POSITION ESTIMATION STUDY FOR A MOBILE ROBOT

by

151220122078 YASİN KAĞAN TAŞKIN

151220132069 ŞAFAK TAMBOVA

151220132062 ALAETTİN SÖMER

151220122092 ADEM YÜCEDAL

151220132108 BERKAY KOÇ

A Graduation Project Report

Electrical Electronics Engineering Department

JUNE 2018

POSITION ESTIMATION STUDY FOR A MOBILE ROBOT

by

151220122078 YASİN KAĞAN TAŞKIN

151220132069 ŞAFAK TAMBOVA

151220132062 ALAETTİN SÖMER

151220122092 ADEM YÜCEDAL

151220132108 BERKAY KOÇ

A Report Presented in Partial Fulfilment of the Requirements for the Degree Bachelor of Science in Electrical Electronics Engineering

ESKISEHIR OSMANGAZI UNIVERSITY

JUNE 2018

POSITION ESTIMATION STUDY FOR A MOBILE ROBOT

by

151220122078	**YASİN KAĞAN TAŞKIN**
151220132069	**ŞAFAK TAMBOVA**
151220132062	**ALAETTİN SÖMER**
151220122092	**ADEM YÜCEDAL**
151220132108	**BERKAY KOÇ**

has been approved by

Supervisory Committee

_____

Asst. Prof. Dr. Gökhan Dındış

_____

Prof. Dr. Osman Parlaktuna

_____

Asst. Prof. Dr. Burak Kaleci

_____

Prof. Dr. Rifat Edizkan, Chairperson

ABSTRACT

In the new world order, one of the most important thing is transportation from one point to another. Especially with Industry 4.0, mobile robots take a major place in our companies' production or in autonomous transportation. The aim of this project is to estimate the position and orientation of a mobile robot with less error and without any kind of connection (GPS, internet etc.).

This paper prepared for the final project of the bachelor Electrical and Electronics Engineering at Eskisehir Osmangazi University. Using the measurements of accelerometer and gyroscope from the Inertial Measurement Unit MPU6050, with proper techniques, orientation, and position estimated for a mobile robot with less error. The proper methods are quaternion, Kalman filter, zero update, position update, discrete integral. Methods are explained in detail.

Keywords: *mobile robot, MPU6050, STM32, position estimation, orientation estimation, Kalman filter, quaternion*.

ÖZET

Yeni dünya düzeninde, en önemli şeylerden birisi bir noktadan diğerine ulaşımdır. Özellikle Endüstri 4.0 ile, mobil robotlar şirketlerimizde ve otonom ulaşımda büyük bir yer almaktadır. Bu projenin amacı, bir mobil robotun konumunu ve yönünü daha az hatayla ve herhangi bir bağlantı (GPS, internet vb.) olmaksızın tahmin etmektir.

Bu makale, Elektrik Elektronik Mühendisliği bölümü lisans final projesi için Eskişehir Osmangazi Üniversitesi'nde hazırlandı. MPU6050 üzerindeki ölçüm biriminin gönderdiği ivme ölçer ve jiroskop verileri kullanılarak, uygun tekniklerle, mobil robot için yön ve pozisyon bilgisi az bir hatayla tahmin edildi. Uygun yöntemler, dördey, Kalman filtresi, sıfır güncelleme, pozisyon güncelleme, kesikli integral. Yöntemler detaylı bir şekilde açıklandı.

Anahtar Kelimeler: *mobil robot, MPU6050, STM32, pozisyon tahmini, yön tahmini, Kalman filtresi, dördey, kesikli integral*.

ACKNOWLEDGEMENT

We would like to thank our supervisor, Assistant Prof Dr. Gökhan DINDIŞ, for his useful comments and on-going support. We are also very grateful to Abdurrahim ÖZCAN, for his suggestions regarding the collection of results and useful comments on early drafts of our thesis.

Abdurrahim ÖZCAN
Electrical Engineer, BSc,
Profession: Mechatronics,
Experience: 8 years,
Organization: Modoya Elektronik Otomasyon Mühendislik Danışmanlık San. ve Tic. AŞ, Eskişehir

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol	**Explanation**
π:	Pi number
α:	Phase angle between voltage and current
T_A:	Ambient temperature (^oC)
t_{md}:	Motor disable time
BAqˆ:	Orientation of frame A relative to frame B
ABqˆ:	Orientation of frame B relative to frame A
BCqˆ:	Orientation of frame C relative to frame B
ACqˆ:	Orientation of frame C relative to frame A
$\otimes$:	Quaternion product
*:	Quaternion conjugate
R:	Rotation matrix
Φ:	Roll
θ:	Pitch
ψ:	Yaw

Abbreviation	**Explanation**
MEMS:	Micro-Electro-Mechanical Systems
MPU:	Motion Processing Unit
DMA:	Direct Memory Access
RS-232:	Recommended Standard – 232
I^2C:	Inter-Integrated Circuit
GYR:	Gyroscope
ACC:	Acceleration
MCU:	Microcontroller Unit

1. INTRODUCTION

The motivation of the project is to design a low cost and real-time system which estimates the position and orientation of the object by using an inertial measurement unit (IMU). Most of the previous works were to determine just the orientation of the object using the same sensor. Also, most of the previous works were made by using an Arduino based system which uses an 8-bit processor and often not fast enough to use in real-time systems. In this study STM32F103 series, 32-bit ARM-based microprocessor which supports 72 MHz clock speed internally is used. Position estimation technologies are based on global positioning system (GPS) work outdoor and, when used in prediction of close distance motions they are not sufficiently precise if not supported by some other means of sensor systems. In indoor applications they often fail due to multiple signal reflections. Also, it is more expensive comparing to the MPU6050.

MEMS devices become more popular every passing day. One of the greatest gifts of this technology is the MPU6050 sensor. MPU6050 is a sensor that contains the accelerometer, gyroscope, and magnetometer on a single chip. In our study, the magnetometer is not used. Because the electromagnetic interference issues are not considered in this study, robot electrical and electronic driving system's effects are unknown for the magnetometer measurements. Therefore, its use is avoided. Because of the nature of the sensor, sensor fusion and drift in both position and orientation are the major problems that are solved and explained in detail. Another problem that needs to be paid attention is the gimbal lock problem. Meanings of these terms can be briefly explained like at the end of the motion, the desired position and orientation values go rapidly far away. Using quaternions and Kalman filter technique drift in orientation is eliminated. Zero update and position update algorithms are used to eliminate drift with respect to position. In order to solve the gimbal lock problem again, quaternions are used.

The design has some similarities comparing to the existing design, but the key point is at the differences. Most of the existing designs are using Kalman filter and quaternions in order to solve problems that are mentioned above briefly and explained in detail later on. The difference is at the position calculation, a special algorithm whose name is position update is used in order to estimate the position as desired.

2. REQUIREMENTS SPECIFICATION

Some requirements specifications are described below in details. All the values described below are tested for the best results. Also, advantages and cost are described as well.

- **Performance and Functionality Requirements Specifications**

  In order to make a good estimation, the system is designed as real-time as possible. Also, the robot can be used in any kind of area or surface. The other advantage that the system provides from the point of view of energy consumption.

  -Processor must read data from the sensor as fast as possible.
  -Robot must not have any connection. (Wifi, Bluetooth etc.)
  -Robot must be used in everywhere and every surface. (Indoor, Outdoor etc.)
  -With full charge of batteries, the robot must be run as long as possible.

- **Economic Requirements Specifications**

  The robot provides a low-cost solution for the position estimation operations.

-STM32F103 is 21₺ which is approximately 3.95€ and can be easily found on internet.
-MPU 6050 is 10₺ which is approximately 1.9€ and can be easily found on internet.
-Two step motors are 200₺ which is approximately 37.60€ and can be easily found on internet.
-Mechanical components are 100₺ which is approximately 18.80€.
-Two motor drivers are 10₺ which is approximately 1.88€.
-OLED displayer is 25₺ which is approximately 4.70€.
-Three batteries are 60₺ which is approximately 11.30€.
-The battery holder is 20₺ which is approximately 3.75€.
The LM7805 voltage regulator is 1₺ which is approximately 0.20€.

- **Health and Safety Requirements Specifications**

  The system doesn’t have any danger for health and safety.

- **Manufacturability and Maintainability Requirements Specifications**

  The robot can be manufactured easily and because of its structure, it is easy to maintain.

  -All the electronic components must be soldered on a PCB with proper sockets.
  -If any of the components fail for some reasons, it must be easy to replace it with the new one.
  -All the components used in this project must be easy to find via internet.
  -From the point of view of the software, it must be easy to add and debug new code lines according to requirements.

3. STANDARDS

- **IEC 60228** (*International Electrotechnical Commission's international standard on conductors of insulated cables*.): All cables used in our design have been chosen in accordance with IEC 60228 standards.

- **IEEE 754-2008** (*IEEE Standard for Floating-Point Arithmetic*): Distance values obtained from our sensors have been represented in binary according to double-precision floating-point format as given in IEEE 754-2008 standards.

- **IEEE 1666-2005** *(IEEE Standard System C Language Reference Manual)*: Program of STM32F103 has been written in C language as given in IEEE 1666-2005 standards.

- **RS-232** *(Recommended Standard 232 for Serial Communication)*: Communication between STM32F103 and PC has been provided as given in RS-232 standards.

- **I^2C** *(Inter-Integrated Circuit)*: Communication between STM32F103 and MPU6050, and between STM32F103 and OLED display have been provided as given in I^2C standards.

4. PATENTS

- **System for correcting GPS position by system state estimation, Patent No: WO2009088215A2**

  The present invention provides a system for correcting a GPS position in that a GPS datum is corrected by determining the displacement resulting from vibrations and dynamic loads measured at a GPS positioning device attached to a structure such as a high-rise building, a bridge, a dam, a port and so forth. More particularly, the present invention relates to the system for correcting a GPS position by system state estimation, in which vibrations are measured via an input 3-axis accelerometer and an output accelerometer when surveying the structure, and the data is processed in real time with state equations preset using transfer functions with the input acceleration data and output accelerating datum measured on the two accelerometers, and then the modeling is updated with a model update algorithm such that the GPS data-converged values and real values are converged. Thus, more accurate GPS position data is calculated, the data processing speed can be shortened by real-time data processing, and disturbance-caused noise can be eliminated.

- **Attitude estimation for pedestrian navigation using low-cost mems accelerometer in mobile applications, and processing methods, apparatus, and systems, Patent No: US8694251B2**

  A user-heading determining system for pedestrian use includes a multiple-axis accelerometer having acceleration sensors; a device-heading sensor circuit physically situated in a fixed relationship to the accelerometer; an electronic circuit operable to generate signals representing components of acceleration sensed by the accelerometer sensors, and to electronically process at least some part of the signals to produce an estimation of attitude of a user motion with respect to the accelerometer, and further to combine the attitude estimation with a device heading

estimation responsive to the device-heading sensor circuit, to produce a user heading estimation; and an electronic display responsive to the electronic circuit to display information at least in part based on the user heading estimation. Other systems, circuits and processes are also disclosed.

- **Method for accelerometer-assisted navigation, Patent No: US9733089B2**

  A method for primarily sensor-based navigation includes: in a first time period, collecting geophysical position data using a GPS receiver of a navigation device; in the first time period, collecting a first set of accelerometer data using an accelerometer of the navigation device; analyzing the first set of accelerometer data to produce a first set of vertical vehicular motion data; generating a mapping association between the first set of vertical vehicular motion data and the geophysical position data; in a second time period after the first time period, collecting a second set of accelerometer data using the accelerometer; analyzing the second set of accelerometer data to produce a second set of vertical vehicular motion data; and calculating an estimated location of the vehicle by analyzing the second set of vertical vehicular motion data in light of the mapping association.

- **Accelerometer using acceleration, Patent No: US8498793B1**

  An altimeter using speed, forward acceleration and yaw angle rate to measure altitude. The accuracy of the measurement may be improved by compensating for a position offset of an accelerometer; Kalman filtering with an external altitude measurement; and/or compensating for an accelerometer bias. The accelerometer bias may be calculated based on the last incline angle before a dormant time period and/or by Kalman filtering.

- **Step length estimation method and portable terminal for the same, Patent No: US8280678B2**

  An apparatus and method for estimating a step length of a user are provided. The apparatus and method use a step length estimation algorithm, e.g. a step length estimation parameter coefficient, according to a movement state of a user, i.e. whether the movement state is a walking state or a running state. The movement state of the user is determined on the basis of an acceleration variance value of an

acceleration signal output from an accelerometer. Accordingly, the apparatus and method prevent errors in step length determinations.

- **Method and apparatus for measuring the speed of moving body using an accelerometer, Patent No: US6928382B2**

  Disclosed are an apparatus and a method for measuring the speed of a moving body using an accelerometer. A value of earth's gravitational acceleration component is detected from the measurements from the accelerometer and is removed from the acceleration value. Then, the acceleration value, after removal of the earth's gravitational acceleration component, is used to obtain the speed of the moving body. The earth's gravitational acceleration component is detected using a movement average at a point in time when the speed of the moving body is to be measured. By using the magnitude of a difference between a value obtained by removing an x-axis movement average from an x-axis measurement from the two-axis accelerometer and another value obtained by removing a y-axis movement average from a y-axis measurement from the two-axis accelerometer, it is possible to regulate a window for calculating the movement average and a weight value to each of the measurements included in the window. The present invention saves cost by reducing the number of expensive gyroscopes in implementing a speed measuring apparatus for vehicles. In addition, the speed measuring apparatus designed according to the present invention is superior to that designed according to the prior art in performance and improves the accuracy of position estimation in position estimation apparatuses for vehicles.

5. THEORETICAL BACKGROUND

5.1 Kalman Filter

Kalman filter is a common method using for state estimations in many areas from engineering to finance. MEMS IMUs produce various kinds of noise and errors. Accelerometers are too sensitive to any kind of move either small changes or big. Oppositely, gyroscopes are not sensitive to slow changing bias. As a result, accelerometers have poor dynamic features and gyroscopes have poor static features. Therefore, in order

to compensate the error from both accelerometer side and the gyroscope side an AHRS (Attitude and Heading Reference System) algorithm is needed. The most common AHRS algorithms are the complementary filter and the Kalman filter. A complementary filter uses an analysis in the frequency domain of the signals to combine them to get a better estimation. If the signals are affected by noises with different frequency, two filters, with an appropriate bandwidth, can be applied such that the sum of the two filtered signals cover the full range of useful frequency. For attitude estimation from IMU readings, a complementary filter performs high-pass filtering on the orientation estimated from gyroscope data affected by low-frequency noise, and low-pass filter on accelerometer data affected by high-frequency noise. The reason that avoidance of using complementary filter in this project is, complementary filter has constant gain. With true values constant gain is a good solution for static systems. In this project, Kalman filter is preferred. Because comparing to Kalman filter and complementary filter, Kalman filter's accuracy is better than the complementary filter and AHRS is a dynamic system and Kalman filter gives more accurate values in dynamic systems. From the point of view of implementation, Kalman filter a bit harder to implement comparing to complementary filter.

5.1.1 Application of Kalman Filter

As mentioned above, because of nature of the MPU 6050 the gyro and accelerometer data drifts by the time. The graphic below shows that the number of drifts. According to the graphic below, the pitch angle drifts 40 degrees from the desired state. In order to compensate these drifts and make a better estimation, the Kalman filter is a useful algorithm that takes a series of measurements. Measurements have the statistical noise and uncertainties over time. The basic working principle of the Kalman filter is to make estimations about the current state using the data from the previous state. Using data from both accelerometer and the gyroscope.

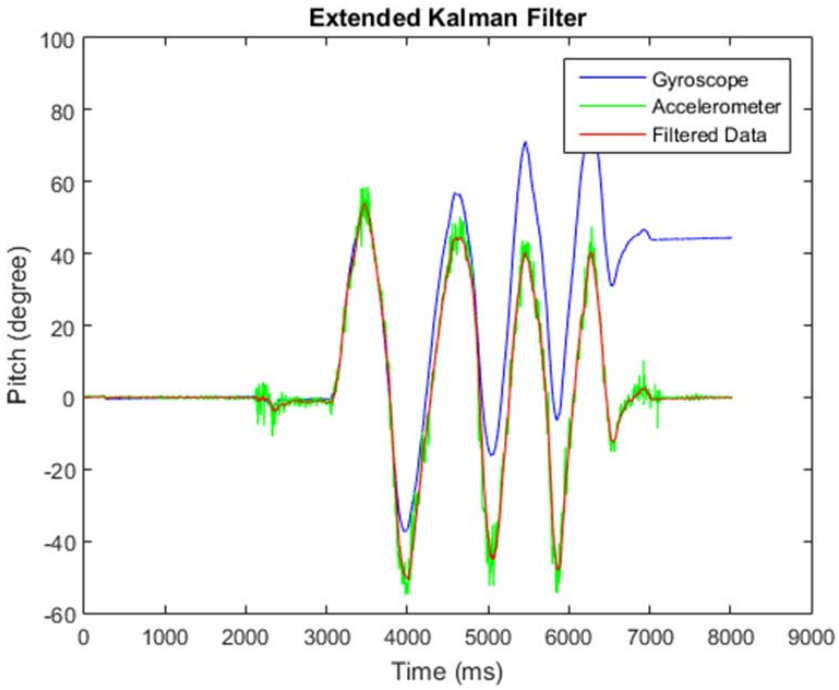

Figure 1: Results After Filter

In order to design a Kalman filter, two major steps must be considered. The prediction step and the update step.

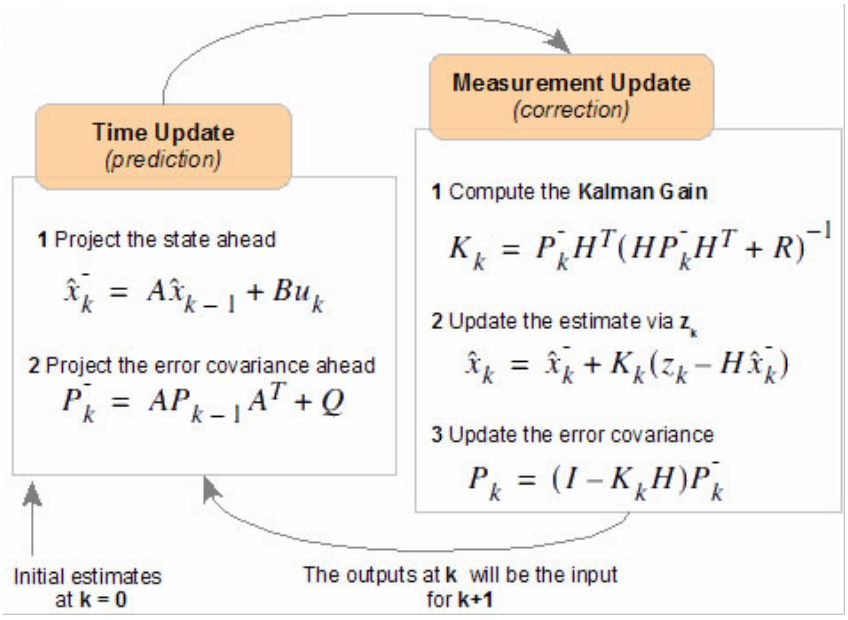

Figure 2: Prediction and Correction Equations of Kalman Filter [1]

In the prediction step, the Kalman filter tries to estimate the current state variables along with their uncertainties. In the update state, Kalman filter takes the measurements, compares it with the prediction and updates the estimate with Kalman gain.

The state space model of an inertial system is,

$$\theta k = \theta k{-}1 + \omega k \Delta t \tag{1}$$

$$zk = ak \tag{2}$$

Where θk is the estimation at time k, ωk is the rotation rate of the gyroscope, ak is the angle calculated from accelerometer data and zk is the measurement at time k.

The state vectors become,

$$x = \theta, \quad u = \omega \tag{3}$$

and the matrices

$$A = 1, \quad B = \Delta t, \quad H = 1, \quad K = K0 \tag{4}$$

So, the Kalman equations read

$$\boldsymbol{\theta}\hat{}k - = \boldsymbol{\theta}\hat{}k{-}1 + \omega k \Delta t \tag{5}$$

$$\boldsymbol{\theta}\hat{}k = (1{-}K0)\hat{}k - + K0ak \tag{6}$$

These equations can be reformatted as

$$\boldsymbol{\theta}\hat{}k = \alpha\boldsymbol{\theta}\hat{}k{-}1 + (1{-}\alpha)ak + \alpha\omega k \Delta t \tag{7}$$

where

$$\alpha = 1{-}K0 \tag{8}$$

Even if the linear Kalman filter one of the common method, it does not give very good results. The reason behind this is, the MPU 6050 ’s noise characteristics are non-linear and non-Gaussian. Because of this, the linear Kalman filter with the assumptions of Gaussian noise is not the best solution. The solution is using other Kalman filter methods. The other methods are again based on the theory explained above but with fewer assumptions.

5.1.2 Extended Kalman Filter

Extended Kalman Filter is again an estimator which estimates the current state from the previous state data but using non-linear noise characteristic which is similar to the MPU 6050‘s noise characteristics. The Kalman filter assumes a system of linear functions. The linear transformation of a normally distributed variable is normally distributed again. This is not valid for non-linear systems. In the real world, almost every system is non-linear. In order to handle non-linear systems, the Kalman filter is extended by approximating non-linearities with Taylor Series Expansion. A first-order expansion is often sufficient. In this project, for a better orientation estimation extended Kalman filter is used.

Consider a non-linear system, system transition,

$$x(k+1) = f[k,x(k),u(k)] + v(k) \tag{9}$$

Measurement equation,

$$z(k) = h[k,x(k)] + w(k) \tag{10}$$

Then the equations to estimate and update the state are:

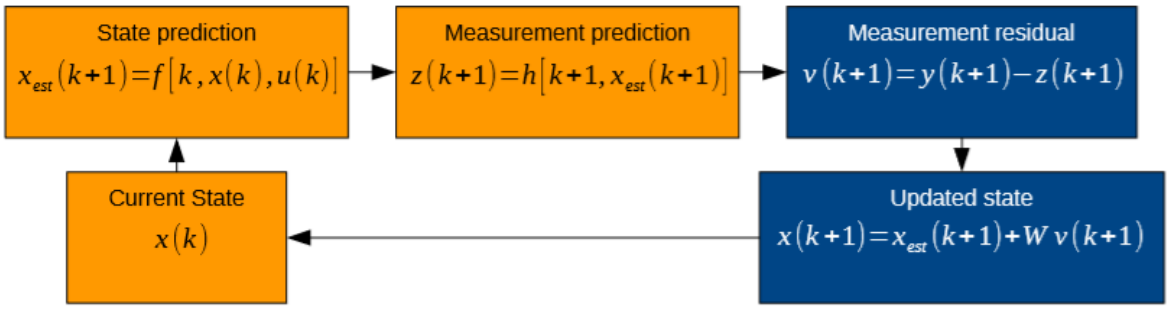

Figure 3: Prediction and Correction Equations of Extended Kalman Filter [2]

The first order Taylor Series Expansion is equivalent to the evaluation of the Jacobians of the functions f and h at a certain point x,

$$F(k)=\partial f[k,x(k),u(k)]/\partial x \,|x=x(k) \quad (11)$$

$$H(k+1)=\partial h[k+1,x(k+1)]\,/\partial x \,|x=xest(k+1) \quad (12)$$

To estimate and update the state covariance is similar to the Kalman filter,

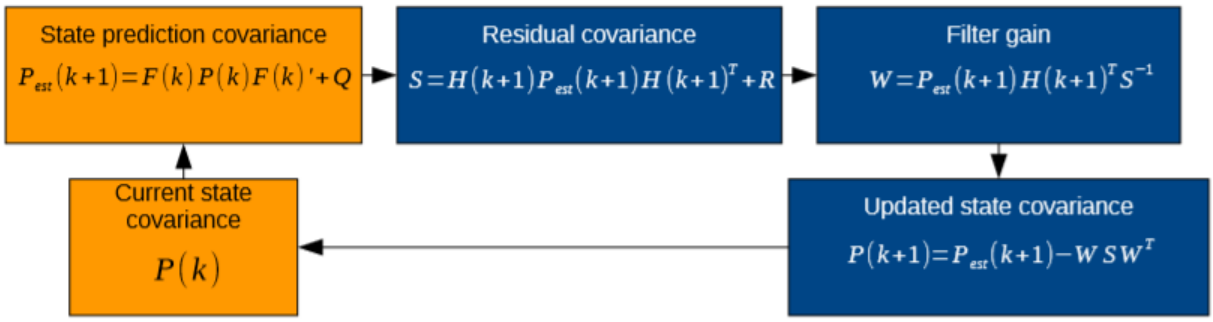

Figure 4: Covariance Estimation Equations [2]

5.2 Quaternions

A quaternion is a four-dimensional complex number that can be used to represent the orientation of a rigid body or coordinate frame in three-dimensional space. An arbitrary orientation of frame *B* relative to frame *A* can be achieved through a rotation of angle θ around an axis ${}^{A}\hat{r}$ defined in frame *A*. This is represented graphically in figure 5 where the mutually orthogonal unit vectors $\hat{x}_A$, $\hat{y}_A$ and $\hat{z}_A$, and $\hat{x}_B$, $\hat{y}_B$ and $\hat{z}_B$ define the principal axis of coordinate frames *A* and *B* respectively. The quaternion describing this orientation, ${}^{A}_{B}\hat{q}$, is defined by equation (1) where r_x, r_y and r_z define the components of the

unit vector ${}^{A}\hat{r}$ in the *x*, *y* and *z*-axes of frame *A* respectively. A notation system of leading super-scripts and sub-scripts adopted from Craig [4] is used to denote the relative frames of orientations and vectors. A leading sub-script denotes the frame being described and a leading super-script denotes the frame this is with reference to. For example, ${}^{A}_{B}\hat{q}$ describes the orientation of frame *B* relative to frame *A* and ${}^{A}\hat{r}$ is a vector described in frame *A*. Quaternion arithmetic often requires that a quaternion describing an orientation is first normalized. It is therefore conventional for all quaternions describing an orientation to be of unit length.

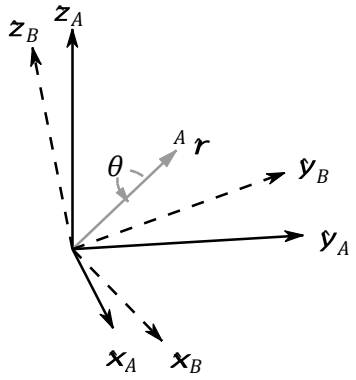

Figure 5: The orientation of frame B is achieved by a rotation, from alignment with frame A, of angle θ around the axis Ar.

Figure 1: The orientation of frame *B* is achieved by a rotation, from alignment with frame *A*, of angle θ around the axis ${}^{A}r$.

$$ {}^{A}_{B}\hat{\boldsymbol{q}} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} & -r_x sin\frac{\theta}{2} & -r_y sin\frac{\theta}{2} & -r_z sin\frac{\theta}{2} \end{bmatrix} \quad (13) $$

The quaternion conjugate, denoted by $*$, can be used to swap the relative frames described by an orientation. For example, ${}^{B}_{A}\hat{q}$ is the conjugate of ${}^{A}_{B}\hat{q}$ and describes the orientation of frame *A* relative to frame *B*. The conjugate of ${}^{A}_{B}\hat{q}$ is defined by equation (14).

$$ {}^{A}_{B}\hat{\boldsymbol{q}}^{*} = {}^{B}_{A}\hat{\boldsymbol{q}} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (14) $$

The quaternion product, denoted by $\otimes$, can be used to define compound orientations. For example, for two orientations described by ${}^{A}_{B}\hat{q}$ and ${}^{B}_{C}\hat{q}$, the compounded orientation ${}^{A}_{C}\hat{q}$ can be defined by equation (15).

$$ {}^{A}_{C}\hat{\boldsymbol{q}} = {}^{B}_{C}\hat{\boldsymbol{q}} \otimes {}^{A}_{B}\hat{q} \tag{15} $$

For two quaternions, *a* and *b*, the quaternion product can be determined using the Hamilton rule and defined as equation (15). A quaternion product is not commutative; that is,

$a \otimes b \neq b \otimes a$.

$$ \begin{aligned} \boldsymbol{a} \otimes \boldsymbol{b} &= \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \end{bmatrix} \otimes \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix} \\ &= \begin{bmatrix} a_0 b_0 - a_1 b_1 - a_2 b_2 - a_3 b_3 \\ a_0 b_1 + a_1 b_0 + a_2 b_3 - a_3 b_2 \\ a_0 b_2 - a_1 b_3 + a_2 b_0 + a_3 b_1 \\ a_0 b_3 + a_1 b_2 - a_2 b_1 + a_3 b_0 \end{bmatrix}^T \end{aligned} \tag{16} $$

A three-dimensional vector can be rotated by a quaternion using the relationship described in equation (16) [5]. ${}^{A}v$ and ${}^{B}v$ are the same vectors described in frame *A* and frame *B* respectively where each vector contains a 0 inserted as the first element to make them 4 element row vectors.

$$ {}^{B}\boldsymbol{v} = {}^{A}_{B}\hat{\boldsymbol{q}} \otimes {}^{A}\boldsymbol{v} \otimes {}^{A}_{B}\hat{\boldsymbol{q}}^{*} \tag{17} $$

The orientation described by ${}^{A}_{B}\hat{q}$ can be represented as the rotation matrix ${}^{A}_{B}R$ defined by equation (17) [5].

$$ {}^{A}_{B}\boldsymbol{R} = \begin{bmatrix} 2q_0^2 - 1 + 2q_1^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2q_0^2 - 1 + 2q_2^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 2q_0^2 - 1 + 2q_3^2 \end{bmatrix} \tag{18} $$

A quaternion may be obtained from a rotation matrix using the inverse of the relationships defined in (17); however, in some practical applications an available rotation matrix may not be orthogonal and so a more robust method is preferred. Bar-Itzhack provides a method [6] to extract the optimal, 'best fit' quaternion from an imprecise and non-orthogonal rotation matrix. The method requires the construction of the symmetric 4 by 4 matrix *K* (equation (18)) where r_{mn} corresponds to the element of the m^{th} row and n^{th} column of ${}^{A}_{B}R$. The optimal quaternion, ${}^{A}_{B}\hat{q}$, is found as the normalized Eigenvector corresponding to the maximum Eigenvalue of *K*. This is defined by equation (19) where v_0 to v_3 define the elements of the normalized Eigenvector.

$$ = \frac{1}{3}\begin{bmatrix} r_{11} - r_{22} - r_{33} & r_{21} + r_{12} & r_{31} + r_{13} & r_{23} - r_{32} \\ r_{21} + r_{12} & r_{22} - r_{11} - r_{33} & r_{32} + r_{23} & r_{31} - r_{13} \\ r_{31} + r_{13} & r_{32} + r_{23} & r_{33} - r_{11} - r_{22} & r_{12} - r_{21} \\ r_{23} - r_{32} & r_{31} - r_{13} & r_{12} - r_{21} & r_{11} + r_{22} + r_{33} \end{bmatrix} $$

$$ K \tag{7} $$

$$ {}^{A}_{B}\hat{\boldsymbol{q}} = \begin{bmatrix} v_3 & v_0 & v_1 & v_2 \end{bmatrix} \tag{19} $$

The ZYX Euler angles φ, θ and ψ describe an orientation of frame *B* achieved by the sequential rotations, from alignment with frame *A*, of ψ around $\hat{z}_B$, θ around $\hat{y}_B$, and φ around $\hat{x}_B$. This Euler angle representation of ${}^{A}_{B}\hat{q}$ can be calculated [7] using equations (20) to (22).

$$ \phi = \operatorname{atan2}\left(2(q_2q_3 - q_0q_1), 2q_0^2 - 1 + 2q_3^2\right) \tag{20} $$

$$ \theta = -\arctan\left(\frac{2(q_1q_3 + q_0q_2)}{\sqrt{1 - (2q_1q_3 + 2q_0q_2)^2}}\right) \tag{21} $$

$$ \psi = \operatorname{atan2}\left(2(q_1q_2 - q_0q_3), 2q_0^2 - 1 + 2q_1^2\right) \tag{22} $$

5.3 Gimbal Lock

Gimbal lock is the loss of one degree of freedom in a three-dimensional, three-gimbal mechanism that occurs when the axes of two of the three gimbals are driven into a parallel configuration, "locking" the system into the rotation in a degenerate two-dimensional space.

The word lock is misleading: no gimbal is restrained. All three gimbals can still rotate freely about their respective axes of suspension. Nevertheless, because of the parallel orientation of two of the gimbals' axes, there is no gimbal available to accommodate rotation along one axis.

5.3.1 Gimbals

A gimbal is a ring that is suspended so it can rotate about an axis. Gimbals are typically nested one within another to accommodate rotation about multiple axes.

They appear in gyroscopes and in inertial measurement units to allow the inner gimbal's orientation to remain fixed while the outer gimbal suspension assumes any orientation. In compasses and flywheel energy storage mechanisms they allow objects to remain upright. They are used to orient thrusters on rockets.

Some coordinate systems in mathematics behave as if there were real gimbals used to measure the angles, notably Euler angles.

For cases of three or fewer nested gimbals, gimbal lock inevitably occurs at some point in the system due to properties of covering spaces (described below).

5.3.2 Gimbals in Engineering

While only two specific orientations produce exact gimbal lock, practical mechanical gimbals encounter difficulties near those orientations. When a set of gimbals is close to the locked configuration, small rotations of the gimbal platform require large

motions of the surrounding gimbals. Although the ratio is infinite only at the point of gimbal lock, the practical speed and acceleration limits of the gimbals—due to inertia (resulting from the mass of each gimbal ring), bearing friction, the flow resistance of air or other fluid surrounding the gimbals (if they are not in a vacuum), and other physical and engineering factors—limit the motion of the platform close to that point.

Consider a case of a level sensing platform on an aircraft flying due north with its three gimbal axes mutually perpendicular (i.e., roll, pitch, and yaw angles each zero). If the aircraft pitches up 90 degrees, the aircraft and platform's yaw axis gimbal become parallel to the roll axis gimbal, and changes about yaw can no longer be compensated for.

5.3.3 Preventing Gimbal Lock

This problem may be overcome by use of a fourth gimbal, intelligently driven by a motor so as to maintain a large angle between roll and yaw gimbal axes. Another solution is to rotate one or more of the gimbals to an arbitrary position when gimbal lock is detected and thus reset the device.

Modern practice is to avoid the use of gimbals entirely. In the context of inertial navigation systems, that can be done by mounting the inertial sensors directly to the body of the vehicle (this is called a strap-down system) and integrating sensed rotation and acceleration digitally using quaternion methods to derive vehicle orientation and velocity. Another way to replace gimbals is to use fluid bearings or a flotation chamber.

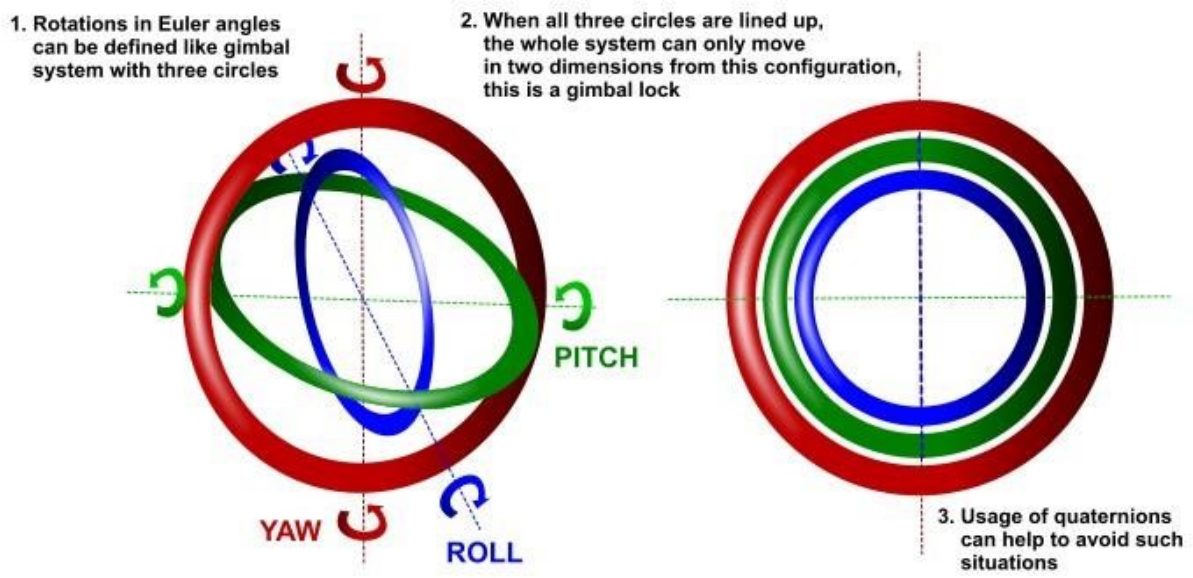

Figure 6: Gimbal Lock Problem[8]

6. METHODOLOGY

In this title, hardware, software, and tools that are used in this project are described detailed.

6.1 System Hardware

6.1.1 Motor Driver Circuit (A3964)

Dual full-bridge PWM motor driver, designed for pulse width modulated (PWM) current control of bipolar stepper motors, the A3964 is capable of continuous output currents to ±800 mA and operating voltages to 30 V. Internal fixed off-time PWM current-control circuitry can be used to regulate the maximum load current to the desired value. An internal precision voltage reference is provided to improve motor peak-current control accuracy. The peak load current limit is set by the user's selection of an external resistor divider and current-sensing resistors. The fixed off-time pulse duration is set by user-selected external RC timing networks. The capacitor in the RC timing network also determines a user-selectable blanking window that prevents false triggering of the PWM

current control circuitry during switching transitions. This eliminates the need for two external RC filter networks on the current-sensing comparator inputs. For each bridge, the PHASE input controls load current polarity by selecting the appropriate source and sink driver pair. For each bridge, the ENABLE input, when held high, disables the output drivers. Special power-up sequencing is not required. Internal circuit protection includes thermal shutdown with hysteresis, transient suppression diodes, and crossover-current protection [9].

Table 1: Absolute Maximum Ratings of A3964 [9]

Absolute Maximum Ratings

Characteristic	Symbol	Notes		Rating	Units
Load Supply Voltage	V_{BB}			33	V
Logic Supply Voltage	V_{CC}			7.0	V
Input Voltage	V_{IN}			–0.3 to V_{DD} + 0.3	V
Sense Voltage	V_S			1.0	V
Reference Output Current	$I_{REF(OUT)}$			1.0	mA
Output Current	I_{OUT}	Output current rating may be limited by duty cycle, ambient temperature, and heat sinking. Under any set of conditions, do not exceed the specified current rating or a junction temperature of 150°C.	t_w = 10 µs	±1.0	A
			Continuous	±800	mA
Package Power Dissipation	P_D	See graph		–	–
Operating Ambient Temperature	T_A	Range S		–20 to 85	°C
Maximum Junction Temperature	T_J(max)	Fault conditions that produce excessive junction temperature will activate the device's thermal shutdown circuitry. These conditions can be tolerated but should be avoided.		150	°C
Storage Temperature	T_{stg}			–55 to 150	°C

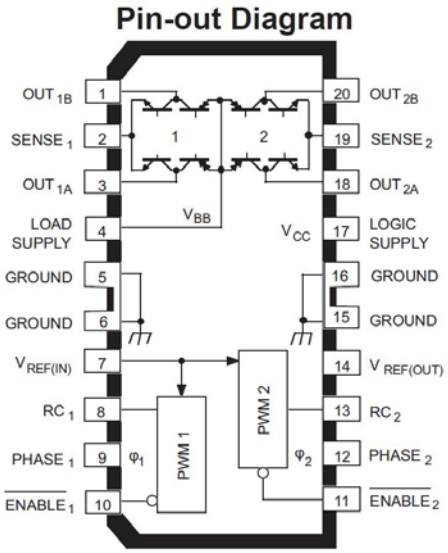

Figure 7: Pin-out Diagram of A3964 [9]

6.1.2 MCU STM32F103C8T6

The STM32F103xx medium-density performance line family incorporates the high-performance ARM®Cortex®-M3 32-bit RISC core operating at a 72 MHz frequency, high-speed embedded memories (Flash memory 64 Kbytes and SRAM 20 Kbytes), and an extensive range of enhanced I/Os and peripherals connected to two APB buses. All devices offer two 12-bit ADCs, three general purpose 16-bit timers plus one PWM timer, as well as standard and advanced communication interfaces: up to two I^2Cs and SPIs, three USARTs, a USB and a CAN.

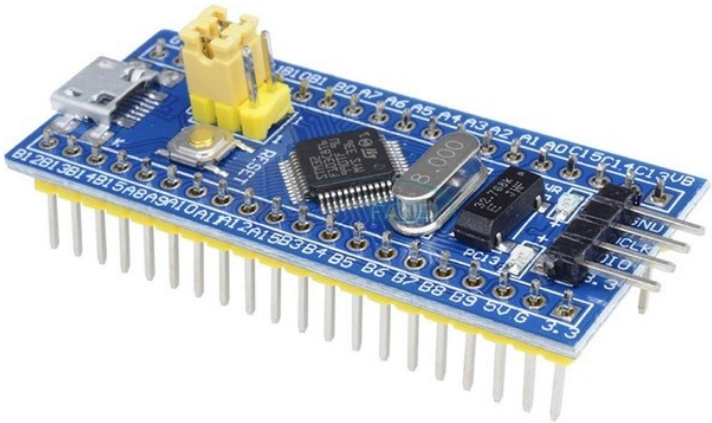

Figure 8: STM32F103C8T6 Board

STM32F103C8T6 has many advantages compared to Arduino Nano (ATMega328) or other compatible Arduino boards. Especially, STM has DMA (Direct Memory Accessing) feature which is used for communication with MPU6050 via I^2C. Communication speed is increased by using DMA. *Table 2* shows the basic differences between STM and Arduino.

Table 2: Differences between STM32F103C8t6 and Arduino Nano [10] [11]

	STM32F103C8T6	**Arduino Nano**
Bus width	32-bit	8-bit
Flash Memory	64 kB Flash, 20kB RAM	32 kB Flash, 2kB RAM
Speed	Up to 72 MHz with PLL	Up to 20 MHz
Architecture	ARM CORTEX-M3	ATMEL AVR
Timers	– Three 16-bit timers, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input – 16-bit, motor control PWM timer with deadtime generation and emergency stop – 2 watchdog timers (Independent and Window) – SysTick timer 24-bit down counter	– Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode – One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

A benchmark is shown in *Table3.*

Table 3: MCU Benchmark [12]

BENCHS (x30000)	**STM32F103C8T6 at 72MHz**	**Arduino Nano at 16MHz**
INT_LOOP	2924 us	7544 us
LONG_LOOP	2926 us	13408 us
FLOAT_DIV	27979 us	154792 us
DOUBLE_DIV	38000 us	154800 us
FLOAT_MUL	20463 us	156744 us
DOUBLE_MUL	25891 us	156736 us

6.1.3 MPU 6050

The MPU-6050 is a commonly used chip that combines a MEMS gyroscope and a MEMS accelerometer and uses a standard I²C bus for data transmission.

The MPU-60X0 is the world's first integrated 6-axis Motion Tracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor (DMP) all in a small 4x4x0.9mm package. With its dedicated I2C sensor bus, it directly accepts inputs from an external 3-axis compass to provide a complete 9-axis Motion Fusion output. The MPU-60X0 Motion Tracking device, with its 6-axis integration, on-board Motion Fusion, and run-time calibration firmware, enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices, guaranteeing optimal motion performance for consumers. The MPU-60X0 is also designed to interface with multiple no inertial digital sensors, such as pressure sensors, on its auxiliary I²C port.

The MPU-60X0 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADC for digitizing the accelerometer outputs. For precision tracking of both fast and slow motions, the parts feature a user-programmable gyroscope full-scale range of ±250, ±500, ±1000, and ±2000°/sec (DPS) and a user programmable accelerometer full-scale range of ±2g, ±4g, ±8g, and ±16g.

For power supply flexibility, the MPU-60X0 operates from VDD power supply voltage range of 2.375V-3.46V. Additionally, the MPU-6050 provides a VLOGIC reference pin (in addition to its analog supply pin: VDD), which sets the logic levels of its I2C interface. The VLOGIC voltage may be 1.8V±5% or VDD. [13]

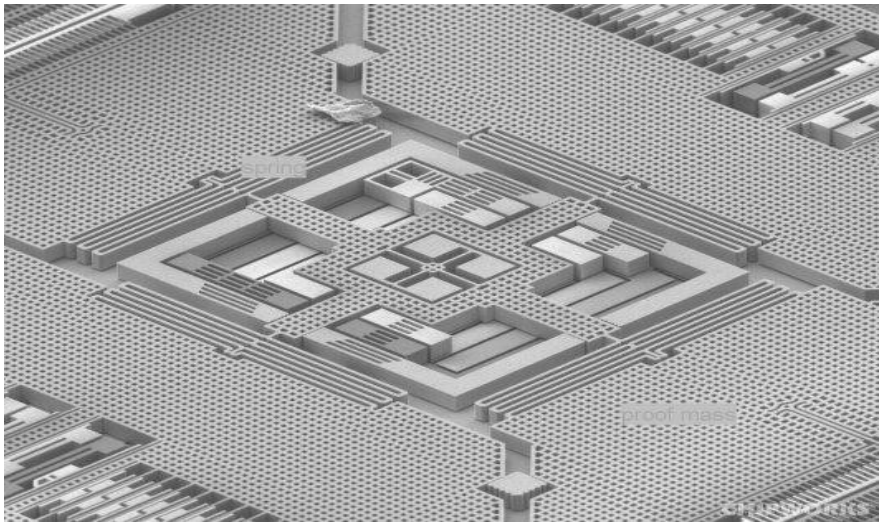

Figure 9: Structure of MEMS

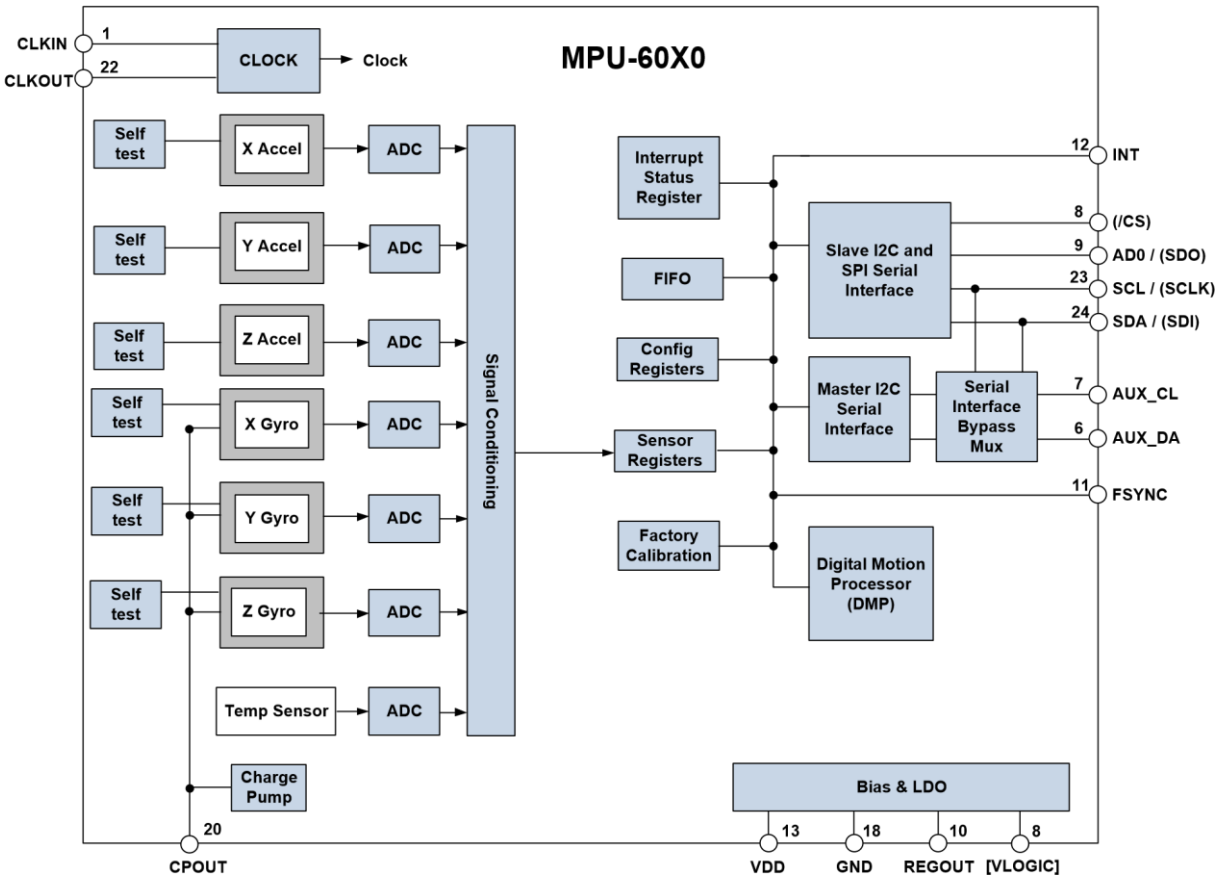

Figure 10: Block Diagram of MPU-6050

6.1.3.1 MEMS Technology

Micro-Electro-Mechanical Systems, or MEMS, is a technology that in its most general form can be defined as miniaturized mechanical and electro-mechanical elements (i.e., devices and structures) that are made using the techniques of microfabrication. The critical physical dimensions of MEMS devices can vary from well below one micron on the lower end of the dimensional spectrum, all the way to several millimeters. Likewise, the types of MEMS devices can vary from relatively simple structures having no moving elements, to extremely complex electromechanical systems with multiple moving elements under the control of integrated microelectronics. The one main criterion of MEMS is that there are at least some elements having some sort of mechanical functionality whether or not these elements can move. The term used to define MEMS varies in different parts of the world. In the United States, they are predominantly called MEMS, while in some other parts of the world they are called "Microsystems Technology" or "micromachined devices".

While the functional elements of MEMS are miniaturized structures, sensors, actuators, and microelectronics, the most notable (and perhaps most interesting) elements are the microsensors and microactuators. Microsensors and microactuators are appropriately categorized as "transducers", which are defined as devices that convert energy from one form to another. In the case of microsensors, the device typically converts a measured mechanical signal into an electrical signal.

6.1.3.2 Working Principles

Nanotechnology is the ability to manipulate matter at the atomic or molecular level to make something useful at the nano-dimensional scale. Basically, there are two approaches to implementation: the top-down and the bottom-up. In the top-down approach, devices and structures are made using many of the same techniques as used in MEMS except they are made smaller in size, usually by employing more advanced photolithography and etching methods. The bottom-up approach typically involves deposition, growing, or self-assembly technologies. The advantages of nano-dimensional devices over MEMS involve benefits mostly derived from the scaling laws, which can also present some challenges as well. [14]

6.1.3.3 Reason of Drift on MEMS

Inside of this little machine is all the elements necessary to demonstrate this on a tiny scale: a chip of silicon (the proof mass) suspended on miniature springs, with sets of interlaced fingers, or "capacitor plates". These fingers don't touch, however, the motion causes the micro-springs to compress or expand as the proof mass attempts to remain at rest, which then causes the plates to move closer together or further apart. These moves cause drift on MEMS even it's not moving.

6.1.3.4 Decision of Using MPU 6050

We have decided to go ahead with the project and construct your own self-localization robot Now we can start by looking at one of the most significant elements of this project; the sensor. Traditionally the sensor of preference for stabilization is a gyroscope. Nowadays gyroscopes are extremely small and very cheap to buy, so they are ideal for amateur electronics projects. Unfortunately, these gyroscopes (both the cheap and the not-so-cheap versions) also come with their own problems. They are good for short-term and quick movements but tend to drift over time as the error accumulates. They also record a lot of jitter and noise, which needs to be filtered by the microcontroller before the data can be used. [15]

To reduce this drifting effect of the gyroscope, it is possible to combine the sensor data with that from an accelerometer. The accelerometer is good at sensing slower and more prolonged movements, rather than the fast motion. Therefore, if we take the best of both worlds and fuse the data together, we will be left with an extremely accurate picture of the motion of the robot.

As a result, we decided to use a combined accelerometer & gyroscope breakout module (the MPU-6050), which is slightly more expensive than a simple gyro but should lead to a superior stabilization performance.

6.1.4 18650 Batteries

18650 batteries, large and small solar panels to the hair clipper from toys 18650 batteries are used in all electronic equipment, grow our distress every time that gives us energy and we need energy.

The structure of the 18650 is very similar to the battery we used today in our smartphones or laptops. Lithium-ion batteries, which basically generate electricity from ion movements between different materials, have a very practical use as they can be recharged repeatedly. This practicality with Lithium-ION batteries enables them to be used in virtually any electronic charging device, from mobile phones to laptops. The 18650 battery, which is very similar in structure to the Lithium-ION battery, is a frequently preferred model since it is extremely light at the same time.

The 18650 battery is similar in structure to the Lithium ION battery used in smartphones and laptops. In fact, these batteries, which are found in smartphones and computers and in other products, are extremely miserable as well as rechargeable. This is why they are often preferred in the electronics industry.

At the same time, famous companies that produce many electronic products, from telephones to computers to televisions to tablets, produce 18650 Batteries. Companies such as Sony, Samsung, LG, Panasonic manufacture 18650 Batteries is a sign that this industry is quite alive. So, the 18650 Battery is a product we use more often than you think.

Anyone who has heard the 18650 battery is convinced that the battery is at power 18650 mAh. In fact, the 18650 battery is an industrial rechargeable battery, and it also has battery dimensions. 18650 batteries; It has a diameter of 18 mm and a dimension of 65 mm. In fact, you can easily see that 18 and 65 in 18650 symbolize diameter and height. Your mind may come to the question of what the last "0" means. The 18650 Battery model has 18 diameters, 65 lengths and 0; batteries format icons. 0, 18650 indicates that the battery is a cylindrical battery.

The 18650 batteries can provide several volts of electricity depending on the quality. At the moment, the top quality 18650 battery provides an energy output of 3.7, 3.9 or 4.2 volts. Of course, this is the case for recognized and known 18650 battery

manufacturers. If you have a 18650 battery in average quality, your battery will provide you with an output of up to 4.2 volts.

6.1.5 OLED Display

This display is small, only about 1" diagonal, but very readable due to the high contrast of an OLED display. This display is made of 128x64 individual white OLED pixels, each one is turned on or off by the controller chip. Because the display makes its own light, no backlight is required. This reduces the power required to run the OLED and is why the display has such high contrast. I^2C interface is used for communication with MCU. Ultra-low power consumption, in full screen lit 0.08W.

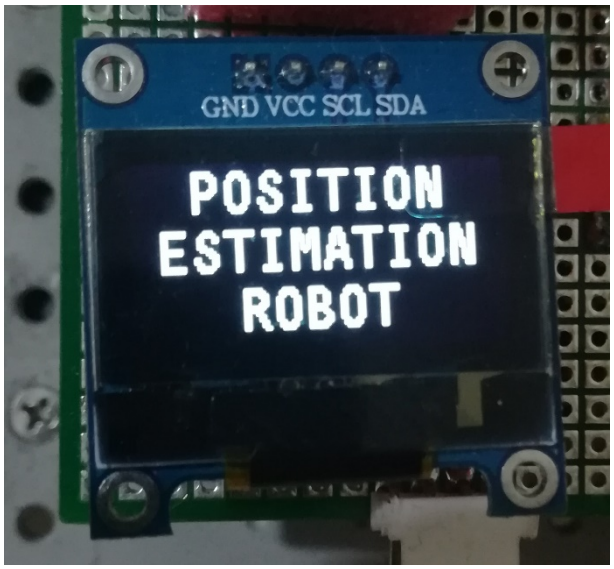

Figure 11: OLED Display

6.2 Software

6.2.1 Motor Drive

Robot calculates its position and orientation. Movement of the robot is done according to these calculated values. In this way, robot checks the movement whether it arrives the target position and orientation. Movement of the robot is provided with the motor drive

part in software. Motor drive part consists of 4 sub-functions. These functions are as follows:

- Forward: In this function, left and right motors are driven to forward direction at the same time.
- Backward: In this function, left and right motors are driven to backward direction at the same time.
- Left: In this function, left motor is driven to backward direction and right motor is driven to forward direction at the same time. Turning axis is the center point between left and right motors.
- Right: In this function, right motor is driven to backward direction and left motor is driven to forward direction at the same time. Turning axis is the center point between left and right motors.

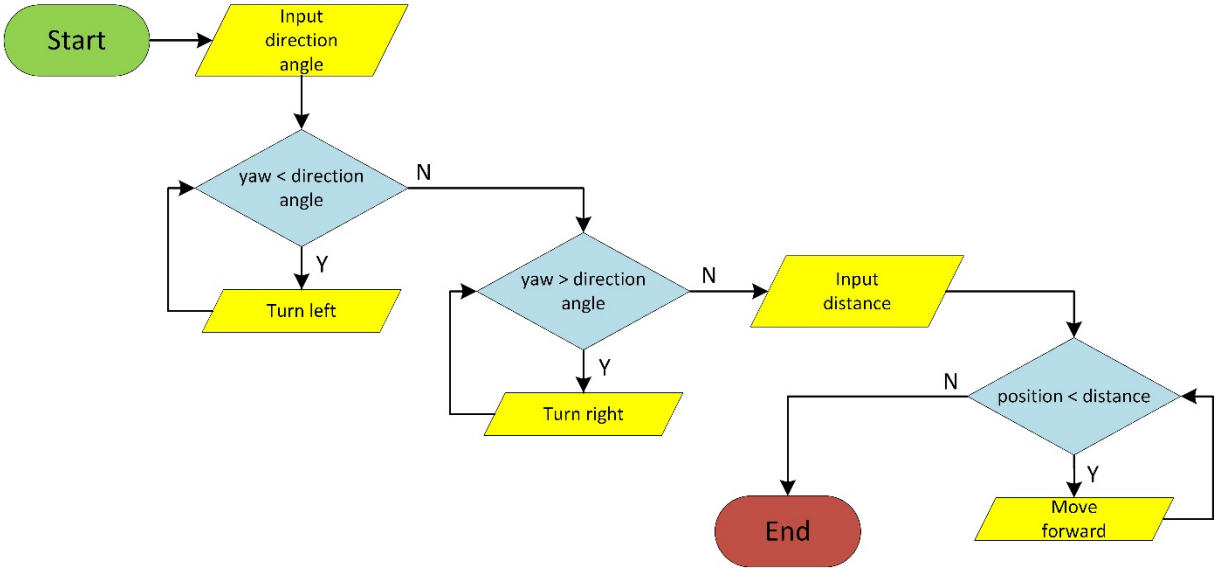

Figure 12: Algorithm of Motor Drive Function

Figure 12 shows the algorithm of the motor drive.

Doing optimization is necessary while driving a stepper motor. Current flows through driver circuit must not be very high. This high current can cause a damage to driver circuit and heat it. Temperature rising causes cooling problems. The energy

consumption of the motor goes up. Energy consumption is very important because battery pack should be used efficiently in a mobile robot. Necessary optimization must be done.

Table 4: Truth Table of A3964 [9]

ENABLE	**PHASE**	**OUT_A**	**OUT_B**
H	X	Off	Off
L	H	H	L
L	L	L	H

Table 4 shows the truth table of the motor driver circuit (A3964). Motor drive function is written according to this table. The driver circuit (A3964) is explained in hardware title detailed.

The speed of the motor depends on period of the E1 and E2 signal. Motor almost runs at maximum speed with period of 6 ms in *Figure 13.* Driving the motor in low speed causes noise on the stepper. Noise affects the accelerometer and gyroscope badly. In this case, filtering becomes harder. On the other hand, in high-speed driving, sensitivity of motion decreases, especially while turning left or right, it is very hard to fix the yaw angle. Speed of the motor should be arranged properly.

6.2.1.1 Control Signals in Forward and Backward Straight Motion

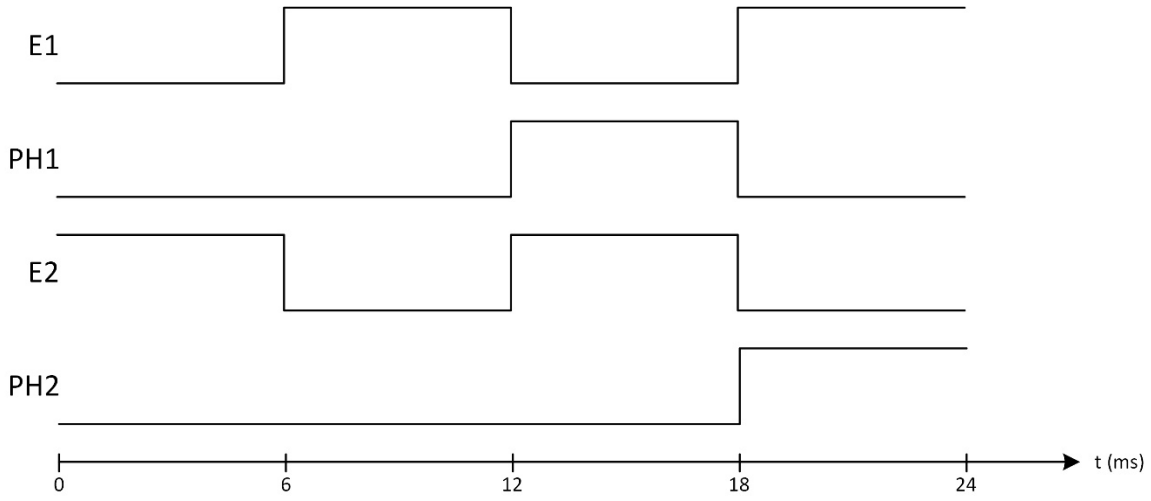

Figure 13: Control Signals of A3964 in Straight Motion

Signals in *Figure 13* are used while driving in forward or backward direction. In forward or backward motion, there is no turning motion and it is not necessary to fix yaw angle thus speed is not critical. Driving could be at high speed.

Driving motor like in *Figure 13* has no optimization and causes high current and heat on driver circuit. Energy consumption is very high. The advantage of this method is maximum torque gain, but maximum torque is not necessary for movement. Energy consumption becomes more critical in this case.

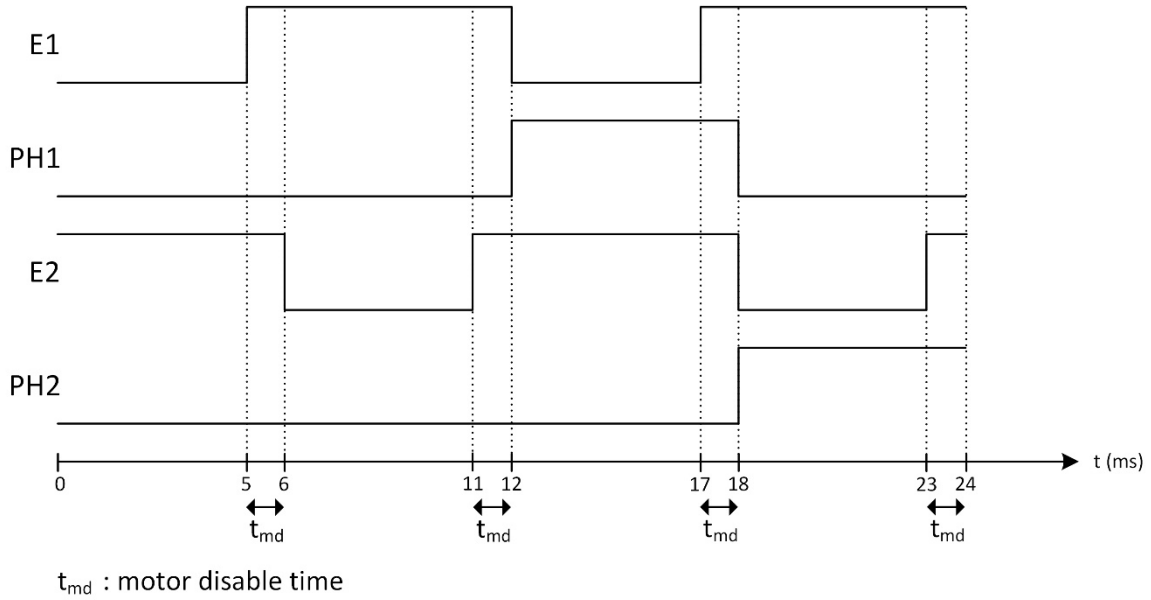

Figure 14: Control Signals of A3964 in Straight Motion After Optimization

After doing some optimization as in *Figure 14* current decreases and energy consumption goes down. Motors are disabled for 1 ms in every 6 ms period by changing the E1 and E2 signals. Torque is a little bit small, but the robot has the optimal torque to move properly. Speed does not change from *Figure 14* in this case.

6.2.1.2 Control Signals in Left and Right Turning Motion

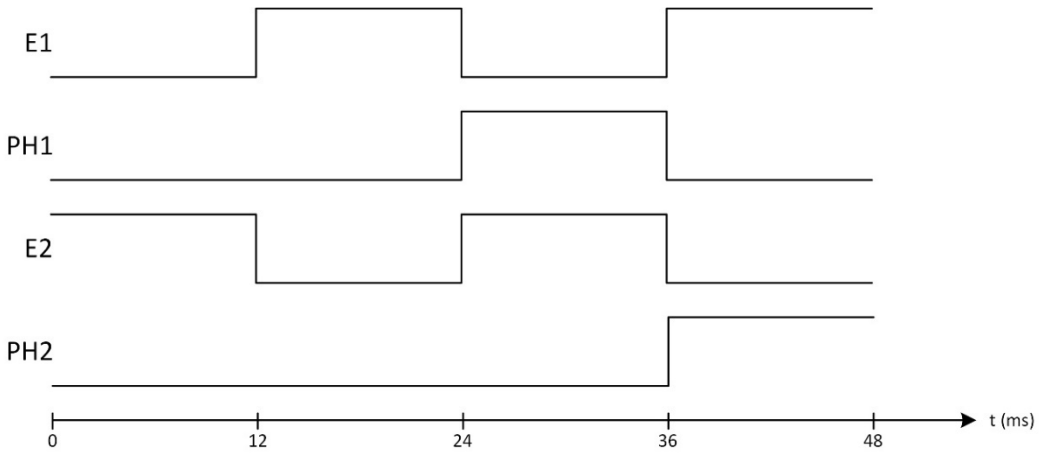

Figure 15: Control Signals of A3964 in Turning Motion

Signals in *Figure 15* are used while driving in left or right turning. In left or right motion, there is a turning motion and it is necessary to fix yaw angle thus speed becomes critical. Driving could not be at high speed. Speed is decreased by increasing the period of E1 and E2 signals from 12 ms to 24 ms.

Driving motor like in *Figure 15* also has no optimization and causes high current and heat on driver circuit. Energy consumption is very high. The advantage of this method is maximum torque gain, but maximum torque is not necessary for movement. Energy consumption becomes more critical in this case.

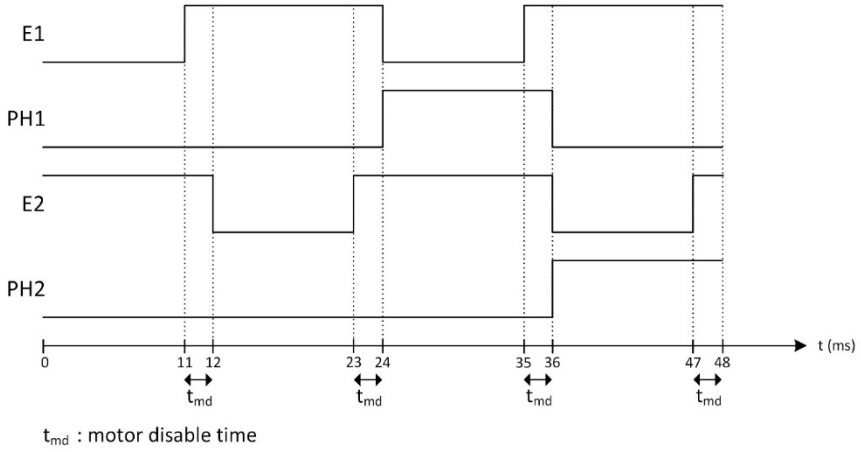

Figure 16: Control Signals of A3964 in Turning Motion After Optimization

After doing some optimization as in *Figure 16* current decreases and energy consumption goes down. Motors are disabled for 1 ms in every 12 ms period by changing the E1 and E2 signals. Torque is a little bit small, but robot has the optimal torque to move properly. Speed does not change from *Figure 16* in this case.

6.2.2 Trapezoidal Discrete Integration

The process of measuring the area under a function plotted on a graph is called integration. Accelerometers and gyroscope sensors are producing linear acceleration (m/s/s) and angular velocity(w/s), these values are required to take the time integral. In this way, time-independent (instant) orientation and position data can be obtained.

The data generated by the sensor in a discrete manner must be integrated cumulatively. When the integration method is selected, the calculation cost and numerical error rates are taken into account.

The sampling rate is more important than the complexity of the selected method. Since the MPU 6050 sensor operates at a constant 1 kHz sampling rate, it is important to be able to integrate each sample into the process without missing any of the data. In this way, rather than using more sophisticated and complex method, accepting more frequent

data with small errors will be more convenient. The creation time for each sample data set is fixed and equal to 1 millisecond unless the configuration is changed.

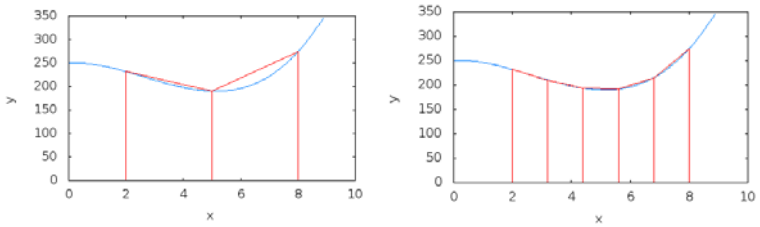

Figure 17: Low and High Sampling Rates [15]

The discrete integral is equal to the cumulative sum of the areas under each sample. However, if this method is applied directly, the computational errors will also be large and will increase with time. These errors are formally known as sampling losses. Since our sampling frequency is high, these errors will be smaller. But in order to further reduce the errors, first order Closed Newton-Cotes method (Trapezoidal rule) is used (table 5) [16].

Table 5: Closed Newton-Cotes method [16]

Open Newton–Cotes Formulae

Common name	step size	Formula	Error term	Degree
Rectangle rule, or midpoint rule	$\frac{b-a}{2}$	$(b-a)f_1$	$\frac{(b-a)^3}{24}f^{(2)}(\xi)$	2
Trapezoid method	$\frac{b-a}{3}$	$\frac{b-a}{2}(f_1+f_2)$	$\frac{(b-a)^3}{36}f^{(2)}(\xi)$	3
Milne's rule	$\frac{b-a}{4}$	$\frac{b-a}{3}(2f_1-f_2+2f_3)$	$\frac{7(b-a)^5}{23040}f^{(4)}(\xi)$	4
Willms' rule	$\frac{b-a}{5}$	$\frac{b-a}{24}(11f_1+f_2+f_3+11f_4)$	$\frac{19(b-a)^5}{90000}f^{(4)}(\xi)$	5

6.2.3 PC Data Recorder Software – RS 232

Instead of directly creating the algorithm with working on the robot, our algorithm was developed in PC software such as MATLAB. This made it easier for us to visualize what problems might be in which conditions. At the same time, it was quite easy to produce solutions to the problems which visually and numerically examined. For these features, it is first necessary to transfer the data of the MPU 6050 sensor directly to the pc.

The MPU 6050 sensor sends each data in 2 bytes packets to the STM32 microcontroller via I^2C.

Table 6: MPU data Registers [13]

Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT_H	R	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT_L	R	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT_H	R	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT_L	R	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT_H	R	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT_L	R	ACCEL_ZOUT[7:0]							
41	65	TEMP_OUT_H	R	TEMP_OUT[15:8]							
42	66	TEMP_OUT_L	R	TEMP_OUT[7:0]							
43	67	GYRO_XOUT_H	R	GYRO_XOUT[15:8]							
44	68	GYRO_XOUT_L	R	GYRO_XOUT[7:0]							
45	69	GYRO_YOUT_H	R	GYRO_YOUT[15:8]							
46	70	GYRO_YOUT_L	R	GYRO_YOUT[7:0]							
47	71	GYRO_ZOUT_H	R	GYRO_ZOUT[15:8]							
48	72	GYRO_ZOUT_L	R	GYRO_ZOUT[7:0]							

When trying to process these data on the processor and send it as a string, it is not possible to send every data at 1ms. Since the length of the data converted to a string is variable, it also causes timing problems. Moreover, conversion to string takes time. Instead of, the data received from the sensor is taken with DMA (Direct Memory Access) and the raw data is transferred to the PC via RS232 directly. In this way, possible to fetch data without losing data. However, we cannot use the carriage-return operations that exist in the string to specify the beginning of the data. To do this, we need to add a 2-byte header per data packet. A 1-byte counter to control data misses and lastly to see that each data packet was sent at a time of less than 1ms, elapsed time was added in a format divided by 100 for fit in a byte. The length of an entire data packet has reached 18 bytes.

```
0.  header 1 0x21
1.  header 2 0x0F
2.  counter
3.  accelX MSB
4.  accelX LSB
5.  accelY MSB
6.  accelY LSB
7.  accelZ MSB
8.  accelZ LSB
9.  temprature MSB
10. temprature LSB
11. gyroX MSB
12. gyroX LSB
13. gyroY MSB
14. gyroY LSB
15. gyroZ MSB
16. gyroL lSB
17. elsapesed time
```

Figure 18: Raw Data Package

HEADER	CNT	ACC X	ACC Y	ACC Z	TMP	GYR X	GYR Y	GYR Z	ELAPS T
543	0	-352	-640	15024	-3232	250	-1	-92	51
543	1	-248	-656	15092	-3232	249	2	-81	51
543	2	-272	-584	15268	-3232	246	-7	-80	51
543	3	-252	-576	15252	-3264	249	-14	-81	51
543	4	-136	-592	15514	-3232	247	-10	-91	51
543	5	-216	-516	15518	-3232	246	5	-93	51
543	6	-228	-564	15088	-3232	248	-6	-98	51
543	7	-288	-536	15180	-3232	247	-7	-510	51
543	8	-252	-592	15200	-3232	248	-8	-90	51
543	9	-296	-596	15132	-3232	257	-7	-91	51
543	51	-284	-564	15112	-3232	260	-12	-93	51
543	11	-144	-604	15020	-3280	260	-18	-87	51
543	12	-224	-616	15120	-3248	256	-11	-87	51
543	13	-384	-576	15140	-3248	259	-11	-86	51
543	1	-248	-656	15092	-3232	249	2	-81	51
543	2	-272	-584	15268	-3232	246	-7	-80	51
543	3	-252	-576	15252	-3264	249	-14	-81	51
543	4	-136	-592	15514	-3232	247	-10	-91	51
543	5	-216	-516	15518	-3232	246	5	-93	51

Figure 19: Stored Data on Txt File

There is no free software that can record at these speeds, combine the data and find the header and export it to the txt file. For this reason, it is necessary to write a pc software. The C# language is chosen for this job because it's easy and fast to implement it. In this interface, com port settings and txt recording button have been added. On the backplane, after the header found at the incoming data from the serial port, the data was assembled according to the sequence. The data converted to the string is stored in txt file for reading in MATLAB.

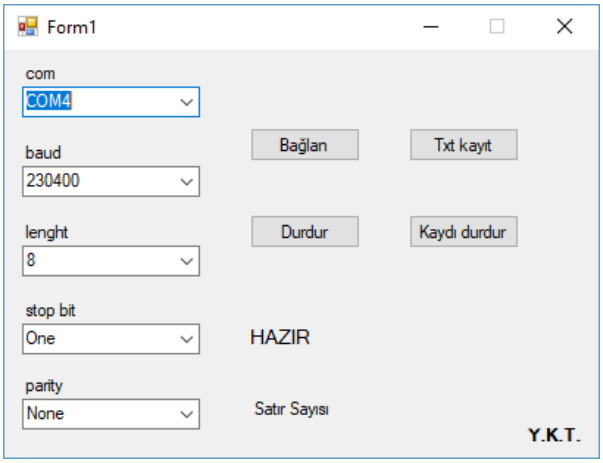

Figure 20: Data Recorder Interface

6.2.4 Motion Detection and Zero Update

One of the algorithms we use to remove the drift effect is to determine if it is moving and then make some adjustments accordingly. Acceleration data was used to determine the presence of motion, and this data was examined in a few different ways.

6.2.4.1 Total Unit Acceleration

The data from the accelerometer also contains gravity when it is raw. In this case, the vector sum of the acceleration data must be 1g (unity) as the scale, if no action is in question at that moment. If there is a linear acceleration in any axis, geometric the vector sum of the acceleration data will be more or less than 1g. It can be understood whether or not there is movement on this count.

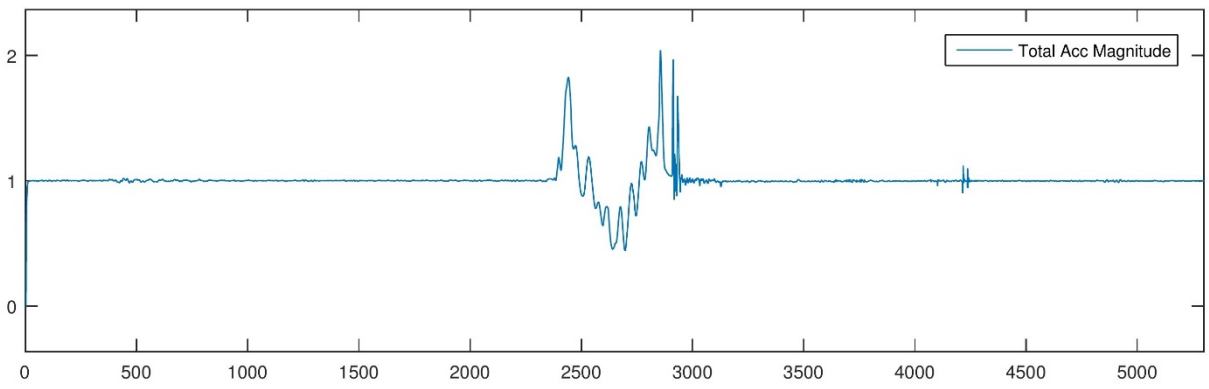

Figure 21: Total Acc Magnitude

But when it comes to an example that starts to move and continues to move at a constant speed, the unit acceleration will be the issue. Hence, it is not possible to trust this method alone.

6.2.4.2 Noise Differentiation

Using the accelerometer with the highest scale (more sensitive to movement) also brings a few advantages. One of these is the acceleration data which is changing even in the case of being fixed at hand. Even though it still gives a value of 0 on average, there is a noticeable difference on the amplitude and frequency between the noise standing on a hard floor and the noise we entered our hand between the sensor and same hard floor. Following this frequency and amplitude changes, it is quite accurate to follow if there is any movement.

The intersection of the outputs of these two motion detection algorithms gives us precise results. Detection of complete moments of inactivity may not be instantaneous in this way, but the moments detected are fairly true.

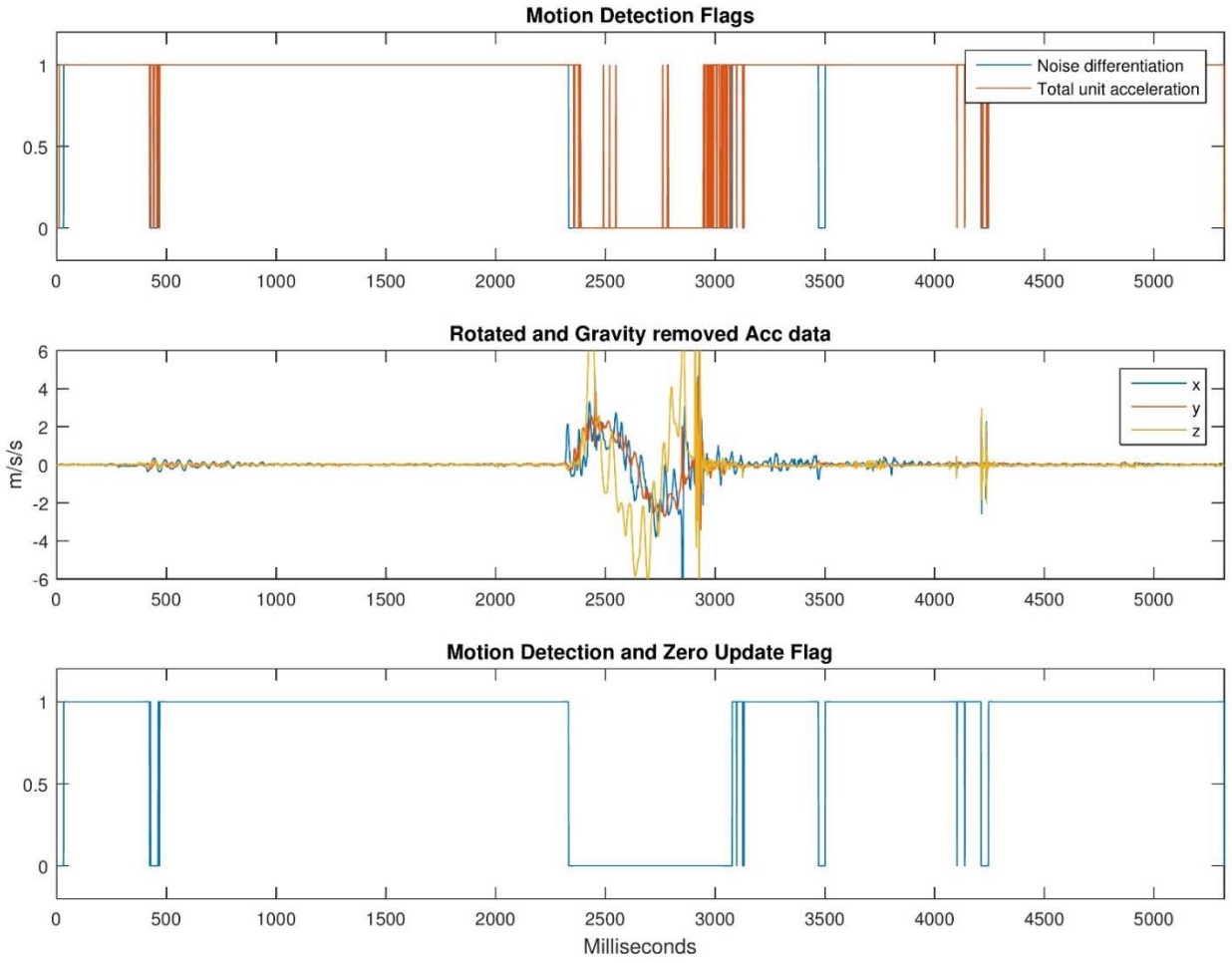

Figure 22: Motion Detection

Even after a good calibration, such sensors suffer from calibrations over time due to their vibrating susceptibilities. Depending on the intensity of vibration, this time can be shortened or increased. For example, the vibration of the stepper motors on the vehicle we use is quite large, and the time of the deterioration of the calibration is only a few seconds. For this reason, it is necessary to ignore some area (dead area) of the accelerometer data. But directly ignoring this area can cause the errors.

To overcome this, after finding the information that the change in the movement detection data has started, the errors are minimized by removing the dead area ignorance. Usually, this dead area is being processed at the moment of inactivity. It is easily understood from the rapidly rising accelerations in the same direction that come after the motion that started motionlessly.

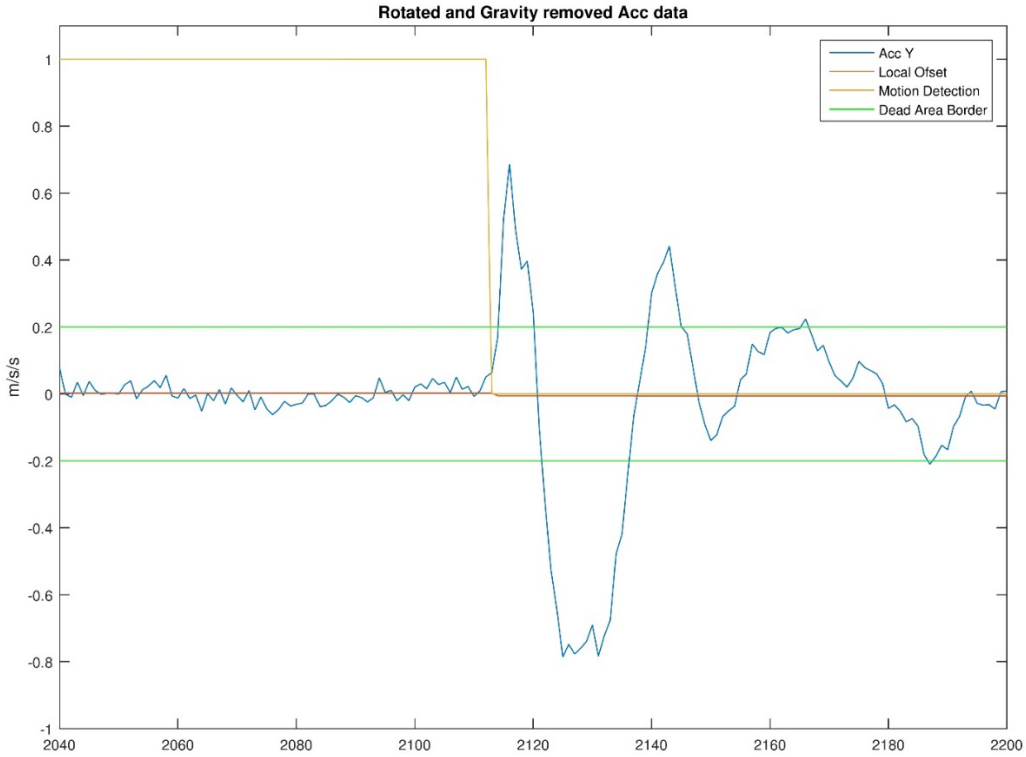

Figure 23: Dead Area

Ignorance of the dead area as part of the process obstacles to erroneous measurements. When this algorithm is not working, areas that do not participate in integration in the dead zone will cause larger errors every time.

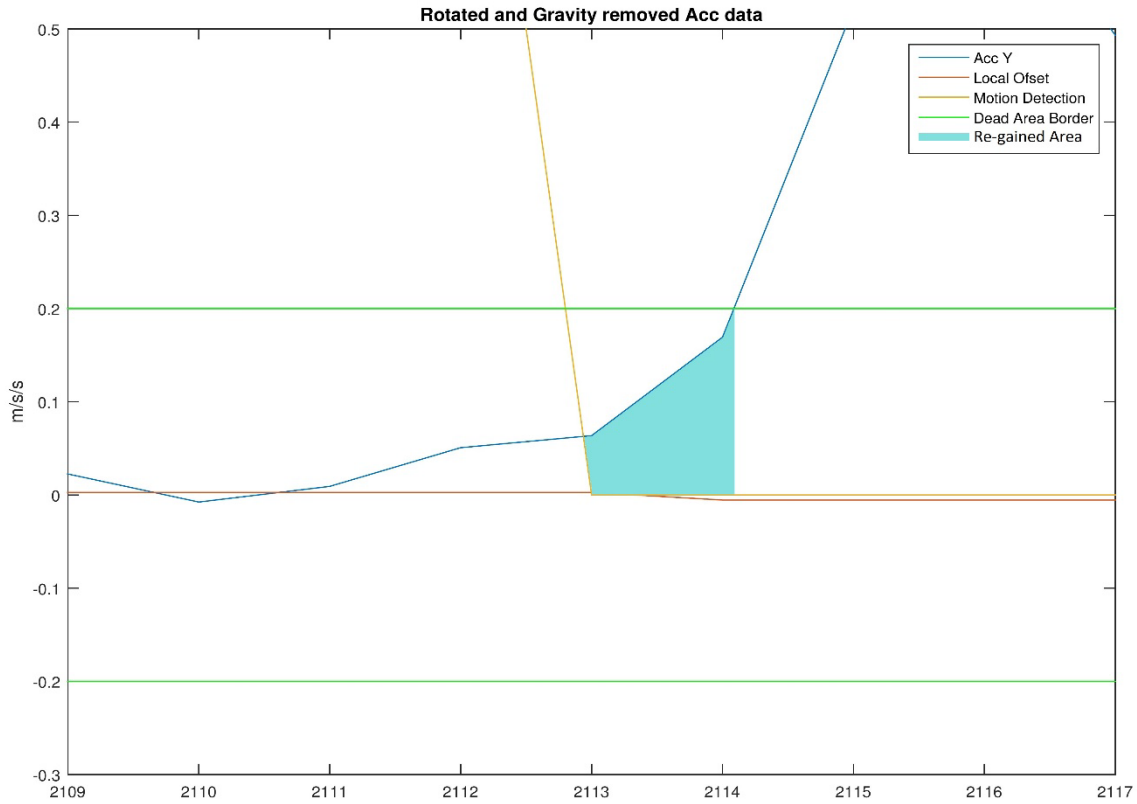

Figure 24: Re-gained Area

If the moment of inactivity continues for a certain period of time, fine offset adjustment (local offset) is done with a new calibration. When starting to act after 150 ml of inactivity, the drift can be integrated with a minimum reduction.

6.2.5 Position Update

This algorithm is based on correcting the drift that occurs during motion, instead of blocking the drift at the moment of inactivity. At the end of the motion, the speed does not return to zero because of the slip occurring at the moment of motion. This causes the position to increase even when it is stationary. It is possible to estimate the amount of drift in the position by taking the speed directly at zero moments and looking at the speed at the end of the motion.

The start time of motion and the speed at the beginning are stored in a memory. This start moment is captured by looking at the falling edge of the Motion Detection Flag.

The end of the motion is likewise detected by the rising edge of this flag. When the rising edge arrives, a triangle area is calculated using the speed and time information. This triangle field represents the drift in position. At that time, the calculated area was subtracted from the calculated position, and the calculation was tried to be corrected a little more.

Due to the lack of vibration in hand movements, the speed data return to the zero very closely and the calculated position drift update is often small.

However, when the vibration generated by the stepper motors on the vehicle enters into the work, the slip can occur in enormous levels. Even in a motion that moves forward at a constant speed, calculated speed can fall to negative values over time. Position updating in this type of example can greatly reduce the errors that occur.

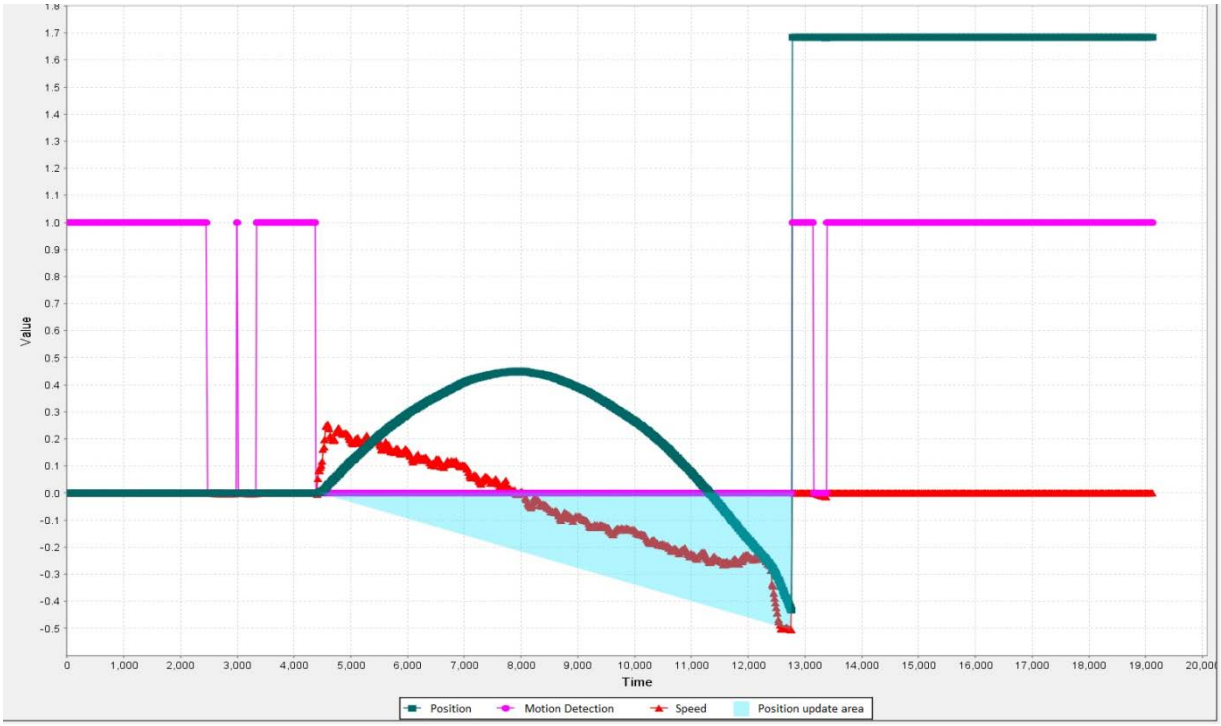

Figure 25: Position Update

6.3 Tools

- MATLAB
- Keil µVision 5

- STM32 CubeMX
- STM Studio
- Microsoft Visual Studio
- Microsoft Visio
- Microsoft Excel
- Gantt Project

7. EXPERIMENTS

7.1 Position Calculation Test

In this experiment, calculated distances are recorded, and percentage of errors are calculated. Two types of linear motions are performed. One of them is performed by hand, robot was moved from one point to another point linearly and the distance between these two points was measured by meter. The other motion is performed by motors, motors run for 4 seconds and the displacement was measured by meter. Calculated values were read from display of robot. Recorded values are as follows:

Table 7: Position Calculation Test Results

Motion with Hand			**Motion with Motors**		
Real (cm)	**Calculated (cm)**	**Error (%)**	**Real (cm)**	**Calculated (cm)**	**Error (%)**
100.00	99.89	0.11	87.00	89.35	2.70
106.50	107.09	0.55	88.50	89.24	0.83
98.00	96.64	1.38	87.00	85.04	2.25
102.00	106.80	4.70	87.50	82.61	5.58
98.00	97.80	0.20	87.50	87.05	0.51
102.00	101.24	0.74	87.00	93.03	6.93
99.00	96.08	2.94	87.50	82.82	5.34
100.00	96.98	3.02	86.00	87.03	1.19
99.50	98.85	0.65	87.00	85.57	1.64
99.00	98.94	0.06	87.00	84.57	2.79

Average percentage of error in motion with hand is 1.44%, and average percentage of error in motion with motors is 2.98%. Position calculation is satisfactory in both cases. Motor noise affects the calculation badly as seen.

7.2 Energy Consumption Test

In this experiment, current drawn by each motor is tested after motor disable method is used. Current is measured by a multimeter and the results are recorded. Current decreases and energy consumption goes down. Results are as follows:

Table 8: Energy Consumption Test Results

Motion	**Current Drawn by each Motor without Motor Disable (mA)**	**Current Drawn by each Motor with Motor Disable (mA)**
Forward - Backward	390	240
Left - Right	565	415

Energy efficiency is increased. Current is decreased by 38% in forward – backward motion and 27% in left – right motion.

7.3 Battery Performance Test

In this experiment, how long the batteries would last is tested. Voltage of batteries were measured, robot was tested on ramp (5^o) and flat (0^o) in every 10 minutes. Results are as follows:

Table 9: Battery Performance Test Results

Time (min)	**Battery1 (V)**	**Battery2 (V)**	**Battery3 (V)**	**Ramp (5^o)**	**Flat (0^o)**
0	4.03	4.03	4.03	-	-
10	3.98	3.98	3.98	Ok	Ok
20	3.94	3.94	3.94	Ok	Ok
30	3.88	3.89	3.87	Ok	Ok

40	3.84	3.84	3.84	Low Torque	Ok
50	3.79	3.77	3.77	Low Torque	Ok
60	3.73	3.71	3.68	Low Torque	Ok
70	3.68	3.67	3.59	No Motion	Ok
80	3.64	3.64	3.52	No Motion	Ok
90	3.62	3.62	3.16	No Motion	Low Torque

In first 30 minutes, robot runs at every condition. From 30th minute to 70th minute, there is no problem on flat surface, but torque is not enough to run properly on ramp. After 70th minute, motors cannot run on ramp. Batteries last 90 minutes for motion on flat surfaces.

7.4 Ultrasonic Permeability Test

In this experiment, we measured the output signal of the LM324 (in the HCSR-04) in different positions. First of all, we look the LM324 datasheet and find the output pins. Then, solder them.

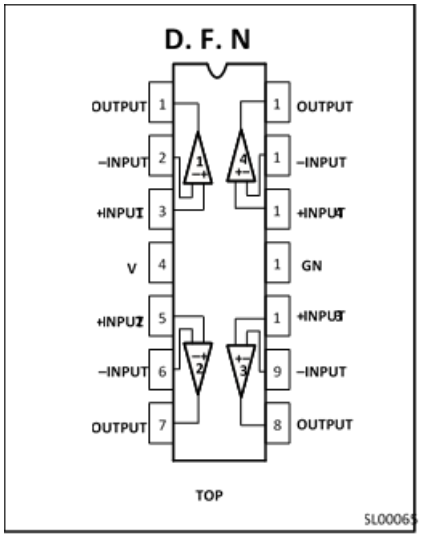

Figure 26: Pin Configuration

Next, we compiled the Arduino. It produces 1 pulse in the every 100ms. Later we observed the signal on the oscilloscope screen.

Table 10: Voltage of Signals

Distance	**Degree**	**The Wall**	**Smooth Board**	**Fabric**
15cm	0°	3.32V	3.28V	1.52V
15cm	15°	2.84V	2.6V	840mV
15cm	30°	1.6V	1.2V	1.44V
15cm	45°	760mV	880mV	840mV
40cm	0°	1.92V	3.12V	640mV
40cm	15°	1.52V	2.2V	680mV
40cm	30°	1.28V	1.56V	720mV
40cm	45°	880mV	720mV	680mV

In conclusion, we note if the degree and distance are an increase, permeability is decrease and walls and smooth wood surfaces are less permeable than fabric.

7.5 Position Update Experiments

The position update is based on the compensation of the velocity at the end of the motion. For this, a search was made to see if there was a certain pattern in the drift at speed. Correction of the drift pattern, a polynomial line was fitted between the bias from the previous motion and the bias after the motion. Polynomials of different order were used to reach the correct results. Polynomial fitting gives good results, but the polynomial order varies from motion to motion, and is generally high-order. Since the calculation cost is big enough for calculating polynomial fit, using the line fit and removing the remaining triangles is the most logical choice. In the graphs, there is experimental results of 1 m movement with polynomial fit and line fit and zero velocity correction.

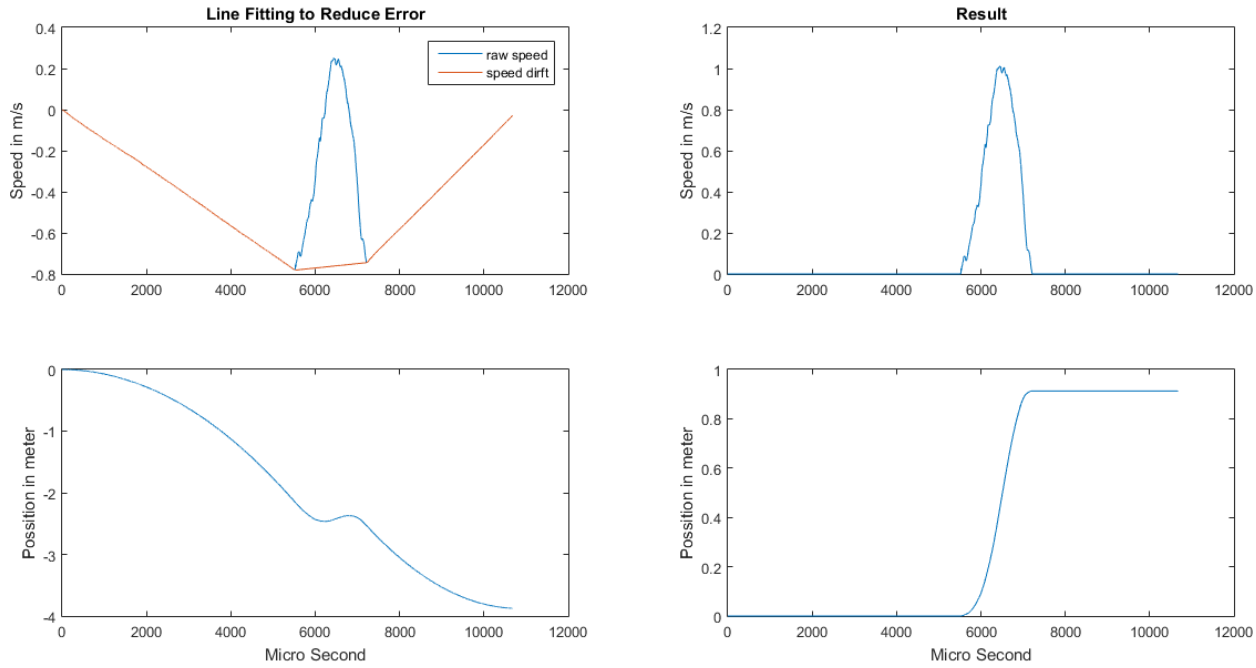

Figure 27: Line Fit

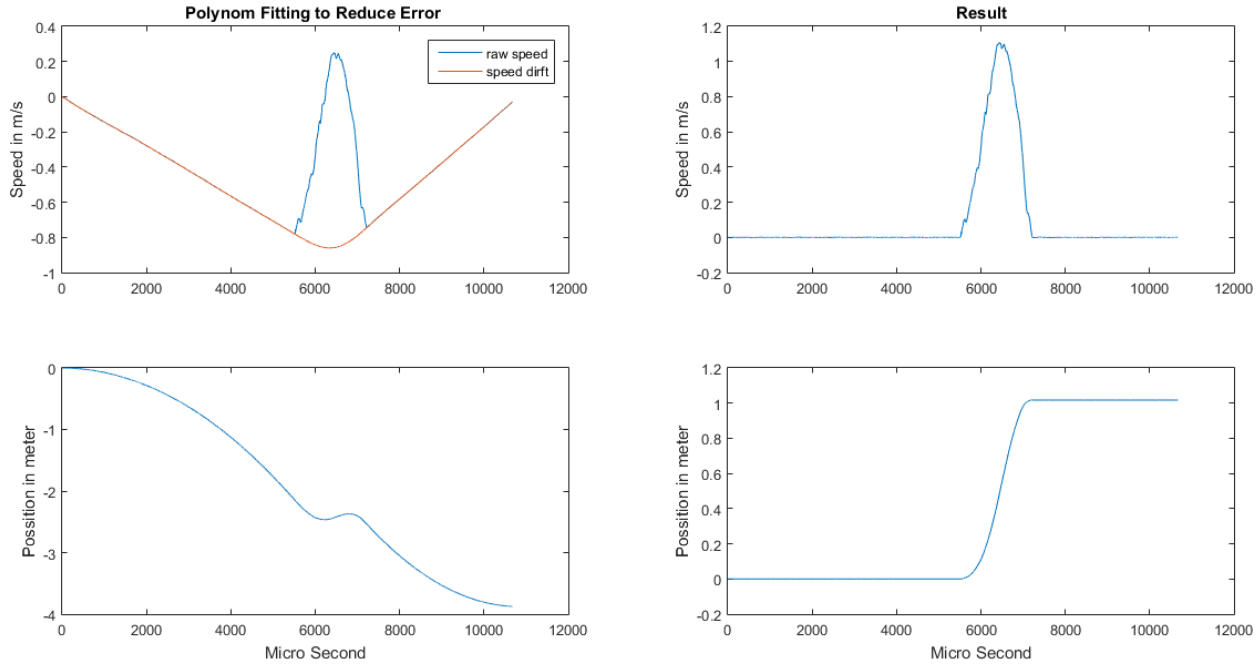

Figure 28: Polynomial Fit

7.6 MPU Signature and Temperature Test

Every MEMS sensor has a unique bias different than the others, due to their manufacturing techniques and operating principles. This value is varying with after movement and, temperature but it is approximately constant. In this experiment, two consecutive measurements were taken from the MPU that was observed on a fixed surface, then moved for a while then fixed again, and 2 more measurements were taken. Each measurement lasted about 10 seconds and the average of the raw data at the end of the measurements is given in the graph.

The purpose of the experiment is to see the effects of the temperature and the movement, with the results, determined the calibration values.

Table 11: MPU Temperature Test

	accX	accY	accZ	Temperature(C	gyrX	gyrY	gyrZ	
NO MOVEMENT	170,4182086	-399,5932536	15464,71197	27,0502015	241,447683	-11	-74,51498281	EXPERIMENT 6
	169,5874732	-400,8209204	15463,35045	27,02850896	241,4496463	-11,03036325	-74,52253398	
AFTER MOVEMENT	168,0760369	-397,9139785	15465,47312	27,16552905	241,0881336	-10,90284178	-74,51824117	
	171,3145526	-399,0675814	15468,90556	27,20612972	240,7093855	-10,86717268	-74,37396001	
NO MOVEMENT	170,3505796	-396,1577608	15471,03025	27,25202099	240,7560079	-10,48430874	-74,38507209	EXPERIMENT 7
	171,3375762	-398,6475288	15467,9063	27,27213071	240,8239675	-10,43750846	-74,11374408	
AFTER MOVEMENT	171,4553507	-397,6372438	15470,01751	27,31645041	240,665465	-10,66412607	-74,16644351	
	168,3592674	-393,5712821	15471,92557	27,33505753	240,2609524	-11,04600733	-74,03824176	
NO MOVEMENT	170,5582725	-398,0317561	15473,69514	27,40510881	240,1608447	-10,6413147	-74,1168625	EXPERIMENT 8
	171,4899592	-397,28	15474,35559	27,39696182	240,3413878	-10,78122449	-74,18791837	
AFTER MOVEMENT	169,8150952	-398,0940583	15473,69947	27,41386165	240,1504932	-10,42039459	-74,08465244	
	167,8958758	-398,2148346	15474,31708	27,41243227	240,3580053	-10,2789713	-74,01364278	
NO MOVEMENT	165,5309887	-396,2208361	15476,13643	27,45751474	240,3047113	-10,25189117	-74,03729263	EXPERIMENT 9
	166,3579935	-397,2630819	15473,47095	27,47854693	239,9135691	-10,28076507	-74,38650307	
AFTER MOVEMENT	166,5115952	-397,7779339	15474,77255	27,47079616	240,3158819	-9,898336847	-74,22440853	
	167,7740688	-396,8923265	15473,88276	27,45920822	240,3700387	-10,0400977	-74,21331162	
NO MOVEMENT	148,3941658	-360,8729989	15409,39879	25,38614079	246,0451797	-14,53788687	-76,44397012	EXPERIMENT 10
	152,8512852	-377,8164015	15415,84394	25,44639427	245,2769278	-14,45746634	-76,37071603	
AFTER MOVEMENT	150,6878595	-419,312746	15415,97154	25,55464426	245,5142295	-14,16409325	-76,14956101	
	152,4879253	-424,5062577	15418,05888	25,58380906	245,6441036	-14,18949409	-76,01586462	
NO MOVEMENT	156,4499925	-418,1855918	15425,45105	25,84475233	244,664912	-13,94119416	-75,85817416	EXPERIMENT 11
	157,5311466	-421,8561541	15424,6783	25,79655927	245,1643094	-14,03370448	-76,09840506	
AFTER MOVEMENT	149,8506375	-413,058895	15432,67031	26,16208686	243,6555859	-12,89101396	-75,37553127	
	155,0741997	-420,0879186	15431,67724	26,15950982	243,7815312	-12,77209028	-75,27939224	

As you can see in the table, the effect of temperature is higher than motion. Look at the graphics to see better visually.

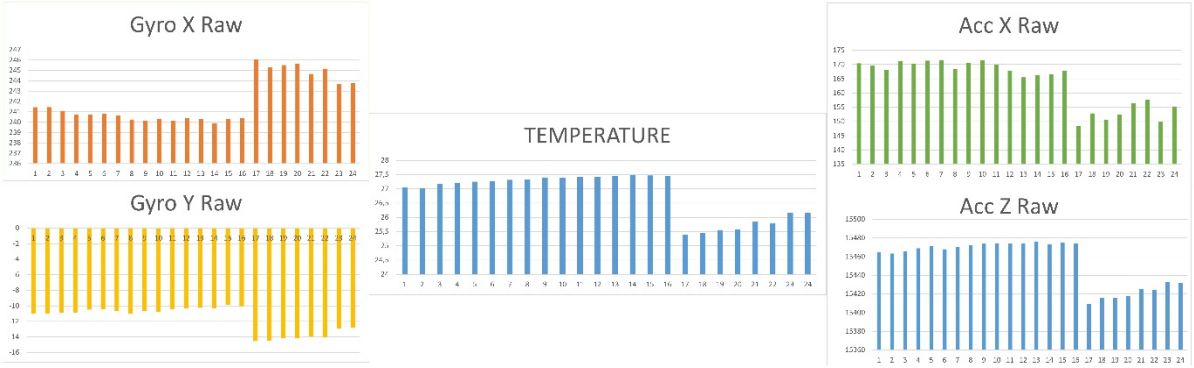

Figure 29: MPU Temperature Test

8. SHARE OF WORK

Yasin Kağan TAŞKIN: Trapezoidal Discrete Integration, Quaternions, Motor Drive, Motion Detection and Zero Update, Position Update, PC Data Recorder Software, OLED Display, Energy Consumption Optimizations, Extended Kalman Filter.

Şafak TAMBOVA: Introduction to Kalman Filter, Applications of Kalman Filter, Applications of Extended Kalman Filter, Quaternions, Ultrasonic Sensor.

Alaettin SÖMER: Motor Drive, Ultrasonic Sensor, OLED Display, Energy Consumption Optimizations, Applications of Kalman Filter.

Adem YÜCEDAL: Researches of MPU6050, Trapezoidal Discrete Integration, Experiments of Ultrasonic Sensor.

Berkay KOÇ: Quaternions, Gimbal Lock, Ultrasonic Sensor and Experiments.

9. PROJECT PLAN

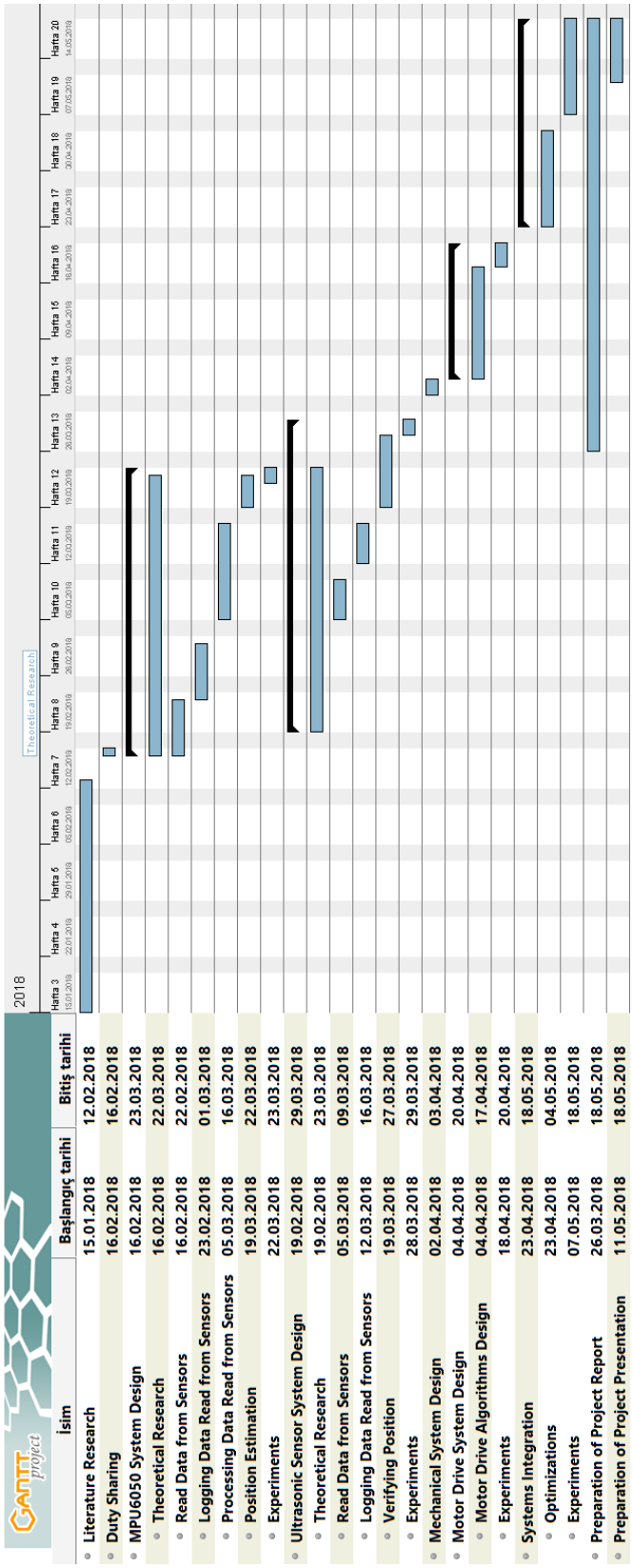

Figure 30: Gantt Chart of Project

10. DEMONSTRATIONS

A number of scenarios have been devised to test and verify the algorithm, for this purpose four different demonstrations have been designed. The selection of these demonstration modes and internal selections in these modes are provided with a button on the robot. A menu is constructed via OLED screen to communicate with the user.

If the mode is not changed within 3 seconds after that mode appears on the screen, the demo is automatically selected.

10.1 Mode 1 - Full Demo

This demonstration is a full-scale demonstration that the whole algorithm and all external factors are tested at the same time.

- After selecting Mode 1 with the button, the "MOVE IT" command appears on the screen, this means, the robot is ready to calculate position done by hand.
- After the motion, “Return initial position and orientation” command is sent via button and robot starts to return initial position and orientation.

10.2 Mode 2 - Motors Effect Only

In this demonstration, stepper motors are driven forward for 4 seconds. Change in the position is calculated during this time interval.

- The motion starts after 1 second when mode selected from the user interface.
- At the end of the motion, the calculated position values are shown on the screen. **(It is possible to see that the values change after the values come to the screen. At this point, the position calculation operations are interrupted, and the movements to be made after that time will be calculated incorrectly.)

10.3 Mode 3 - Yaw Protection

Since there is no additional measurement of the Yaw, it is calculated only by trusting the gyro data. This demo has been added to check the accuracy of these values.

- After selecting Mode 3, "KICK IT" command appears on the screen.

- The robot gives the user 3 seconds to turn the vehicle.
- At the end of 3 seconds, the robot moves to the starting angle.

10.4 Mode 4 - Free Measurement

In this demonstration, only the calculation of movements is considered without the use of motors.

- After selecting Mode 4, "MOVE IT" command appears on the screen.
- At the end of the movement by hand, the current position is displayed on the screen after the button is pressed.

11. CONCLUSION

In conclusion, the position is estimated using MPU 6050 with less error. In order to make a correct estimation and compensate drifts on MPU 6050 Kalman filter designed and tested. Quaternion calculations are performed correctly to get rid of Gimbal lock problems. Another reason using quaternions is, eliminate the gravitational acceleration from the desired move. A proper motor drive algorithm is designed. Numerical integration algorithm is designed in order to obtain position values from acceleration values. Using better integration algorithms numerical errors can be reduced. Using a powerful processor, the system can be designed more real-time and with more readings from the sensor, a better estimation can be performed. Experiment results show that MPU 6050 is a good device for short-term position estimation operations alone. But for using it for long-term, it is not enough to use only MPU 6050. It must be supported and compensated with some other devices. Recommendation from the experiment results for long-term motions, the error occurred from MPU 6050 can be compensate using ultrasonic sensors.

The requirements that achieved in this project are listed below:

-Processor reads data from the sensor and process in every 1 millisecond.
-Algorithm can be applied everywhere without any connection.
-With full charge of batteries, the robot can be run two hours.

-The speed of the robot is 0.22 m/s approximately.

-The motors rotate in 58 rpm for forward direction and 29 rpm to turn left and right.

-Total cost of the system 447₺ which is approximately 84€ which is a low-cost solution comparing to similar systems in order to make position estimation.

-All the electronic components are soldered on a stripboard with proper sockets.

-If any of the components fail for some reason, it is easy to replace it with the new one.

-All the components used in this project are easy to find via internet.

-From the point of view of the software, it is easy to add and debug new lines according to requirements.

REFERENCES

[1] ece.montana.edu.com, 'Under Navigation System', 2018. [Online]. Available: http://www.ece.montana.edu/seniordesign/archive/SP14/UnderwaterNavigation/Kalman_filter.html. [Accessed: 28- October- 2013].

[2] M. Strohmeier and S. Montenegro *Quaternion Based Extended Kalman Filter*. Würzburg, Germany 2016, pp. 17-18.

[3] Zhi, Ruoyu, "A Drift Eliminated Attitude & Position Estimation Algorithm In 3D" (2016). Graduate College Dissertations and Theses. Paper 450, pp. 22-26.

[4] John J. Craig. Introduction to Robotics Mechanics and Control. Pearson Education International, 2005.

[5] J. B. Kuipers. Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality. Princeton University Press, 1999.

[6] Itzhack Y Bar-Itzhack. New method for extracting the quaternion from a rotation matrix. AIAA Journal of Guidance, Control and Dynamics, 23(6):10851087, Nov.Dec 2000. (Engineering Note).

[7] Mei Wang, Yunchun Yang, R.R. Hatch, and Yanhua Zhang. Adaptive filter for a miniature mems based attitude and heading reference system. Position Location and Navigation Symposium, 2004. PLANS 2004, pages 193 – 200, apr. 2004.

[8] 3dvrm.com, ' Quaternions for Ouest3D ', 2018. [Online]. Avaible: http://3dvrm.com/quat/z [Accessed: 2011].

[9] *Allegro MicroSystems A3964 Datasheet,* Allegro MicroSystems Inc., Worcester, MA, 2010.

[10] *STMicroelectronics STM32F103x8 and STM32F103xB Datasheet,* STMicroelectronics, 2017.

[11] *Atmel ATMega328/P Datasheet,* Atmel Corporation, San Jose, CA, 2016.

[12] forum.arduino.cc, ' Benchmark STM32 vs ATMega328 (nano) vs SAM3X8E (due) vs MK20DX256 (teensy 3.2)', 2016. [Online]. Available: https://forum.arduino.cc/index.php?topic=431169.0. [Accessed: 15- May- 2016].

[13] invensense.com, 'Datasheet of MPU6050', 2018. [Online]. Available: https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/ [Accessed: 28- May- 2014].

[14] mems-exchange.org, 'What is MEMS technology', 2018. [Online]. Available: https://www.mems-exchange.org/MEMS/what-is.html. [Accessed: 10-August-2015].

[15] stevecurd.me, 'Personal blog, working about gyroscope', 2018. [Online]. Available: https://stevecurd.me/2015/08/25/mems-and-me-how-does-my-fitbit-know-im-walking/ [Accessed: 10- August- 2015].

[16] wikipedia.org, 'Trapezoidal rule', 2018. [Online]. Available: https://en.wikipedia.org/wiki/Trapezoidal_rule. [Accessed: 30- May- 2015].

[17] wikipedia.org, ' Newton–Cotes formulas', 2018. [Online]. Available: https://en.wikipedia.org/wiki/Newton%E2%80%93Cotes_formulas. [Accessed: 30-May- 2015].

[18] S. O. H. Madgwick, A. J. L. Harrison and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," 2011 IEEE International Conference on Rehabilitation Robotics, Zurich, 2011, pp. 1-7.

[19] Implementing Positioning Algorithms Using Accelerometers, Freescale Semiconductor, Inc., NXP Semiconductors. 2007, Rev 0.