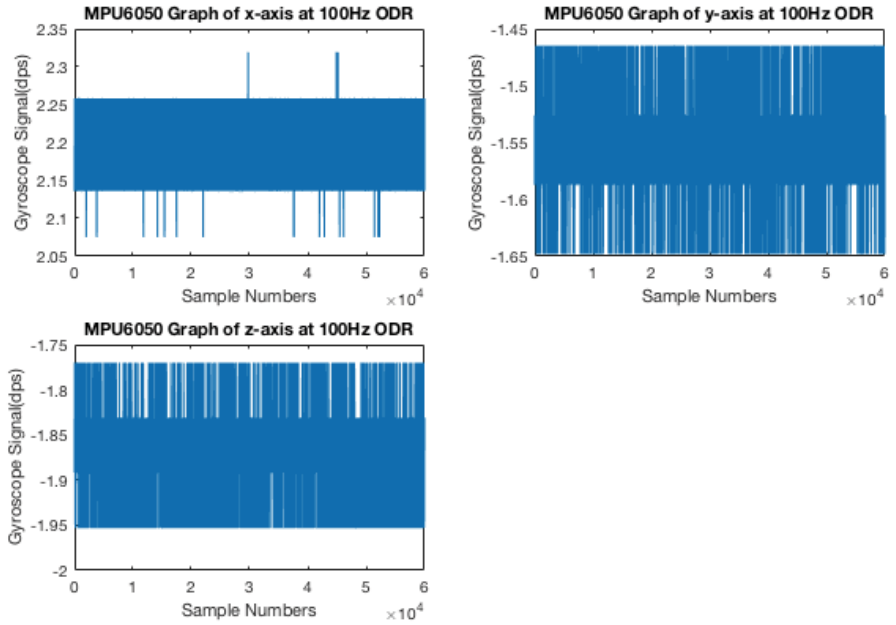
$$\phi = \int_0^t \omega dt = \int_0^t \omega_p + \omega_{oe} \, dt = \int_0^t \omega_p dt + \int_0^t \omega_{oe} dt \tag{5.4}$$

The integration of the offset error value ω_{oe} causes the error in angle calculation increase continuously. Assuming a constant offset, gyroscope offset error can be determined and eliminated by using the following method.

- Collect data from gyroscope in a specific time interval in the stationary condition.
- Calculate the mean value of this dataset that gives the offset values for all axes of the gyroscope.

The mean values are subtracted from gyroscope output signal to eliminate the offset error. In the digital gyroscope, offset errors can be eliminated by writing the offset values to specific registers of the sensor.

As seen in Figure 5.11, Figure 5.12 and Figure 5.13, all gyroscope sensors used in this work have offset values. However, BMX055 gyroscope sensor is nearly pre-calibrated. The output signals of the gyroscope sensors that have nearly zero mean values after offset calibration are illustrated in Figure 5.14, Figure 5.15 and Figure 5.16.



(a)

Figure 5.11. Output signal of the MPU6050's gyroscope before offset calibration.

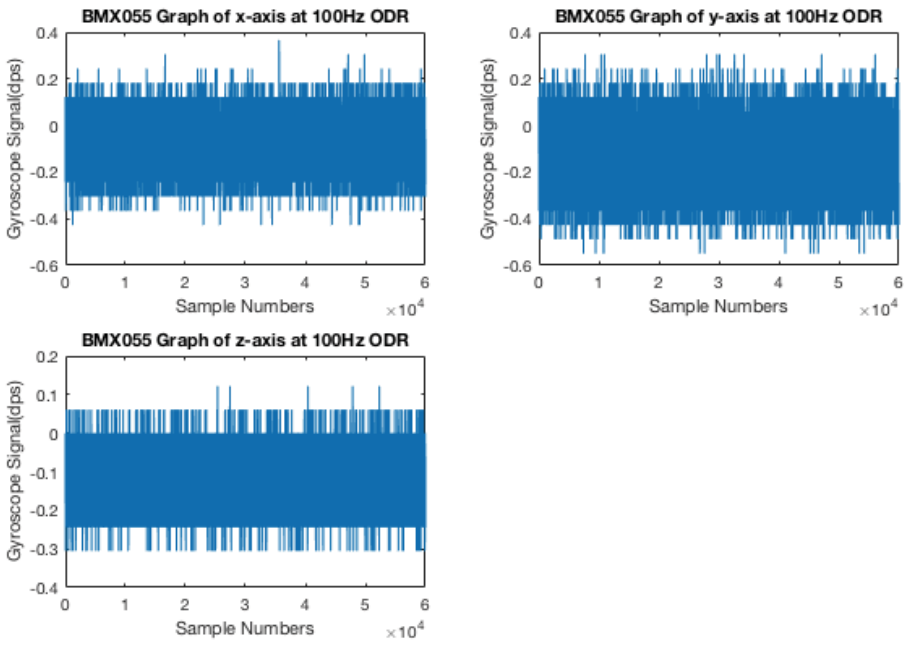


Figure 5.12. Output signal of the BMX055's gyroscope before offset calibration.

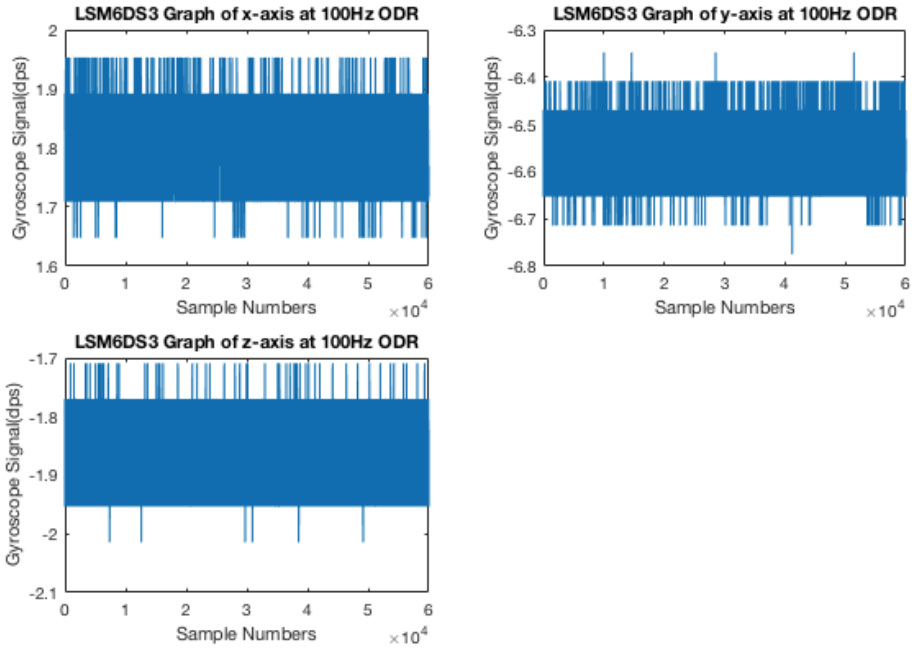


Figure 5.13. Output signal of the LSM6DS3's gyroscope before offset calibration.

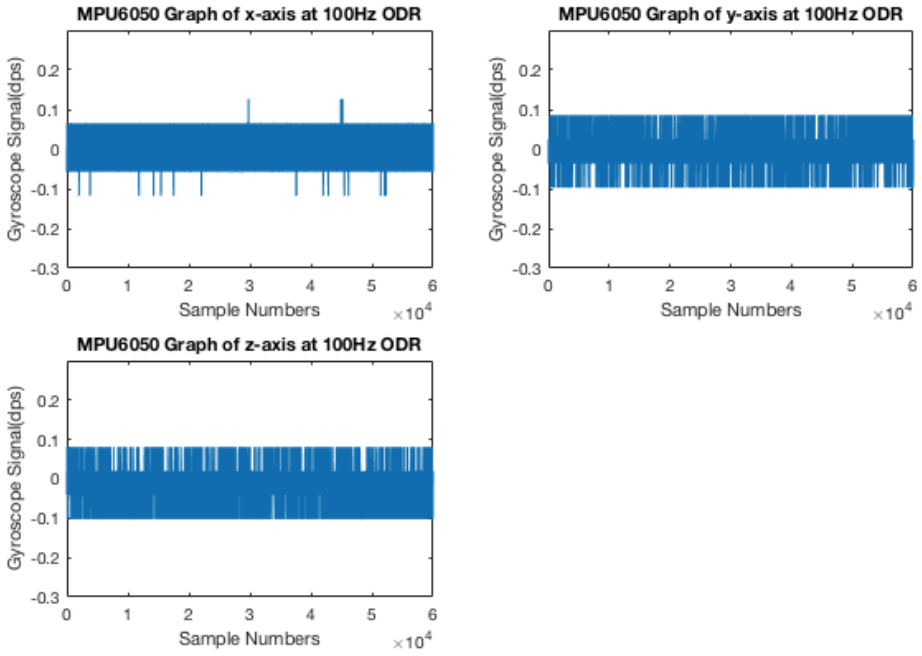


Figure 5.14. Output signal of the MPU6050's gyroscope after offset calibration

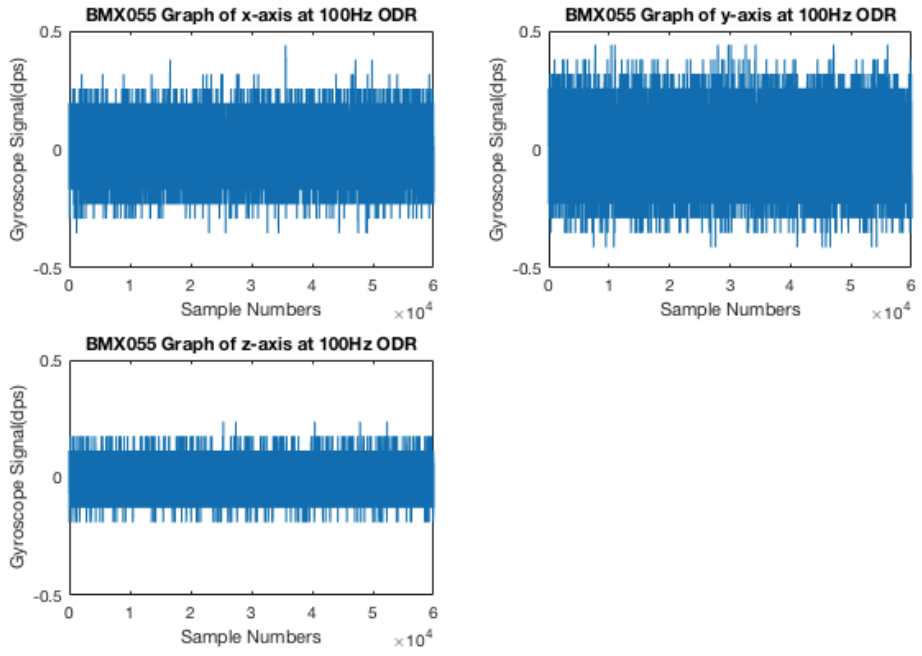


Figure 5.15. Output signal of the BMX055's gyroscope after offset calibration

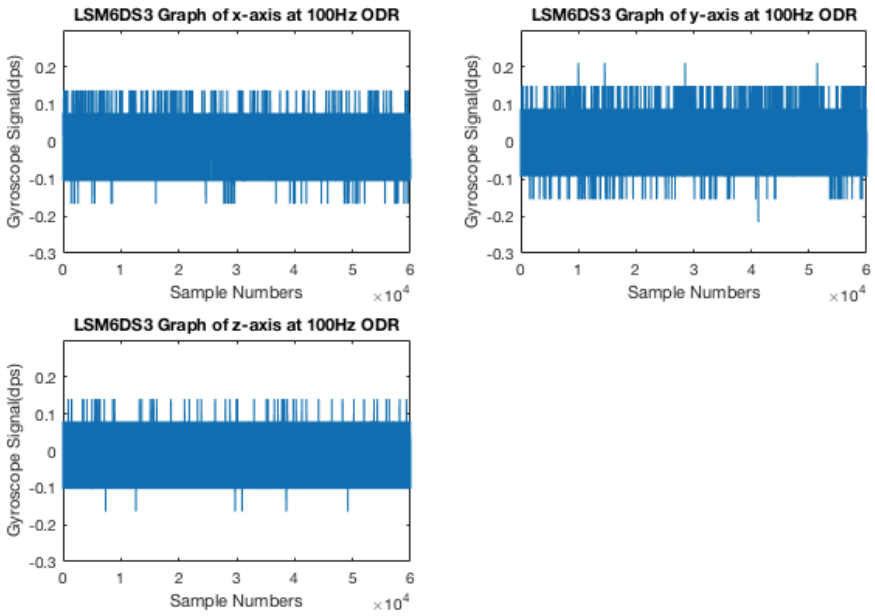


Figure 5.16. Output signal of the LSM6DS3's gyroscope after offset calibration

By using the offset calculation algorithm, offset values of the gyroscopes are determined as seen in Table 5.5. BMX055 has the lowest offset values in all three axes.

Table 5.5. Offset value of the gyroscope sensors before offset calibration

Sensors	Offset value for x-axis (dps)	Offset value for y-axis (dps)	Offset value for z-axis (dps)
MPU6050	2.1930	-1.5510	-1.8510
BMX055	-0.0737	-0.1362	-0.1157
LSM6DS3	-1.8151	-6.5589	-1.8498

The output data of gyroscopes were collected for ten minutes while sensors are in stationary condition. After that, the integral of the output signal of gyroscope is calculated. As seen in Table 5.6, the angle drift of each axis is over 90 degree per minute for MPU6050 and LSM6DS3 gyroscopes. In contrast, BMX055 has drift values below 8 degree per minute.

Table 5.6. Angle drift of gyroscope sensors before offset calibration

Sensors	Angle drift for x-axis (degree per minute)	Angle drift for y-axis (degree per minute)	Angle drift for z-axis (degree per minute)
MPU6050	131.8779	-93.3726	-112.26
BMX055	-4.5574	-7.9445	-6.8093
LSM6DS3	109.4638	-396.7025	-112.1077

After the offset calibration, the drift values fall from ~100 degrees per minute to less than 1 degree per minute for all sensors as seen in Table 5.7.

Table 5.7. Angle drift of gyroscope sensors after offset calibration

Sensors	Angle drift for x-axis (degree per minute)	Angle drift for y-axis (degree per minute)	Angle drift for z-axis (degree per minute)
MPU6050	-0.1969	-0.3037	-0.9658
BMX055	-0.4753	0.6906	0.1190
LSM6DS3	-0.8509	0.4860	-0.1431

5.3.2. Gyroscope Output Noise

Low frequency noise sources, such as thermal effects and flicker noise in the electronic components affect the output of gyroscopes. Because of this situation, the output signal fluctuates over time. RMS error of the gyroscopes' output signal can be evaluated to see how gyroscope responds to these noise sources.

As seen in Table 5.8, all gyroscopes have roughly the same RMS error values. However, MPU6050's gyroscope has the most stable output according to RMS-error values. The second one is LSM6DS3 and the third one is BMX055 among these gyroscopes.

Table 5.8. RMS error values for gyroscope sensors after offset calibration

Sensors	RMS error value for x-axis (dps)	RMS error value for y-axis (dps)	RMS error value for z-axis (dps)
MPU6050	0.0288	0.0328	0.0367
BMX055	0.0877	0.0751	0.0569
LSM6DS3	0.0526	0.0565	0.0468

5.3.3. Gyroscope Thermal Stability

A gyroscope is affected by temperature fluctuations due to environmental conditions and sensor power dissipation. If the output error varies linearly with temperature, then it can be compensated easily utilizing the on-chip temperature sensors available in all IMUs. On the other hand, compensation of errors resulting from a nonlinear temperature dependency is much more difficult or it is impossible in some cases.

Like the accelerometer, the sensor is warmed up, and then placed in the refrigerator to monitor the effect of temperature change on the sensor. Sensor data is collected at the stationary position while it was cooling in the refrigerator. As seen in Figure 5.17, LSM6DS3 has a linear relationship between temperature and the output signal. The change in output of gyroscope is roughly equal to 2 dps for all axes in 13ºC to 44ºC temperature range. In contrast, there is no detectable output signal change in BMX055 gyroscope sensor as seen in Figure 5.18, and it is the most stable sensor against temperature variation. It is also noted that MPU6050 has a nonlinear output variation as a function of temperature as seen in Figure 5.19.

If an IMU is to be used in the presence of large temperature variations, then BMX055 may be preferred because its output signal does not change against temperature variations. LSM6DS3 can be calibrated by using the embedded temperature sensor of the IMU due to the linear variation of the offset, however, MPU6050 cannot be calibrated against temperature changes because of the nonlinear output variation.

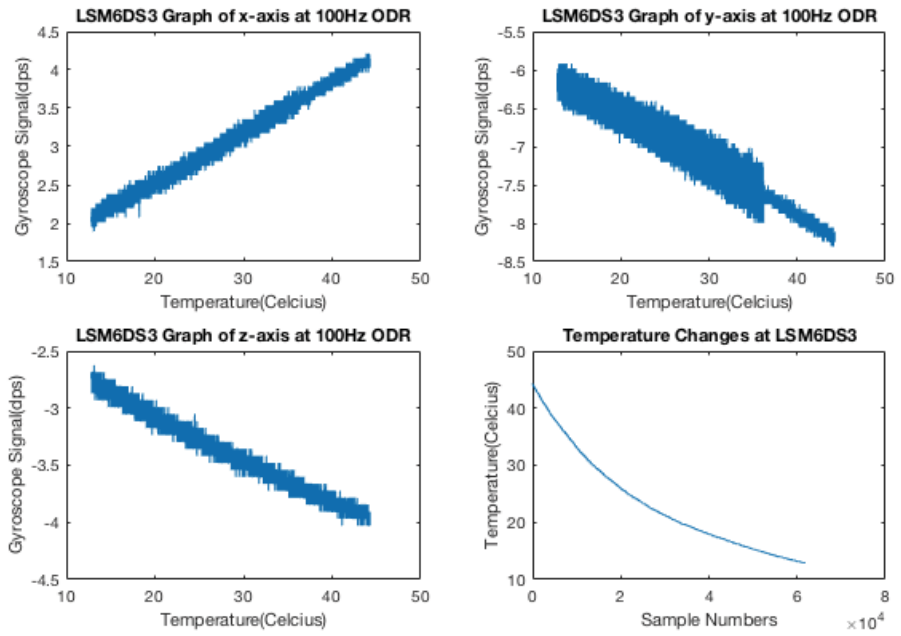


Figure 5.17. Temperature dependency of the LSM6DS3's gyroscope

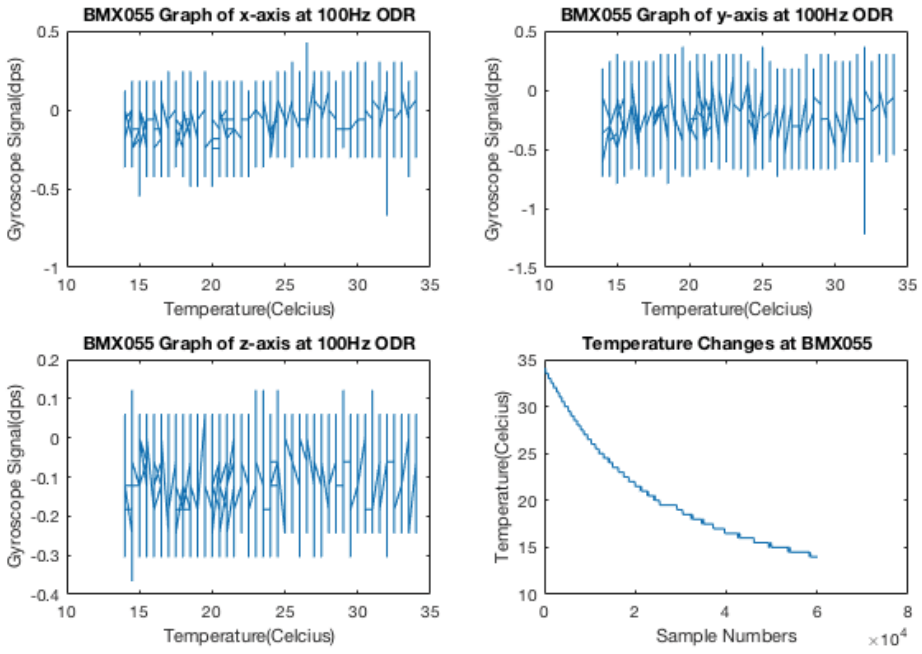


Figure 5.18. Temperature dependency of the BMX055's gyroscope

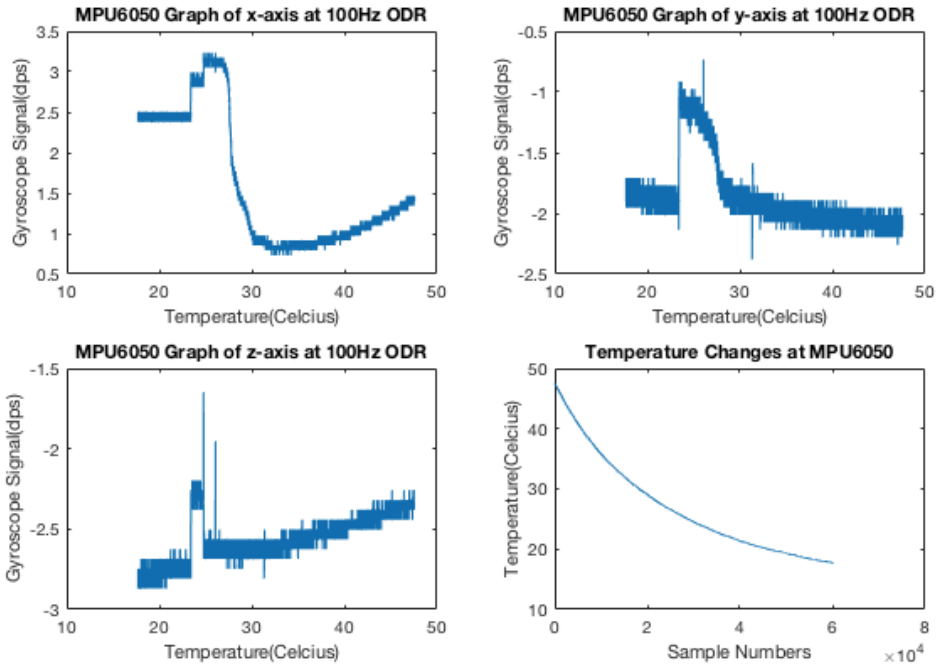


Figure 5.19. Temperature dependency of the MPU6050's gyroscope

5.4. Automatic Calibration of the IMU

The MEMS accelerometer and gyroscope sensors used in this study are made of mechanical parts that are etched into silicon. Offset values should be determined independently for every IMU device, and mechanical impacts may affect the offset values of sensors over long time. In addition, the MTC that has the IMU will be sterilized before each operation, heating up the IMU to a high temperature which may cause permanent damage to sensors very quickly. Because of these, IMUs must be calibrated automatically before every operation. A simple auto-calibration procedure has been developed based on the information provided by (*42, 43*).

The following method is used for auto-calibration of IMU's offset values. It is the simplest method and the only requirement is that, the IMU must be kept motionless on a flat surface. Additionally, this method does not require any specific equipment or subsequent rotation of the IMU. The auto-calibration method is summarized as follows.

- Wait for 10 seconds for stabilization of IMU output.
- Collect output data from accelerometer and gyroscope for 10 seconds.
- Calculate the mean value of the collected data.

- Write the calculated mean values into specific IMU registers.

All digital MEMS IMU sensors have specific registers to store the calculated offset values. The offset values of the IMU are automatically eliminated by using these registers.

CHAPTER 6

MOTION CONTROL SIMULATION

The performance of the IMU as a user interface for motion control was evaluated in simulation studies with different motion mapping methods and control system response times. The evaluation software developed for characterization and calibration of the IMU devices was extended to support these simulation studies. All evaluation tasks were accessible in several program sections that can be activated by using tabs on a single graphical user interface panel.

6.1. Evaluation System Software

The evaluation software used in this study was developed by using *Processing*, which is an open source programming language and development environment based on the Java language (*44*). Processing project was initiated by Ben Fry and Casey Reas in 2001, and it has been widely used as a graphical development environment supported by the *Processing Foundation* since 2012. The main functions and features of the developed evaluation software are described in the following paragraphs.

The evaluation software receives the accelerometer and gyroscope data from the scanner card through USB interface and calculates the orientation of the MTC by using various sensor fusion algorithms in the ORIENTATION section. Also, the algorithm that calculates the initial position of the MTC when the activation button is pressed, can be tested by using the BUTTON TEST command. As seen in Figure 6.1, the software supports several options related to the sensor fusion algorithms described in Chapter 3.

As shown in Figure 6.2, the accelerometer, gyroscope, gravity and linear acceleration signals can be monitored in the OSCILLOSCOPE section. Also, the calculated Euler angles and quaternion values can be displayed in this section.

The position-to-position and position-to-speed mapping methods are tested in the SIMULATION section shown in Figure 6.3. During the tests, users are asked to move the field-of-view (FOV) of a virtual camera towards a series of randomly generated targets and match the position and size of the target displayed on the screen. The evaluation

software creates a report file that registers the elapsed time and the maximum and minimum angle variations while the user completes each step of the experiment. By doing various tests, the users are able to determine the best control settings for their abilities and specialize in the control of the simulated FOV.

The sensor data coming from USB is collected in a specified time interval and saved in CSV (comma-separated values) spreadsheet format in the COLLECT DATA section shown in Figure 6.4. The CSV format is compatible with MATLAB, and all the datasets that are used in this thesis were obtained by using the COLLECT DATA section.

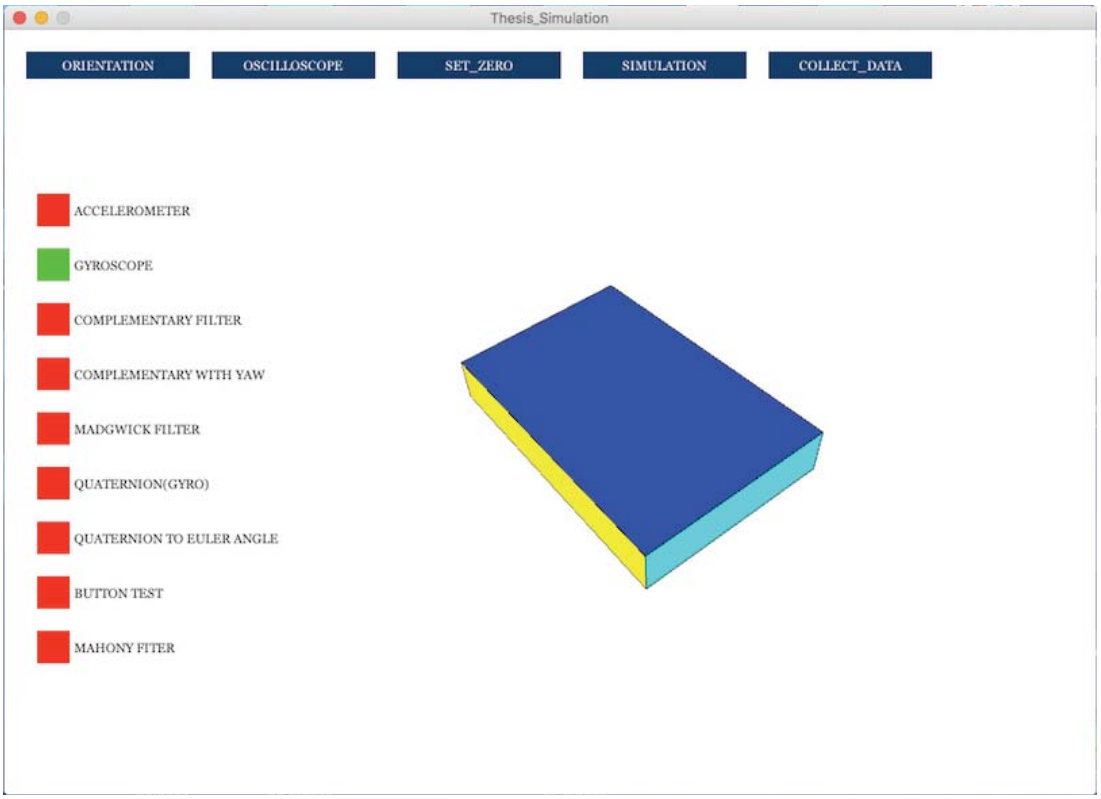


Figure 6.1. Orientation section of the evaluation software

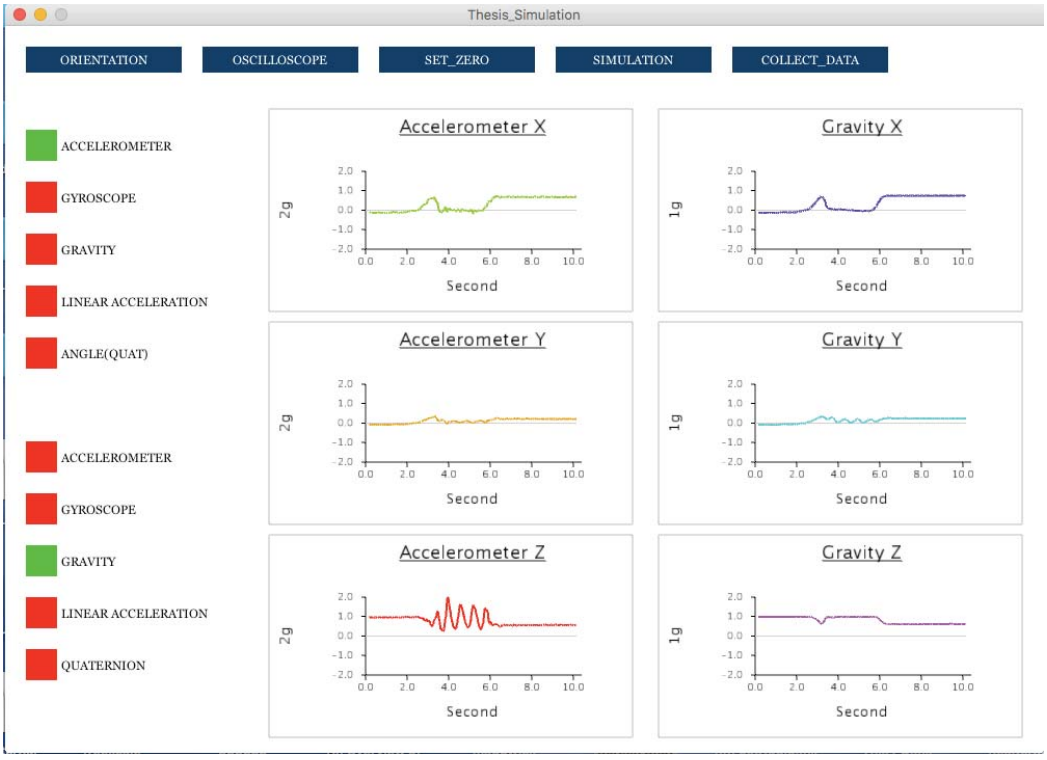


Figure 6.2. Oscilloscope section of the evaluation software

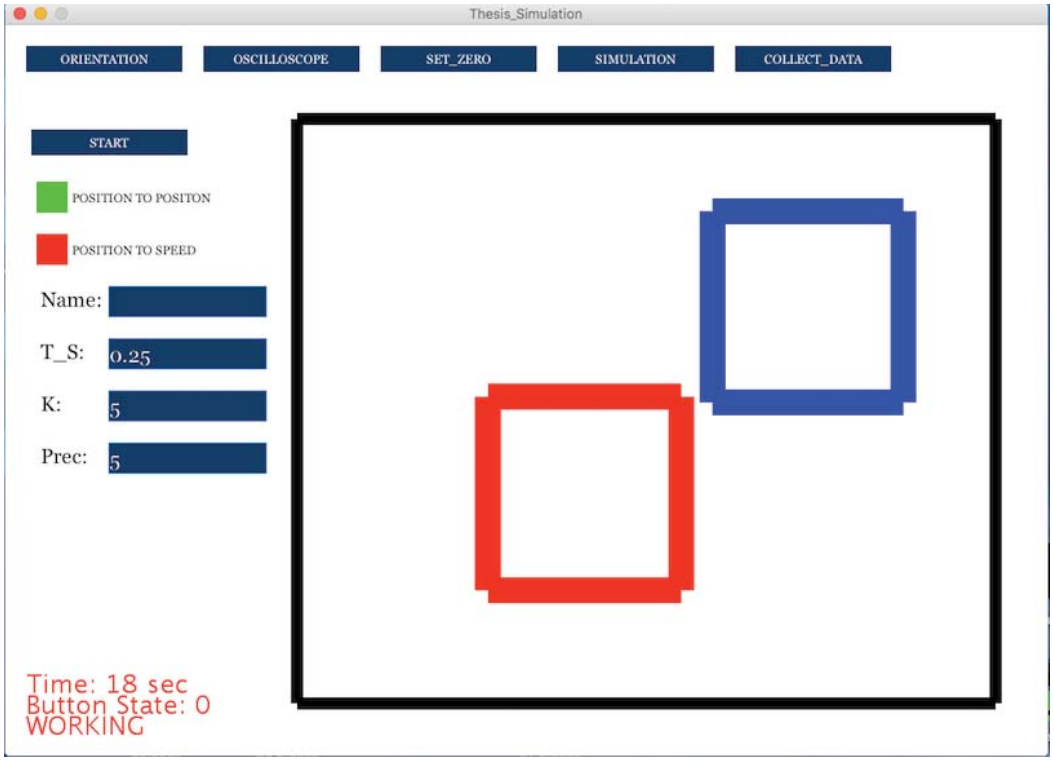


Figure 6.3. Simulation section of evaluation software

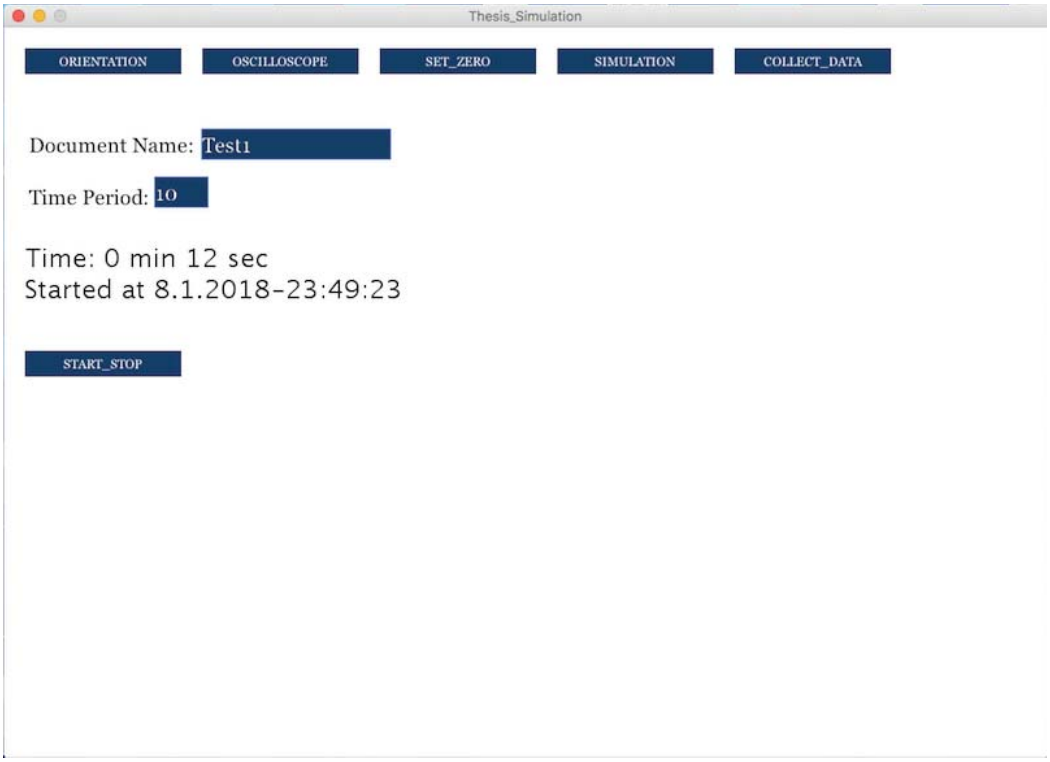


Figure 6.4. Data collection section of evaluation software

6.2. Position Control of the Endoscopic Robot

The evaluation software simulates the control process of the robot according to the MTC status determined by the activation button. The MTC status in the incoming data is checked, and no operation is performed as long as the MTC remains inactive (i.e. MTC status is 0). If the button is pressed for less than 0.5 seconds, then the MTC status is set to 1, and angular position of the MTC is used to control x and y position of the camera FOV. If the button is pressed for longer than 0.5 seconds then the MTC status becomes 2, and the x position data received afterwards are used for zoom-in and zoom-out operations that change the size of camera FOV. The active status of the MTC terminates when the activation button is pressed again for less than 0.5 seconds.

The motion mapping operation is separated into two phases, because in the evaluation studies it has been observed that simultaneous three-axis control for FOV position and size is a challenging task for the users. The control steps of the robot that are performed by the evaluation software, are summarized as follows.

1. Get IMU data and MTC status.

2. If MTC is inactive then go back to step-1 to get new data,

3. If the initial MTC orientation was determined, then continue with step-4. Otherwise, use the received data to calculate the initial MTC orientation and return back to step-1 to get new data.

4. Calculate the angular displacement of the MTC with respect to the initial orientation of the MTC.

5. Multiply the angular displacement with the gain factor K.

6. Apply a low pass filter to obtain a delayed response.

7. Calculate the control commands by using the position-to-position or positon-to-speed mapping method:

7.1. If the MTC status is 1, control the x and y position of FOV.

7.2. If the MTC status is 2, control the size of FOV according to x position (i.e. zoom-in or zoom-out).

8. Display the modified FOV and return to step-1

When the activation button is pressed, firstly the initial MTC position is determined by using Euler angles that are calculated according to accelerometer measurements. After that, the initial quaternion values are calculated based on the Euler angles as described by the equations 3.54, 3.55, 3.56, and 3.57. The initial position of the MTC for z-axis orientation is accepted as 0º because this reference information cannot be obtained with the accelerometer. As it is stated before, the magnetometer sensor or another external signal source must be used to determine the z-axis reference. However, the magnetometer is not used in this thesis, since the operating room has several ferromagnetic objects that affect the reliability of the magnetometer measurements.

Two restrictions are imposed during the simulation studies in order to avoid problems that may result from lack of reference information in the z-axis. The MTC is positioned to obtain 0º z-axis orientation at the beginning of each experiment, and the angular motion of MTC around x and y axes is kept within ±90º. These restrictions do not affect the user performance in the simulation studies, since the motion control system was designed to operate with small angular movements, and in fact all users were able to complete the targeting experiments with less than ±45º movements.

In a real surgical control application, the requirement for z-axis reference can be eliminated by mapping the reference axes of target camera position directly to the sensor frame axes. The MTC movements can be determined solely based on the gyroscope

outputs, without making use of the accelerometer data and fusion algorithms. In that case, prevention of gyroscopic drift will be an essential requirement. The calibration results given in Chapter 5 indicate that the positioning errors due to gyroscopic drift can be kept below ±1° by applying calibration and temperature compensation methods on the properly selected IMUs.

After the initial orientation values of the MTC calculated, the angular positioning errors due to the gyroscopic drift are corrected according to the accelerometer's gravity signal by using the Mahony filter described in section 3.4.3. Accumulation of angular positioning errors due to the gyroscope offset error in the z-axis will be more significant in the absence of a reference source. On the other hand, the surgeon is expected to activate the IMU for brief time periods that are shorter than one minute during surgery. Accumulation of angular positioning errors can be kept below 1° degrees in these short activation periods by applying the proper calibration and temperature compensation techniques.

The angular displacement of the MTC is calculated with respect to the initial orientation of the MTC. These angular displacements are multiplied by a gain factor K to obtain a convenient control range for the camera FOV with variable precision. The angular displacement values are passed through a low-pass filter to slow down the simulated robot behavior in order to obtain a more realistic response. Details of the filter calculations are given in the following section.

6.2.1. Simulation of Mechanical Response Time

Any real mechanical system will have a finite response time. The user evaluation tests were performed with a delayed system response in order to obtain a comparison of the two mapping algorithms under more realistic conditions.

A first order digital filter is implemented to introduce a delayed response similar to the response of an RC low-pass filter. The continuous domain representation of the filter is given by the following equation.

$$\tau\, \dot{\phi}_{dly}(t) + \phi_{dly}(t) = \phi_{mtc}(t) \tag{6.1}$$

where, τ is the filter time constant, $\phi_{mtc}(t)$ is the angular position of the MTC, and $\phi_{dly}(t)$ is the delayed position used as an input to the motion mapping algorithm. The discrete time filter response is given by

$$\tau \left(\frac{\phi_{dly}[k] - \phi_{dly}[k-1]}{\Delta t} \right) + \phi_{dly}[k] = \phi_{mtc}[k] \tag{6.2}$$

where Δt is the sampling time interval. An efficient way for calculation of the filter response can be obtained as follows.

$$\phi_{dly}[k] = a\,\phi_{dly}[k-1] + (1-a)\,\phi_{mtc}[k] \tag{6.3}$$

$$a = \frac{\tau}{\tau + \Delta t} \tag{6.4}$$

The filter time constant τ can be set as a parameter on the SIMULATION panel of the evaluation software. The evaluation software calculates the filter parameter a for the present values of τ and Δt, and generates the filter output in real time during the evaluation tests. The digital filter implementation is verified by monitoring the exponential response to a step input obtained with four different time constants (τ = 0.1s, 1.0 s, 2.0 s and 10.0 s).

6.2.2. Position Mapping Methods

Two different mapping methods were proposed to translate rotation of the MTC into targeting commands for the robot controller. The first method is the position-to-position mapping, that maps the angular position changes of the MTC directly to the target position of the camera FOV. The second method is the position-to-speed mapping, that controls the speed of the FOV motion according to the angular position of the MTC. Both of these mapping methods were implemented on the evaluation software to compare their efficacy in implementation of the user interface.

6.2.2.1. Position-to-Position Mapping Method

In simulation of the position-to-position mapping method, the camera FOV is directly controlled according to the MTC rotation relative to the initial position determined when the MTC is activated. The target changes in the FOV position are also calculated relative to the last FOV location displayed on the screen at the time of MTC activation. The resultant relative changes Δx and Δy in the FOV position are given by the following equations.

$$\Delta x = \left(\psi_{dly}[k] - \psi[0]\right) * K \tag{6.5}$$

$$\Delta y = \left(\theta_{dly}[k] - \theta[0]\right) * K \tag{6.6}$$

There, $\psi[0]$ and $\theta[0]$ refer to the initial MTC orientation, and $\psi_{dly}[k]$ and $\theta_{dly}[k]$ are the MTC positions at the low-pass filter output calculated for each sample data set. Similarly, the relative change in the FOV size is given by the relative position of MTC around the x axis when the active MTC status is 2.

$$\Delta z = \left(\psi_{dly}[k] - \psi[0]\right) * K \tag{6.7}$$

The relative changes, Δx, Δy, and Δz specify the number of pixels on the simulation screen to modify the FOV position and size, and unit of the gain factor *K* is pixel per degree.

6.2.2.2. Position-to-Speed Mapping Method

In position-to-speed mapping method, angular position of the MTC is linked to the speed of the robot motion. The aim of position-to-speed mapping is to achieve a larger robot control range with smaller angular variations. The speed values for FOV motion are calculated as:

$$v_x = \left(\psi_{dly}[k] - \psi[0]\right) * K \tag{6.8}$$

$$v_y = \left(\theta_{dly}[k] - \theta[0]\right) * K \quad (6.9)$$

$$v_z = \left(\psi_{dly}[k] - \psi[0]\right) * K \quad (6.10)$$

where, the initial MTC orientation is equal to $\psi[0]$ and $\theta[0]$ at x and y axes, respectively, and the delayed output is $\psi_{dly}[k]$ and $\theta_{dly}[k]$. In this case, the unit of the gain factor K is pixel/s per degree.

FOV position at x and y axes is controlled by taking time integral of v_x and v_y, when the active MTC status is 1.

$$\Delta x[k] = \Delta x[k-1] + \Delta t\, v_x[k] \quad (6.11)$$

$$\Delta y[k] = \Delta y[k-1] + \Delta t\, v_y[k] \quad (6.12)$$

Similarly, the relative position of MTC around the x axis is used to control size of the FOV, when the active MTC status is 2. FOV size (i.e. zoom-in or zoom-out) is controlled by taking time integral of v_z as follow.

$$\Delta z[k] = \Delta z[k-1] + \Delta t\, v_z[k] \quad (6.13)$$

6.2.3. Problems Caused by Activation Switch

The user starts control of the robot by pressing an activation push-button switch on the MTC. A mechanical vibration is generated when this activation button is pressed, and this vibration affects IMU measurements. Therefore, a button spike filter is developed to determine the duration of this mechanical noise and to eliminate it before further processing. The filtering algorithm suspends the processing of IMU data while the activation button is pressed, and it waits for a specified period of time after the activation button is released to enable the processing again. Figure 6.5 shows the spikes appearing in the MPU6050's accelerometer measurements when the activation button is pressed. The same accelerometer data after elimination of the spikes are given in Figure 6.6. The waiting period from the release of activation button to the enabling of processing is set to 20 ms according to the duration of vibrations seen in these accelerometer data sets.

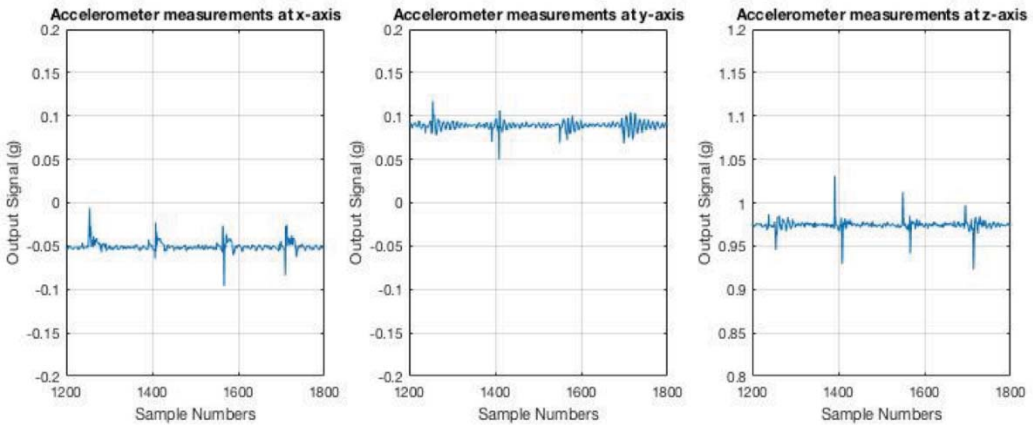


Figure 6.5. Spikes in the accelerometer measurements caused by the button press

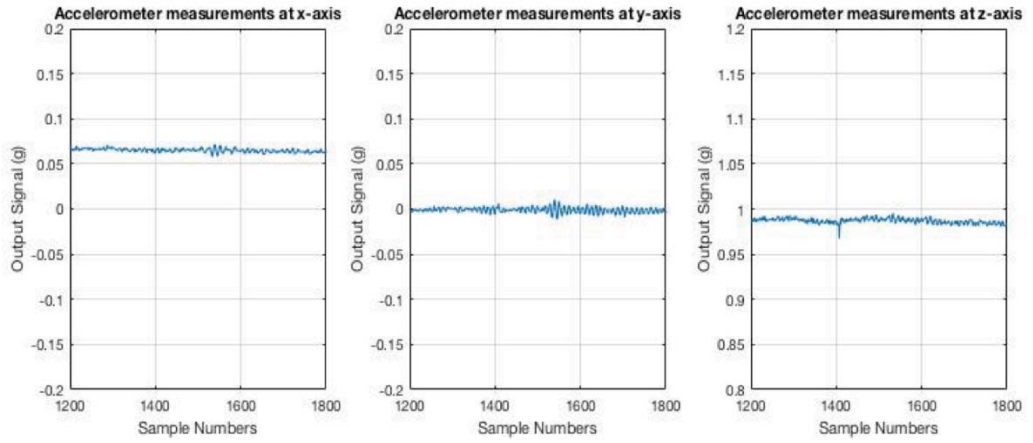


Figure 6.6. The accelerometer measurements after button spike filter

6.3. Comparison of Mapping Methods

The position-to-position and position-to-speed mapping methods described above were tested in the SIMULATION section of the evaluation software. The users are asked to match the randomly generated targets (red squares) with the simulated FOV of the camera (blue squares) as fast as they can with minimum angular rotation of MTC interface. The target is successfully matched when position and size of the blue FOV overlap the red target square with a matching tolerance specified in pixels. A new target is generated automatically after a target match, and this process is repeated for five targets in each test. The simulation program records the elapsed time, angular variations, and changes in the activation button state for each user to compare efficacy of the mapping methods.

Firstly, the position-to-speed mapping method and position-to-position mapping method are compared. In this comparison, the gain factor K is set to 8 for position-to-speed mapping and position-to-position mapping. Time constant is equal to 0.25 s for both methods. The matching tolerance is set to 9 pixels for both mapping methods.

Both mapping methods have been tested by six users, and all of them completed the test in a shorter time by using the position-to-position mapping method, as seen in Figure 6.7.

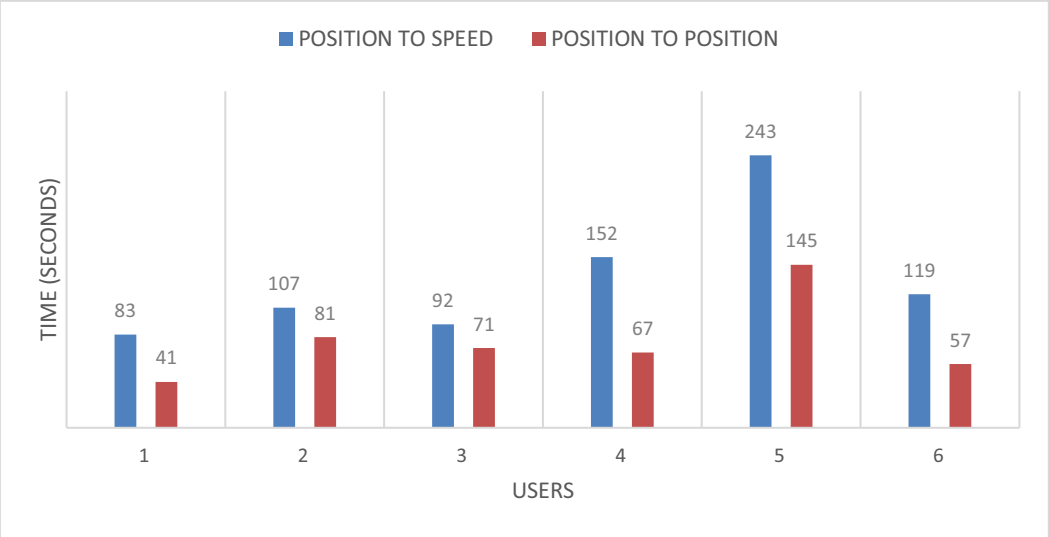


Figure 6.7. Elapsed time recorded for six users with two mapping methods

Comparison of number of activation button clicks required to complete the target sequence is given in Figure 6.8. All users except user 2 completed position-to-position mapping with fewer button clicks, however, the number of button clicks for user 2, 3, and 6 are almost the same for both methods.

Another comparison parameter was the average of maximum angular rotations of the MTC in each activation period required to successfully match the FOV targets. All users except 3 and 6 made smaller angular changes with the position-to-position mapping as seen in Figure 6.9. All users completed the tests with position-to-position mapping with less than 25º rotation for x and y axes. On the other hand, the 4th and 5th users needed more that 30º rotation in the tests with position-to-speed mapping.

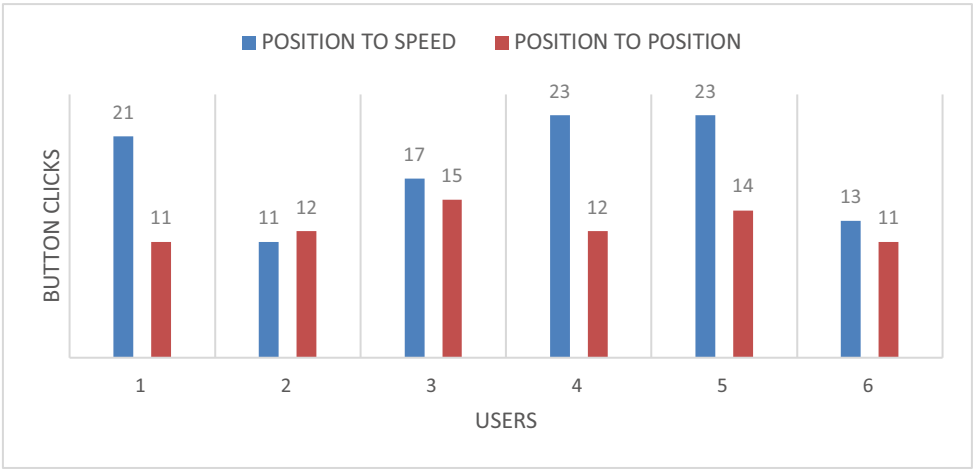
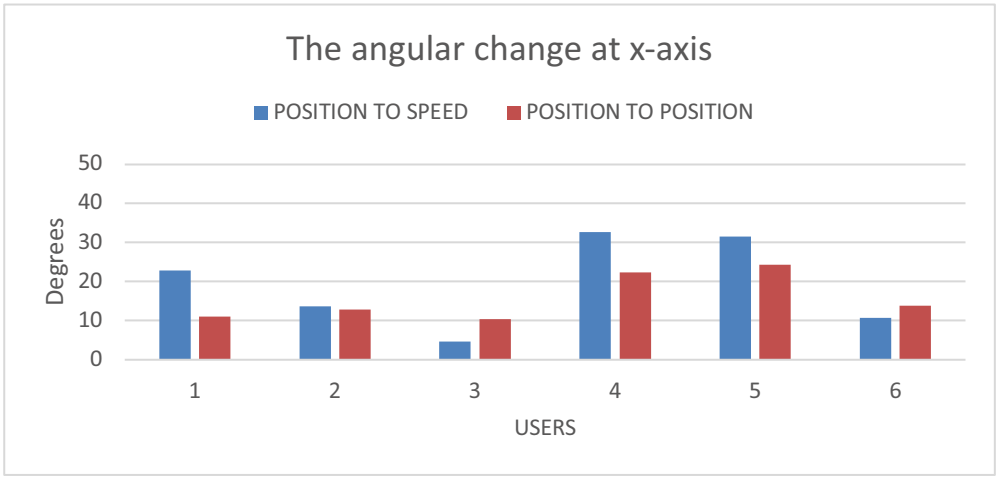
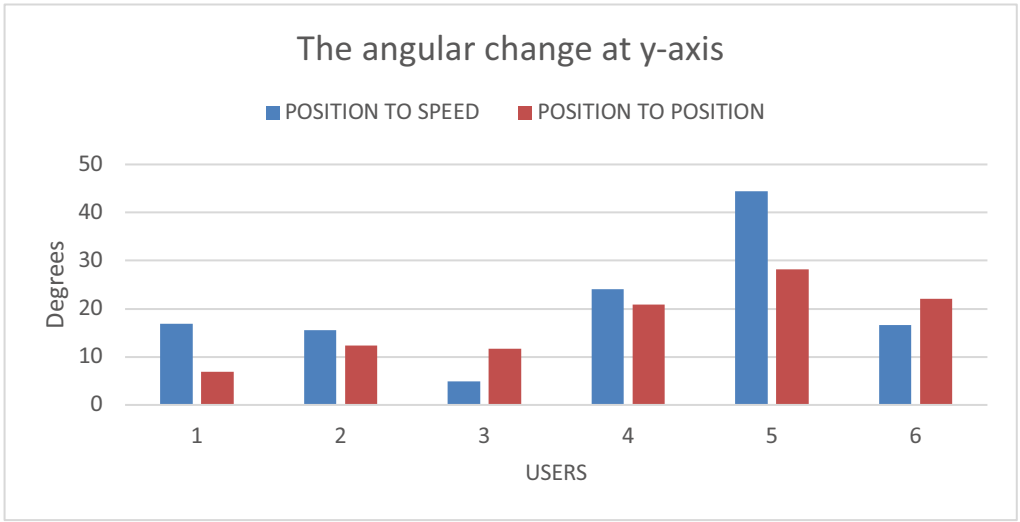


Figure 6.8. Total button clicks recorded for six users with two mapping methods



(a)



(b)

Figure 6.9. Average angular rotations recorded for six users with two mapping methods

It has been observed that the position-to-speed mapping method makes the control process more difficult and generally it requires more angular rotation in the tests performed with all users. However, as the users become familiar with position-to-speed mapping behavior, they can control the camera FOV with less angular rotation, but the elapsed time during simulation still remains longer due to the limited rate of position change. As a result, FOV can be controlled with smaller angular motion range by using position-to-speed mapping method, and this will be an important advantage in real surgical conditions where the user's movements are restricted.

6.4. Effect of Mechanical Response Time

The targeting experiments were repeated with filter time constant settings longer than 0.25 s while all other simulation parameters were kept the same. In these experiments, no significant changes were observed in the user performances for time constant settings as long as 1 s, so the comparison results for all users are given only for 0.25 s and 2 s time constants. As seen in Figure 6.10, all users completed the targeting tests with position-to-speed mapping in a longer time for 2 second time-constant. Position-to-position mapping with 2 second time-constant also requires a longer time for all users as seen in Figure 6.11.

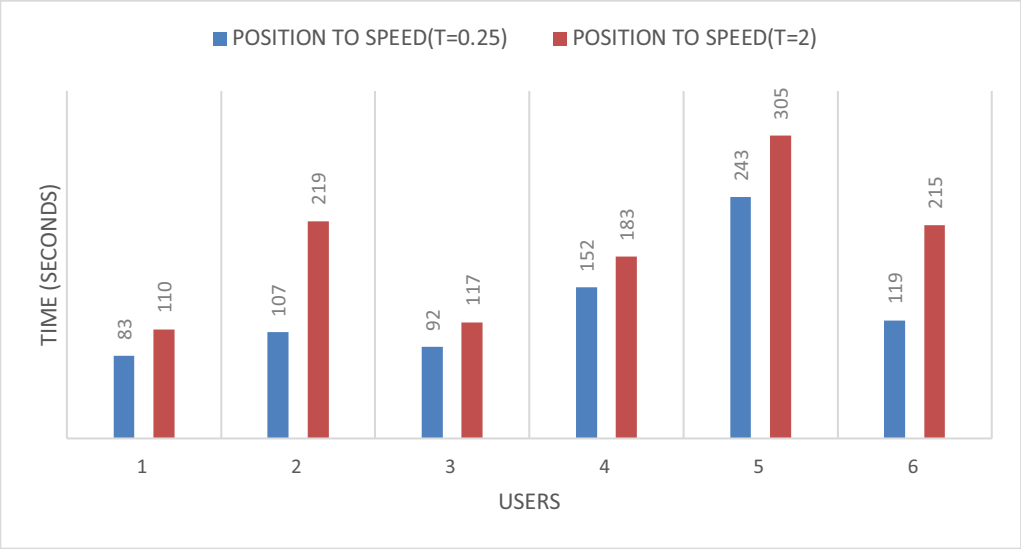


Figure 6.10. The comparison of position-to-position method with different time constants

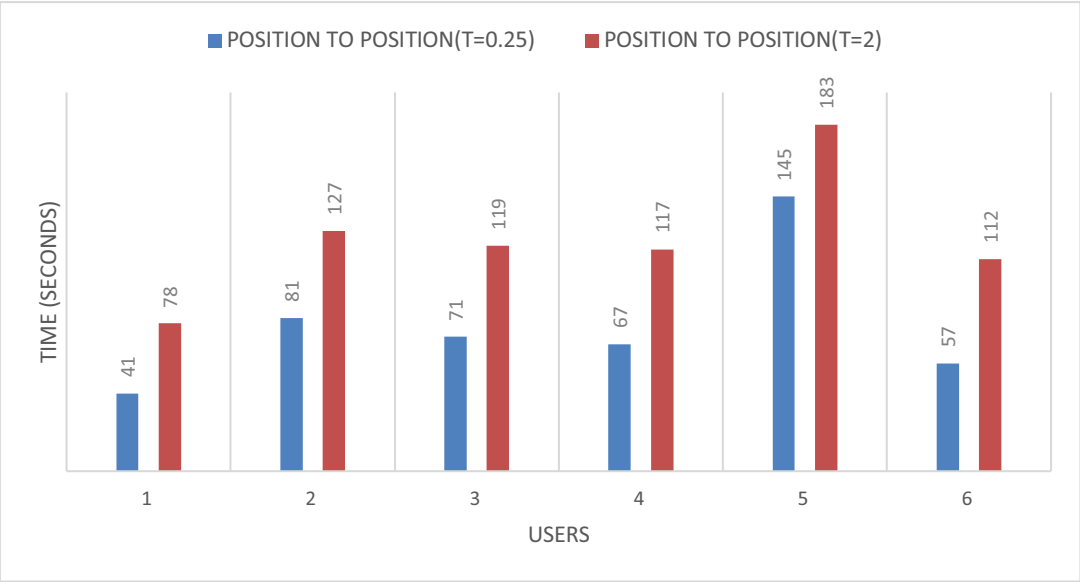


Figure 6.11. The comparison of position-to-speed method with different time constants

Table 6.1. Elapsed time recorded for position-to-position method with K = 8

	τ = 0.25 s	**τ = 0.5 s**	**τ = 1 s**	**τ = 2 s**
Experiment 1 (s)	47	46	37	55
Experiment 2 (s)	30	32	36	63
Experiment 3 (s)	42	35	47	51
Experiment 4 (s)	33	42	37	53
Experiment 5 (s)	30	40	31	50
Average Time (s)	36.4	39	37.6	54.4

The evaluation of motion control efficiency with different time constants was performed by the user who scored the shortest elapsed time in the previous tests. In this case, the elapsed time is recorded with four time-constant settings while all other parameters were kept the same. The experiments by using position-to-position mapping method were repeated five times as seen in Table 6.1. The average elapsed time to match

the FOV targets is almost the same for 0.25 s, 0.5 s and 1 s time constants, and it increases by 44 % when the time constant is 2 s.

Another set of five experiments were performed with position-to-speed method by the same user. Also, in these experiments, the average elapsed time remains the same for time constant settings up to 1 s, and it increases by 21 % when the time constant is 2 s, as seen in Table 6.2.

Table 6.2. Elapsed time recorded for position-to-speed method with K = 8

	τ = 0.25 s	**τ = 0.5 s**	**τ = 1 s**	**τ = 2 s**
Experiment 1 (s)	52	57	59	72
Experiment 2 (s)	60	65	63	75
Experiment 3 (s)	53	60	62	61
Experiment 4 (s)	50	50	63	64
Experiment 5 (s)	69	58	51	79
Average Time (s)	56.8	58	59.6	70.2

According to these results, significant changes in the elapsed time were seen only for time constants longer than 1 s with both of the mapping methods. This indicates that an experienced user is capable of compensating for increasing controller response time by utilizing the visual feedback. The delays in the controller response become significant when such delays are comparable to the time required to move the camera FOV to a target position.

CHAPTER 7

CONCLUSION

The endoscopic camera positioning with a robotic arm is a closed-loop control process where the surgeon is a part of the control loop. The robotic arm alone is a complex electromechanical structure that implements several feedback control mechanisms, but in the entire positioning process, the surgeon is the main controller making decisions based on the visual feedback. The endoscopic camera and the display screen are the major components in the feedback path of the process. The robotic arm and the motion tracking device that transmits surgeon's commands to the robotic arm are the forward path components in this process.

In all closed-loop systems, inaccuracies in the forward path components can be corrected by the feedback mechanism of the overall system. In case of the endoscopic camera positioning, the motion tracking device is a part of the forward path, and this fact should be kept in mind while deciding on the requirements of the device. The surgeon does not need to know how many degrees of rotation is required to perform a certain positioning action, and in fact, the surgeon does not care about the required motion range unless this requirement adversely affects the utilization of other surgical tools. Therefore, small variations in the motion tracking sensitivity and linearity can be tolerated. Also, the surgeon can easily compensate for slow changes in the reference position as a result of a few degrees of gyroscopic drift.

The purpose of the motion tracking device is to establish an efficient user interface that allows surgeon to smoothly control the endoscope position with minimal gestures. The critical requirements of the motion tracking device designed for this propose are: 1) sufficient position detection resolution ($<0.5^{o}$) to obtain a smooth controller response, 2) low tracking noise which should be negligible compared to the detection resolution, 3) offset calibration and compensation to keep gyroscopic drift in a tolerable range, 4) detection of initial orientation and position in less than 0.1 s with a fast startup procedure after every activation, 5) wireless data transmission and battery powered operation for unconstrained usage.

An evaluation system with a graphical user interface was developed to evaluate the critical properties of MEMS IMU devices and the efficiency of motion control methods with simulated targeting experiments. A wireless interface is established between the IMU modules and the evaluation software by using Bluetooth Low Energy technology to support transfer of accelerometer and gyroscope sensor readings 100 times per second together with device status information.

The interpretation of angular motion tracking data can be done directly in the sensor frame that changes with the sensor orientation, or it can be done relative to a fixed frame of reference. The latter requires translation of sensor data into the reference frame with a rotation matrix that is calculated according to the current sensor orientation in the reference frame. Quaternion representation is preferred in this study to avoid singularities in rotation operations. Three sensor fusion algorithms were implemented for accurate tracking of sensor position by combining gyroscope readings and the gravity measurements obtained with accelerometers. The Mahony filter fusion algorithm was preferred, because it generated a less noisy output and it allowed retrieval of initial position in a shorter time compared to the other methods.

The initial position of the IMU is determined every time the module is activated with a push-button switch. The large mechanical vibration observed at the accelerometer output during switching is eliminated by masking the sensor data for 20 ms after every activation. Gravity measured by the accelerometer is utilized to determine the initial IMU position in less than 0.1 s.

The fusion algorithms can eliminate the errors due to gyroscopic drift at the expense of additional noise that originate from the accelerometer. The reference positioning information around x and y axes is obtained based on the gravity detected by the accelerometer. A magnetometer can provide the z-axis reference in navigation applications, but it is not utilized in this study, since it is not expected to generate a reliable output in operating room conditions. Instead, the present z-axis position is taken as the reference after every activation of the motion tracking module. In a real application, other external reference sources can be used, or the motion tracking can be done as a function in the sensor frame, in which case the minimization of errors due to gyroscopic drift becomes an essential requirement.

Three MEMS IMU devices which are BMX055 from Bosch, MPU6050 from Inversense, and LSM6DS3 from STMicroelectronics, were evaluated. All of these modules have similar sensitivity and linearity specifications and they can be used in

design of wireless IMU modules that can operate for longer than 24 hours with a typical 3 V, 220 mAh CR2032 battery. The most critical parameters evaluated in this study are the gyroscope output noise, the gyroscope output offset and its temperature dependency. The RMS noise values measured from the three gyroscopes were between 0.0288 and 0.0877 degree per second. The worst-case peak-to-peak noise at the BMX055 gyroscope output was found to be nearly three times higher compared to the other IMU modules. It was possible to obtain a smooth position tracking information with better than 0.1 degree resolution by using any of the tested modules.

The angular position drift resulting from gyroscope output offset can be kept below 1 degree per minute after an offset calibration. This indicates that motion tracking in the sensor frame is possible with sufficient accuracy without any fusion algorithm or source of reference because of short integration time. Assuming that the activation periods of the controller will be shorter than one minute during surgery, tracking errors smaller than 1 degree per minute can be easily compensated by the user due to the closed loop nature of the overall control system. Although an offset calibration can be performed after power-up, the temperature dependency of gyroscope output offset still remains as a critical parameter. In the evaluation studies, BMX055 had nearly zero temperature dependency, probably due to a very good internal temperature compensation mechanism. LSM6DS3 gyroscope offset had a fairly linear temperature dependency that can be compensated easily by using the on-chip temperature sensor. However, 10ºC temperature variation in LSM6DS3 may cause more than 40 degrees positioning error in one minute without the temperature compensation. MPU6050 gyroscope offset had an erratic variation with temperature that makes the temperature compensation of gyroscopic drift very risky, if not impossible. Such an erratic behavior can be the result of a failing internal compensation mechanism or a partially defective IMU component.

In the simulation studies, angular position of a wireless motion tracking card (MTC) was used to control the position and size of the virtual camera FOV with position-to-position and position-to-speed mapping methods. It was necessary to assign separate control modes to change the position and size of the FOV, since simultaneous three-axis steering of the MTC was seen as a difficult task by all users. Efficiency of mapping methods was evaluated by six users while they tried to match the position and size of randomly generated FOV targets displayed on the screen. Although some of the users stated that position-to-speed mapping was easier to use, all users completed the targeting tasks in a shorter time with position-to-position mapping. Eventually, position-to-speed

mapping required integration of the user-entered position data that increased the order of the control process.

As the users gain experience with the position-to-speed mapping method, they were able to complete the targeting tasks by using a smaller angular motion range. This is an important advantage in operating room conditions, since the surgeon is working in a hazardous environment that seriously restricts the range of control movements. Also, the gain settings in mapping calculations and the speed limits required to ensure patient safety in a real application are the other important control parameters.

The targeting experiments were repeated with a delayed controller response by changing the time constant of the digital low-pass filter applied to the motion tracking data. The simulation studies done with both of the mapping methods indicated that the elapsed time in targeting experiments remained the same for the time-constant settings shorter than or equal to 1 s. This unexpected result can be explained by the users' ability in compensating for longer controller response time in the presence of visual feedback. This finding also shows that digital filters for noise reduction or other processing methods to achieve a smoother and safer controller response can be applied to a certain degree without degrading the user performance.

The development work in this thesis demonstrate that it is possible to build motion tracking devices for precise robot control applications by using commercially available MEMS IMUs and low-cost microprocessors. There are several options and control parameters that need to be determined according to personal preferences and operating conditions, and a control system should be able to accommodate all of these preferences. The end user –in case of an endoscopy robot, the surgeon– should be given sufficient time to specialize on the control system under realistic conditions in order to make healthy decisions.

Usage of an external magnetic field source can be investigated in a future study to obtain the z-axis reference, if it is required for interpretation of motion tracking data in a fixed frame of reference. The IMU modules readily equipped with 3-axes magnetometers can detect pulsed or alternating magnetic fields stronger than the earth's magnetic field, and they can make differential measurements to determine the z-axis reference in a few milliseconds. Also, non-linear mapping methods can be developed for better positioning precision and control efficiency. Further improvements on the motion tracking user interface is possible by implementing alternative activation methods, such as receiving motion-sensitive activation commands or placing a switch away from the

tracking card to avoid mechanical vibrations during switching. The motion tracking unit can be worn as a ring, or it can be attached to head or foot at the surgeon's discretion. A single receiver module can communicate with up to 8 Bluetooth devices allowing other configurations, such as two separate switching and tracking units on the index and middle fingers.

REFERENCES

1. Camarillo, D. B.; Krummel, T. M.; Salisbury, J. K. Robotic technology in surgery: past, present, and future. *The American Journal of Surgery* **2004,** *188* (4), 2-15.

2. Pandya, A.; Reisner, L. A.; King, B.; Lucas, N.; Composto, A.; Klein, M.; Ellis, R. D. A Review of Camera Viewpoint Automation in Robotic and Laparoscopic Surgery. *Robotics* **2014,** *3* (3), 310-329.

3. Sparkfun. Accelerometer Basics. https://learn.sparkfun.com/tutorials/accelerometer-basics (accessed April 2017).

4. Titterton, D. H.; Weston, J. L. *Strapdown Inertial Navigation Technology - 2nd edition;* The Institution of Electrical Engineers and The American Institute of Aeronautics and Astronautics: London, 2006.

5. Clifford, M. *Detecting Freefall with Low-G Accelerometer;* Application Note AN3151; NXP, 2006.

6. Woodman, O. J. *An introduction to inertial navigation;* University of Cambridge Computer Laboratory, 2007.

7. Sparkfun. Gyroscope Basics. https://learn.sparkfun.com/tutorials/gyroscope (accessed April 2017).

8. Awrejcewicz, J.; Koruba, Z. *Theory of Gyroscopes. In Classical mechanics. Applied Mechanics and Mechatronics;* Springer: New York, 2012; pp. 125-147.

9. Yazdi, N.; Ayazi, F.; Najafi, K. Micromachined Inertial Sensors. *IEEE* **1998,** *86* (8), 1640-1659.

10. LaValle, S. M.; Yershova, A.; Katsev, M.; Antonov, M. Head Tracking for the Oculus Rift. *2014 IEEE International Conference on Robotics and Automation (ICRA)* **2014,** 187-194.

11. LaValle, S. M. Oculus Developer Blog. https://developer.oculus.com/blog/magnetometer/ (accessed April 2017).

12. LaValle, S. M. Oculus Developer Blog. https://developer.oculus.com/blog/sensor-fusion-keeping-it-simple/ (accessed April 2017).

13. Higgins, W. T. A Comparison of Complementary and Kalman Filtering. **1975,** *AES-11* (3), 321-325.

14. Hide, C.; Moore, T. GPS and Low Cost INS Integration for Positioning in the Urban Environment. *Proceedings of the Institute of Navigation* **2005**.

15. Kim, N.; Kim, Y. Indoor Positioning System with IMU, Map Matching and Particle Filter. *Recent Advances in Electical Engineering and Computer Science* **2015**.

16. Ifixit. Samsung Galaxy S7 Teardown. https://www.ifixit.com/Teardown/Samsung+Galaxy+S7+Teardown/56686 (accessed January 2018).

17. Karpenko, A.; Jacob, D.; Baek, J.; Levoy, M. *Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes;* Stanford University Computer Science Tech Report, **2011**.

18. Reilink, R.; Bruin, G. D.; Franken, M.; Mariani, M. A.; Misra, S.; Stramigioli, S. Endoscopic Camera Control by Head Movements for Thoracic Surgery. *2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics* **2010,** 510-515.

19. Lin, W.; Navarro-Alarcon, D.; Li, P.; Wang, Z.; Yip, H. M.; Liu, Y.; Tong, M. C. F. Modeling, Design and Control of an Endoscope Manipulator for FESS. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* **2015,** 811-816.

20. Bihlmaier, A. *Endoscope Robots and Automated Camera Guidance. In: Learning Dynamic Spatial Relations;* Springer Vieweg, 2016; pp 23-102.

21. Jia, Y. B. *Rotation in the Space;* Iowa State University Computer Science 477/577 Notes, 2017.

22. Kok, M.; Hol, J. D.; Schon, T. B. Using Inertial Sensors for Position and Orientation Estimation. *Foundations and Trends on Signal Processing* **2017,** *1-2* (11), 1-153.

23. Slabaugh, G. G. *Computing Euler angles from a rotation matrix*, 1999.

24. Pedley, M. *Tilt Sensing Using a Three-Axis Accelerometer;* Application Note AN 3461; NXP, 2013.

25. Colton, S. *The Balance Filter;* Rev. 1: Submitted as a Chief Delphi White Paper, 2007.

26. Jia, Y. B. *Quaternion and Rotation;* Iowa State University Computer Science 477/577 Notes, 2017.

27. Dam, E.; Koch, M.; Lillholm, M. *Quaternions, Interpolation and Animation;* Technical Report DIKU-TR-98/5; University of Copenhagen, 1998.

28. Madgwick, S. O. H. *An efficient orientation filter for inertial and inertial/magnetic sensor arrays;* Report x-io and University of Bristol, 2010.

29. Madgwick, S. O. H.; Harrison, A. J. L.; Vaidyanathan, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. *IEEE International Conference on Rehabilitation Robotics* **2011,** 1-7.

30. Mahony, R.; Hamel, T.; Pflimlin, J. Non-linear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control* **2008,** *53* (5), 1203-1217.

31. Alam, F.; ZhaiHe, Z.; JiaJia, H. A Comparative Analysis of Orientation Estimation Filters using MEMS based IMU. *In Proceedings of the International Conference on Research in Science, Engineering and Technology* **2014**.

32. Townsend, K. Introduction to Bluetooth Low Energy. https:// learn.adafruit.com / introduction-to-bluetooth-low-energy/introduction (accessed December 2017).

33. Astebol, K. A short introduction to Bluetooth Low Energy. https:// devzone. nordicsemi.com/tutorials/b/getting-started/posts/introduction-to-bluetooth-low-energy-and-ble-devel (accessed January 2018).

34. Invensense MPU6050 Datasheet. https://www.invensense.com/wp-content/ uploads / 2015 / 02 / MPU-6000-Datasheet1.pdf (accessed November 2016).

35. STMicroelectronics LSM6DS3 Datasheet. https://www.st.com/resource/en/ datasheet/ lsm6ds3.pdf (accessed November 2016).

36. Bosch BMX055 Datasheet. https:// ae – bst . resource . bosch. Com / media / _tech / media / datasheets/BST-BMX055-DS000-02.pdf (accessed November 2016).

37. Dadafshar, M. *Accelerometer and Gyroscopes Sensors: Operation, Sensing, and Applications;* Application Note AN5830; Maxim Integration, 2015.

38. Anolog Devices. Acceleromter Specifications and Quick Definitions. http://www.analog.com/en/products/landing-pages/001/accelerometer-specifications-definitions.html (accessed January 2017).

39. *Everything about STMicroelectronics' 3-axis digital MEMS gyroscopes;* STMicroelectronics Technical Article TA0343, 2011.

40. Tuck, K. *Implementing Auto-Zero Calibration Technique for Accelerometers*, Application Note AN3447; Freescale Semiconductor, 2007.

41. Looney, M. *A Simple Calibration for MEMS Gyroscopes,* Analog Devices, 2010.

42. Tedaldi, D.; Pretto, A.; Menega, E. A Robust and Easy to Implement Method for IMU Calibration without External Equipments. *IEEE International Conference on Robotics and Automation* **2014**.

43. Cheuk, C. M.; Lau, T. K.; Kai, W. L. Automatic Calibration for Inertial Measurement Unit. *12th International Conference on Control Automation Robotics & Vision (ICARCV)* **2012,** 1341-1346.

44. Processing Home Page. https://processing.org (accessed January 2017).