AN ABSTRACT OF THE REPORT OF

Shuxian Qin for the degree of Master of Science in Electrical and Computer Engineering presented on August 16, 2021.

Title: Exploring Indoor Navigation Based on 9-dof Kalman Filter and SLAM Algorithm

Abstract approved: _____

Huaping Liu

Traditional localization techniques rely on triangulation or trilateration, where in a set of three or more stationary known locations is used to estimate a "client" position. For inertial navigation, these techniques can estimate client positions merely using the measured data from tri-axial accelerometers and gyroscopes. However, the use of double integration of the accelerometer data and a single integration of the gyroscope data to determine the client's position are greatly affected by gyro-drift and other noise, causing small errors to explode towards infinity. This process could be improved by applying reasonable algorithms and even deep learning tools such as the Kalman filter and simultaneous localization and mapping. Therefore, this research aims at exploring the possibility of indoor navigation based on inertial sensors that uses the fusion algorithms. Meanwhile, a multitude of formulas involving navigation are refined and corrected. These equations and optimized

results provide accurate raw data for future deep learning models.

Exploring Indoor Navigation Based on 9-dof Kalman Filter and SLAM Algorithm

by

Shuxian Qin

A REPORT

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented August 16, 2021
Commencement June 2021

Master of Science report of Shuxian Qin presented on August 16, 2021.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

---

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my report will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my report to any reader upon request.

Shuxian Qin, Author

ACKNOWLEDGEMENTS

I would like to express my highest regard for and thanks to my best advisor, Prof. Huaping Liu. I could not have finished my research without his daily excellent guidance and timely concerns. I really appreciated that Prof. Liu always supported me and provided tremendous help when the research stalled during the research. Prof. Liu's serious scientific attitude and rigorous academic spirit also inspired me constantly. Meanwhile, I would like to say thank you to all of my committee members, Prof. David Allstot and Prof. Tejasvi Anand for their support in my research and their time spent reading this report. Finally, I want to say thank you to my mother because she always supports me and encourages me, telling me not to give up.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

Chapter 1: Introduction

In the contemporary world, navigation plays an indispensable role in our daily travel along with the rapid and smart development of modern transportation. During the last decade, the surge of cellphones and other smart handheld devices has increased the demand for location-based services, offering users precise real-time positioning. However, this high-accurate positioning cannot exist independently of full-fledged global satellite communication. Among all of the mature satellite systems, the Global Positioning System (GPS) is the most versatile and is tied to numerous location services. Applications of GPS include are car navigation, aviation, and several other outdoor scenarios which allow for continuous connection with satellite [1]. However, the particular system is limited and fails to navigate within indoor environments. Signal attenuation and shielding due to steel and concrete walls greatly reduce the feasibility of indoor navigation, so GPS cannot offer the precise or even reliable local coordinates in such environments [2].

Therefore, users urgently need reliable and efficient systems which require only local information, such as sensor data, WLAN nodes, radio waves, acoustic signals, and so forth [1]. These systems are collectively called the indoor positioning system (IPS) [3]. Although numerous studies have conducted on indoor localization, a uniform standard has not yet been defined for IPS; thus, it may cause multiple compatibility problems for both software and hardware in practice. Likewise, many

existing mature indoor positioning technologies are aimed at other applications rather than at the localization of people [4].

To address these issues, the inertial measurement unit (IMU), one of the microchips for locating people or objects, was created to provide accurate positions in indoor scenarios. Specifically, IMUs are designed to calculate and precisely estimate the proposed position for users or objects by collecting real-time local data [5]. These IMUs use built-in accelerometers, gyroscopes, and sometimes magnetometers to calculate the positions of targets by measuring linear acceleration, angular velocity, or even magnetic fields and computing the body orientation and heading from these measurements. Each sensor has 3 axes according to the real 3-dimensional world, and one axis is defined as one degree of freedom (dof). Usually, the double integral of acceleration based on time intervals provides an accurate position, while the single integral of the gyroscope provides the orientation of objects by using Euler angle including pitch, yaw, and roll. Consequently, combining the datasets of accelerometers and gyroscopes for 6-dof IMUs is the most popular method of computing a user's final position.

Some advanced IMUs also integrate another 3-axis magnetometer as an effective complementary method to correct data errors of the accelerometer and the gyroscope. The detailed principle is as follows: Like a compass, a magnetometer can sense the surrounding magnetic field. When an object is in motion, users can calculate the magnetic field direction pertinent to the earth by comparing magnetic field intensity before and after the object's movement [3]. In spite of this, a magnetometer can hardly be used alone in indoor positioning because of its vulner-

ability to external interferences such as environmental metals and other magnetic sources.

The method of combining the readings from all built-in sensors is called sensor fusion. However, simple sensor fusion may result in inaccurate readings due to accumulative errors and electric noise. Electric noise, usually called device bias, is easy to eliminate by calibrating the sensors beforehand, but calibration achieves little in preventing accumulative errors, namely dynamic noise, because these errors come from each-time measurement and are magnified by the mathematical integral. Therefore, a qualified algorithm is needed to filter accumulative errors. In recent years, many mature and low-cost algorithms have been adopted, such as the Kalman filter, simultaneous localization and mapping (SLAM), the alpha-beta filter, and so forth [6].

The use of machine learning (ML) in position prediction has also been suggested. Trained by historical datasets, ML models can analyze and acquire an object's behavior so that it can predict a position more accurately. In addition, performance prediction follows training intensity, that is, long-term training with a large set of samples significantly improves accuracy. ML techniques confirmed to be effective in indoor navigation include residual neural networks (ResNet), artificial neural networks (ANN), k-nearest neighbor (KNN), support vector machines (SVM), and so forth [5, 7, 8]. Some of these models, as their names imply, involve the neural network (NN), which has several advantages over certain non-neural networks; for example, it provides better generalization results and exhibits few inaccuracies in data training [9]. Nevertheless, the need for a large amount of pre-

trained data and complex network structures make NN training demanding work. For this reason, a significant amount needs a large number of optimization is still need in the future.

In this report, we first focus on the fundamental principles of accelerometers, gyroscopes and magnetometers as well as the transformation of coordinates from a 3D rotation matrix to a quaternion. Then, these existing formulas are integrated, proofread and revised in the methodology for future research. The next section presents specific details regarding of calibrations and the Kalman filter algorithm. Based on these principles, a proposed 9-dof sensor fusion approach is delineated. The experimental results are demonstrated in 2D and 3D trajectories.

Chapter 2: Literature Review

Limited by the available resources, indoor positioning is a difficult but crucial process, so it has been a perennial concern in recent decades, inspiring a significant body of relevant research. In fact, several robust location algorithms have been proposed and developed to estimate distance in indoor navigation [10, 11]. According to [12], these algorithms have been roughly classified into 6 categories: inertial sensor, triangulation, fingerprinting, neighborhood, dead reckoning, and proximity. Therefore, the next section will focus on the fundamental concepts of these algorithms. Notably, the algorithm of our project relies on the inertial motion and sensors, that is, the accelerometer, gyroscope, and magnetometer.

2.1 Inertial sensor

Algorithms using inertial sensors are beacon-free. The data used by inertial sensors, including data from accelerometers, gyroscopes, and magnetometers, can be categorized into 3 types: acceleration, angular velocity and magnetic field. Through the use of a double integral, acceleration can be converted into position. Likewise, a single integral of angular velocity can help to calculate the Euler angles. Finally, the magnetic field provides the object's direction pertinent to the earth. In [13], the author successfully used this double-integrating approach to obtain the rough

position of a user. Meanwhile, The author also combined the user's number of steps and leg angles during movement to fuse the data from both the accelerometer and the gyroscope, implementing a low-cost but relatively accurate algorithm.

Certain additional methods have been derived to implement inertial navigation. In [14], the author used an MEM gyroscope alone to estimate an object's orientation with the extended Kalman filter (EKF). By employing Vicon optical, [14] significantly improved rapid estimator converges for dynamic movements, eliminating the sudden inclination error in their test. In addition, the yaw drifting was also corrected and the root-mean-square error of angles was lowered [14].

The author in [15] proposed a new method of improving orientation by processing the IMU's output readings. In this paper, a new sensor fusion called MUSE, a magnetometer-centric algorithm for orientation tracking, was designed to combine the IMU data. Meanwhile, the author provided another sensor fusion by using ergonomics on human shoulder joint and elbow movement to, simulate the walking poses of humans. Both of these techniques ultimately yield accurate results.

Machine learning and deep learning can also be applied in inertial measurement. In [16], the author presented such a possibility by using an artificial neural network (ANN). The detailed process involves training the IMU data to acquire an object's trajectory, correcting the output readings. This approach currently requires orientation information of stationary and moving state for pre-training, and some other motions, such as jogging and spinning, can also be added into the training model based on ANN.

2.2 Triangulation

Triangulation for indoor positioning imitates GPS calculation to build the geometric mesh for indoor environments. This approach, also called angulation, uses reference points with angles to define the position. AOA, angle of arrival, has been a popular algorithm for implementing triangulation. By determining the direction of the RF signal from an object's antenna array, the device can compute AOA the via the time difference of arrival (TDOA), the return of flight (RToF) and signal intensity [17, 18]. Due to this feature, triangulation for indoor navigation has specific requirements for the measured environment, namely a moderate number of reflectors that can provide ample information without increasing complexity.

2.3 Fingerprinting

Essentially, the fingerprinting algorithm is a variant of k-nearest neighbor, kNN. As the name suggests, this algorithm attempts to construct the database of positions for training, using unique features like a fingerprint to compare the measured signal strength with the database and estimate an object's location. Therefore, fingerprinting involves two steps: training and positioning. In the training phase, fingerprinting chooses a reference point and calculates the relevant accessing signal strength. This information is then saved in a "fingerprint" database. In the next positioning step, the device detects the signal strength again and compares it with the database via probabilistic method [19]. In [20, 21], the authors suggested that fingerprinting requires sufficient reference points to improve accuracy, but

this work is also burdensome and inefficient. The author in [22] also pointed out that fingerprinting is challenging for large-scale deployment, because the RF signal needs to be empirically mapped for each floor plan configuration.

2.4 Connectivity and neighbourhood

The author in [12] defined a new navigation category based on connectivity and a neighbourhood algorithm that implements navigation through the analysis of connectivity and the number of attainable neighbors. If sufficient reference spatial points are set, the object's location can be determined by these points where it can connect with the neighbors. Also, [12] mentioned that if overlaps exist in the signal coverage, the object can export the intersection of the adjacent coverage areas. The auther [23] added that the accuracy of this approach relies on the number and distribution of reference points and their coverage area. Finally, the author pointed out that the signal attenuation and reflection increase the positioning inaccuracy, because different neighbors have different signal ranges, and certain areas have specific dynamics that may affect positioning [23]. Therefore, more research and development are needed to improve connectivity and neighbourhood algorithms in the next decade.

2.5 Dead reckoning

Dead reckoning is the process of computing the current position of a moving object via a previously determined position based on estimated speed and heading direction over elapsed time [24]. This can be a beacon-free algorithm and is usually jointly used with an inertial sensor to provide accurate position information for indoor navigation. Among all the dead reckoning algorithms, the most popular is called pedestrian dead reckoning (PDR). PDR collects a user's moving acceleration, angular velocity, and magnet strength and uses these data to calculate step size and direction in order to achieve position and track the individual [25]. However, PDR is subject to cumulative errors from IMUs and needs a massive dataset to detect different gaits according to different step sizes. Therefore, researchers have attempted to fuse indoor beacons with PDR to implement effective navigation accuracy. In [25], the author used PDR with a Wi-Fi weighted path loss algorithm to combine PDR and received signal strength indication (RSSI) signals, reducing IMU sensor errors and signal fluctuation. By using the model, they improved the average location error to 1.45 meters especially when the user moved within a rectangular [25].

2.6 Proximity

Proximity for indoor navigation is a beacon-based algorithm that uses WiFi, Bluetooth, RFID, and similar tools to transmit location information when the user is in proximity. When the user is close to the target beacons, their built-in receivers

can determine position information by detecting the wireless signal strength [26]. Also, when the connection to this beacon is lost, the system will relocate the user's position by linking them with the highest signal strength beacon. A typical proximity was proposed in [27], and the author separated this technique into 3 layers: an IoT layer, an edge layer and a cloud layer. At the IoT layer, many low-cost Bluetooth devices are deployed for short-range wireless communication. These devices use proximity-oriented beacon technology and transparently interact with users to determine their position. The edge layer provides solutions and supports this navigation by tracking users motion. Finally, the cloud layer is responsible for optimal path computation, and meanwhile cross-relates information from users and indoor environments [28].

Chapter 3: Background and Methodology

3.1 Accelerometer

An accelerometer, the most common motion sensor in IMUs, measures the changing rate of speed corresponding to a time interval in the x-y-z dimensions. Accelerometers can sense the pressure caused by deformation and convert it to an electrical signal. It is then easy to obtain acceleration from a=F/m. Like other acceleration measurement devices, accelerometers obey the inertial coordinate system with a unit of m/s^2. However, the raw readings contain two types of unexpected accelerations: gravitational acceleration and centripetal force acceleration.

First, gravity always exists in the IMU body frame, affecting the acceleration readings. For example, if one accelerometer is static on a flat table, its reading should be $[0, 0, 9.8m/\mathrm{s}^2]^T$ rather than $[0, 0, 0]^T$. Also, the direction of gravity depends on the setting, meaning it is not always along with the z-axis. Another unexpected acceleration is from the centripetal force, namely Coriolis force, caused by the rotation of objects. Usually, the rotation of objects takes place in two ways: the rotation around a spatial point or self-rotation [29]. For both situations, centripetal force equals the product of linear velocity and angular velocity. The equations (3.1) and (3.2) define the extra gravitational and centripetal accelerations mathematically. In the proposed system, these two redundant accelerations

should be erased, as defined in the equation (3.3)

$$\begin{bmatrix} A_{x,g} \\ A_{y,g} \\ A_{z,g} \end{bmatrix} = Orietation(t)^{-1} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{3.1}$$

$$\begin{bmatrix} A_{x,c} \\ A_{y,c} \\ A_{z,c} \end{bmatrix} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \cdot \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{3.2}$$

$$\begin{bmatrix} Acc_{x,l} \\ Acc_{y,l} \\ Acc_{z,l} \end{bmatrix} = \begin{bmatrix} Acc_{x,raw} \\ Acc_{y,raw} \\ Acc_{z,raw} \end{bmatrix} - \begin{bmatrix} A_{x,g} \\ A_{y,g} \\ A_{z,g} \end{bmatrix} - \begin{bmatrix} A_{x,c} \\ A_{y,c} \\ A_{z,c} \end{bmatrix} \tag{3.3}$$

Where:

- $A_{x/y/z,g}$ = Gravity in body frame
- $A_{x/y/z,c}$ = Centripetal accelerations
- g = Gravity, $9.8m/s^2$
- $\omega_{x/y/z}$ = Angular velocities
- $V_{x/y/z}$ = Linear velocities
- $Acc_{x/y/z,raw}$ = Raw linear accelerations
- $Acc_{x/y/z,l}$ = Proposed linear accelerations

Most notably, two coordinate systems exist: body frame and inertial frame [30]. Body frame, where all the data is collected, processed and computed, is the coordinate of a moving IMU. Inertial frame is a stationary coordinate that causes some stationary force, such as gravity. Gravity is symbolized as $[0, 0, 9.8m/\mathrm{s}^2]^T$ with respect to the inertial frame. Using a rotation matrix, one coordinate can be rotated to another, and we can transfer gravity from an inertial frame to a body frame for navigation. In the equation (3.1), $Orientation(t)^{-1}$ is the rotation matrix fused by the accelerometer, gyroscope, and magnetometer. The details of the rotation matrix will be demonstrated later.

The proposed linear accelerations can be integrated to obtain the velocity and position with respect to time [31]. There are two steps to implement this simple algorithm. The first step is to calculate the integral of proposed accelerations to obtain the velocity of objects as shown in equation (3.4), where v_{t-1} denotes the velocity for the last time frame and T represents the time interval.

$$v_t = v_{t-1} + \int_0^T a dt = v_{t-1} + aT \tag{3.4}$$

Plugging equation (3.4) into (3.5), we recalculate a second integral of the previous velocity to obtain the position of an objects. Like equation (3.4), y_{t-1} represents the position at last time frame. Using double integration, we can calculate the accurate position of the object. Nonetheless, the double integral of linear acceleration fails to produce precise positions because of dynamic and electrical noise. These issues can be solved after calibration and sensor fusion.

$$y_t = y_{t-1} + \int_0^T v_t dt = y_{t-1} + v_{t-1}T + \frac{1}{2}aT^2 \tag{3.5}$$

3.2 Gyroscope

The gyroscope, the vital component in IMUs, is usually a vibrating structure gyroscope that uses the Coriolis effect [32]. In the IMU system, a gyroscope detects angular velocity based on Newton's law of conservation of momentum [33]. Like the accelerometer, the gyroscope takes measurements in 3-axis coordinates and its output is converted to an electrical signal. Specifically, the gyroscope measures the rotation angular velocity by referring to a body frame in the Euler angle system: roll (x-axis), pitch (y-axis), and yaw(z-axis) shown in figure 3.1. Using the Euler angles, we can construct a rotation matrix to implement the transformation from an inertial frame to a body frame. Equations (3.6) to (3.8) demonstrate the single integrals of the Euler angles.

$$\theta = \int_0^T \omega_\theta dt = \omega_\theta T \tag{3.6}$$

$$\phi = \int_0^T \omega_\phi dt = \omega_\phi T \tag{3.7}$$

$$\psi = \int_0^T \omega_\psi dt = \omega_\psi T \tag{3.8}$$

Where:

- θ : Roll ϕ : Pitch ψ : Yaw

- $\omega_{\theta/\phi/\psi}$ = Angular velocities of roll, pitch and yaw
- T = Time intervals

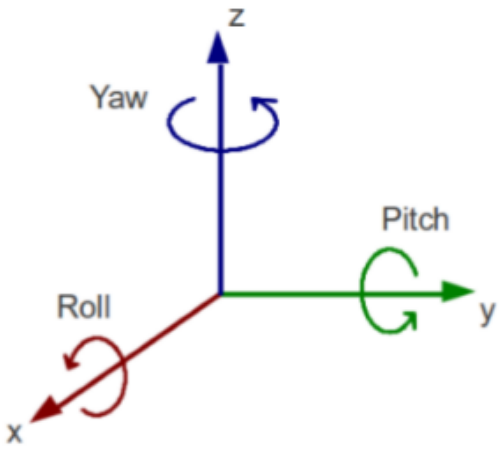


Figure 3.1: Euler angles in x,y,and z directions

However, gyroscope drift overtime makes the integration unreliable in prolonged measurement: small errors will accumulate and cause relatively large bias [33]. In addition, the speed rate of the gyroscope will not return zero when the object transitions from movement to a standstill. In detail, the measured angular velocity ω_t of a static object at present differs from the velocity ω_{t-1} detected on last static condition, if this object is moved in the time interval. Consequently, the gyroscope cannot work alone in inertial measurement for indoor positioning. These issues can be addressed by using a filter algorithm in sensor fusion.

3.2.1 Rotation matrix

The establishment of a rotation matrix aims at representing the object's rotation by using Euler angles. In this project, a rotation matrix has been used to transform a two different coordinate systems. In the 3-dimension coordinate system, the whole rotation matrix is divided into 3 element matrices related to each axis. The following equations (3.9) to (3.11) rotate vectors via the angle α.

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix} \tag{3.9}$$

$$R_y(\alpha) = \begin{bmatrix} cos\alpha & 0 & sin\alpha \\ 0 & 1 & 0 \\ -sin\alpha & 0 & cos\alpha \end{bmatrix} \tag{3.10}$$

$$R_z(\alpha) = \begin{bmatrix} cos\alpha & -sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

Matrix multiplication of each element derives the complete rotation matrix. However, the Euler angle rotational sequence should be defined before the matrix multiplication. The reason is that distinct rotation sequences cause different orientations of the object, even if each Euler angle is identical [34]. In this project, a Z-Y-X sequence, namely Yaw-Pitch-Roll, has been used because the underlying

logic of our sensors is based on it. Equation (3.12) demonstrates the ZYX rotation matrix. In the final matrix, each element is symbolized as m_{ij} for easier description later. However, the Euler rotation creates singular points, making the rotation invalid. This phenomenon, called gimbal lock, occurs when two axes in the Euler coordinates coincide through rotation. In the case of gimbal lock, the object loses one degree of freedom unless the rotation sequence is changed. More information on gimbal lock will be introduced in the next section. To improve the inertial measurement, the possibility of gimbal lock needs to be minimized.

$$
\begin{aligned}
R(\psi,\phi,\theta) &= R_z(\psi)R_y(\phi)R_x(\theta) \\
&= \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\phi & 0 & sin\phi \\ 0 & 1 & 0 \\ -sin\phi & 0 & cos\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} \\
&= \begin{bmatrix} cos\psi cos\phi & cos\psi sin\phi sin\theta - cos\theta sin\psi & cos\psi cos\theta sin\phi + sin\psi sin\theta \\ cos\phi sin\theta & sin\psi sin\phi sin\theta + cos\psi cos\theta & cos\theta sin\psi sin\phi - cos\psi sin\theta \\ -sin\phi & cos\phi sin\theta & cos\phi cos\theta \end{bmatrix} \\
&= \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}
\end{aligned}
\tag{3.12}
$$

3.2.2 Gimbal lock

Gimbal lock refers to the loss of one degree of freedom in a 3D mechanism which occurs when two axes are rotated into a parallel configuration. In this case, two Euler angles have identical performances, seemingly locked mutually. Unless the rotating sequence is changed, the loss of one dof cannot be restored. This constitutes a challenging problem because the rotating sequence should be fixed before measurments are taken. This phenomenon can be described in the rotation matrix, shown as equation (3.13). Assuming a pitch angle along the y axis, where ϕ is just rotated into $-\frac{\pi}{2}$, the x and z direction are actually parallel. With $cos(-\frac{\pi}{2}) = 0$ and $sin(-\frac{\pi}{2}) = -1$, the simplified ZYX rotation matrix is as follows.

$$
\begin{aligned}
R &= \begin{bmatrix} cos\psi cos(-\frac{\pi}{2}) & cos\psi sin(-\frac{\pi}{2})sin\theta - cos\theta sin\psi & cos\psi cos\theta sin(-\frac{\pi}{2}) + sin\psi sin\theta \\ cos(-\frac{\pi}{2})sin\theta & sin\psi sin(-\frac{\pi}{2})sin\theta + cos\psi cos\theta & cos\theta sin\psi sin(-\frac{\pi}{2}) - cos\psi sin\theta \\ -sin(-\frac{\pi}{2}) & cos(-\frac{\pi}{2})sin\theta & cos(-\frac{\pi}{2})cos\theta \end{bmatrix} \\
&= \begin{bmatrix} 0 & cos\psi sin\theta + cos\theta sin\psi & cos\psi cos\theta - sin\psi sin\theta \\ 0 & sin\psi sin\theta - cos\psi cos\theta & cos\theta sin\psi + cos\psi sin\theta \\ 1 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & sin(\psi+\theta) & cos(\psi+\theta) \\ 0 & -cos(\psi+\theta) & sin(\psi+\theta) \\ 1 & 0 & 0 \end{bmatrix}
\end{aligned}
\tag{3.13}
$$

Notably, the angles ψ and θ here appear in pairs, which indicates the changing of ψ, and $\psi + \Delta$ is equivalent to θ, $\theta + \Delta$. The original pose is determined by θ, ψ and ϕ, but now θ and ϕ alone can control the orientation of objects, causing the loss of one degree of freedom. This change leads to some loss of orientation.

Since gimbal lock is caused by the three axial rotation of Euler angles, one solution is to represent an object's orientation in other ways. Using quaternions is one of the best methods avoiding gimbal lock. Through this approach, each orientation will be treated as a unique value or vector rather than as three related angles.

3.2.3 Quaternion

Quaternion is a complex-number-based system, that was first proposed by Irish mathematician William Rowan Hamilton in 1843 [35, 36].The quaternion system uses a spiral vector in a 3-dimensional complex coordinate system to construct a 4-dimensional new vector based on rotation angle, as $[\theta, x, y, z]^T$. This method links an object's the orientation to a unique complex number composed outside the rotating sequence, avoiding the gimbal lock. To construct a quaternion, we must initially define a 3-axial complex coordinate, as shown in equation (3.14).

$$i^2 = j^2 = k^2 = ijk = -1 \tag{3.14}$$

Scaling the unit vector, the quaternion,q, is denoted as equation (3.15), where w is the rotation angle.

$$q = w + x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k} = [w, x, y, z]^T \tag{3.15}$$

In this project, the quaternion can replace the spiral coordinate and express the rotation from inertial frame to body frame. Even if the original quaternion works in a complex number system, it can be used in the normal tri-axial coordinate and inherit its operation rules through analytic continuation. The following equation (3.16) records how to transfer coordinates from Euler angles into quaternion systems, where w, x, y and z satisfy equation (3.17).

$$\begin{aligned}
w_0 &= cos(\frac{\phi}{2})cos(\frac{\psi}{2})cos(\frac{\theta}{2}) + sin(\frac{\phi}{2})sin(\frac{\psi}{2})sin(\frac{\theta}{2}) \\
x_0 &= sin(\frac{\phi}{2})cos(\frac{\psi}{2})cos(\frac{\theta}{2}) + cos(\frac{\phi}{2})sin(\frac{\psi}{2})sin(\frac{\theta}{2}) \\
y_0 &= cos(\frac{\phi}{2})sin(\frac{\psi}{2})cos(\frac{\theta}{2}) + sin(\frac{\phi}{2})cos(\frac{\psi}{2})sin(\frac{\theta}{2}) \\
z_0 &= cos(\frac{\phi}{2})cos(\frac{\psi}{2})sin(\frac{\theta}{2}) + sin(\frac{\phi}{2})sin(\frac{\psi}{2})cos(\frac{\theta}{2})
\end{aligned} \tag{3.16}$$

$$w^2 + x^2 + y^2 + z^2 = 1 \tag{3.17}$$

Notably, all of the Euler angles are divided by 2 because the rotation in a quaterion can be represented as following equations (3.18), where P and P' refer to the initial vector and rotated vector. Here q^{-1} is the inverse (also called the conjugate in some references) of q. If the product of q is doubled, the final P' contains $cos^2\frac{\theta}{2}$ equivalent to $cos\theta$. The vector is also would be converted in this case into $[0, x, y, z]^T$. A quaternion like this one, with zero real component is called a pure quaternion.

$$P^{'} = qPq^{-1} \tag{3.18}$$

$$q^{-1} = w - x\boldsymbol{i} - y\boldsymbol{j} - z\boldsymbol{k} = [w, -x, -y, -z]^T \tag{3.19}$$

Unlike matrix multiplication, quaternion multiplication is complex because it contains both dot and cross products. It has been derived thoroughly in [35]. Based on [35], the rotation of the quaternion can be converted into a rotation matrix, as equation (3.20) demonstrates. Therefore, the equation $P^{'} = qPq^{-1}$ becomes $P^{'} = R_qP$, where R_q is normalized.

$$R_q = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & 1-2x^2-2z^2 & 2yz-2wx \\ 2xz-2wy & 2yz+2wz & 1-2x^2-2y^2 \end{bmatrix} \tag{3.20}$$

Using this quaternion-based rotation matrix, we can calculate the gravity estimate on body frame with equation (3.1). Here, the gravity $[0, 0, g]^T$ should be normalized as $[0, 0, 1]^T$.

$$G_q = R_q \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2xz+2wy \\ 2yz-2wx \\ 1-2x^2-2y^2 \end{bmatrix} \tag{3.21}$$

Plugging equation (3.17) into (3.20) reveals a more typical quaternion-based gravity vector:

$$G_q = \begin{bmatrix} 2xz + 2wy \\ 2yz - 2wx \\ w^2 - x^2 - y^2 + z^2 \end{bmatrix} = \begin{bmatrix} G_{x,n} \\ G_{y,n} \\ G_{z,n} \end{bmatrix} \tag{3.22}$$

If gravity is subtracted from the raw acceleration, the gravity-free acceleration vector can be defined as:

$$Acc_{g-free} = \begin{bmatrix} Acc_{x,raw} \\ Acc_{y,raw} \\ Acc_{z,raw} \end{bmatrix} - g \begin{bmatrix} G_{x,n} \\ G_{y,n} \\ G_{z,n} \end{bmatrix} \tag{3.23}$$

3.3 Magnetometer

The Magnetometer, the third sensor in IMU, can measure magnetic fields. It determines a magnetic field's strength via detecting the earth's magnetic fluctuation [37]. The magnetic field of the earth points to the north with an 11-degree angular bias. Since a magnetometer's readings originate from the earth, any processed direction it obtains, to some degree, is the absolute coordinate to the earth. Despite this, a magnetometer cannot work alone in indoor environment because it is vulnerable to external interferences, such as surrounding metals and other magnetic sources. Consequently, some integrated magnetometers in smartphones, officially named electronic compass, are placed in undisturbed location. Nonetheless, the use of magnetometer in 9-dof sensor fusion is regarded as an effective correctional

method.

Like other sensors, magnetometers provide three dimensions on x-y-z axes. After data processing, we can calculate an object's headings in relation to the strong magnetic field (usually to the earth's north) with a body frame reference. However, even if their headings are the same headings, objects may have different poses which should be identified by an accelerometer and a gyroscope. This method of improving the accuracy of a magnetometer by using other sensors is called tilt compensation. Therefore, the best approach is to use magnetometer only as a geostationary-based directional sensor to the earth's north. In this project, magnetometer readings produce a yaw angle which is combined with the roll and pitch angles from an accelerometer. The Euler angles from the accelerometer and magnetometer are then fused with the Eular angle from the gyroscope to obtain an accurate pose for the rotation matrix.

3.4 Calibration

In this project, calibration refers to calibrating the static data in three sensors to zero, which is a significant preliminary task in indoor positioning. Specifically, calibration attempts to eliminate the sensor's offset but achieves little in offsetting measurement noise, such as Gaussian noise and accumulated noise. These offsets usually exist when they are produced and may increase in external environments. For example, the offset of a magnetometer will be permanently magnified by a longstanding magnetic source.Accelerometers and gyroscopes may also have ad-

ditional circuitry reasons causing offsets. Therefore, each sensor has its unique calibration method; these methods will be explained later. After all the offsets are detected, they will be subtracted from their original readings. All corrections will be saved in the memory of the relevant sensor.

3.4.1 Accelerometer calibration

One common method of calibrating an accelerometer is called 6-sided calibration. This method uses gravity as the reference by rotating the side of the accelerometer to move each measured axis in and out of gravity [38]. During the calibration, the accelerometer should be statically placed on a flat plane, so that no Coriolis force affects the final results. Among the 6 sides, every two parallel sides match one axis, corresponding to positive and negative gravity. For each axis, 6-sided calibration can be divided into 3 steps: 1) Choose the plans where the measured axis is and collect readings for period with a proposed frequency; 2)Repeat step 1 for another parallel plane, and 3)Use equation (3.13) to calculate the actual accelerations. Since Gaussian noise exists in each time measurement, more samples help to average it out. Usually, 500 samples for each side (1000 for each axis) is an acceptable number.

$$
\begin{aligned}
A_{+g} &= \sum_{n=1}^{N1} a_{n,+g} \\
A_{-g} &= \sum_{n=1}^{N2} a_{n,-g} \\
\overline{B}_a &= \frac{A_{+g} + A_{-g}}{N1 + N2} \\
Gain &= \frac{A_{+g} - A_{-g}}{(N1 + N2) * g} \\
A_{act} &= \frac{A_{raw} - \overline{B}}{Gain}
\end{aligned}
\tag{3.24}
$$

Where:

- $A_{+g/-g}$ = Sum of acceleration of in and out of gravity for each axis
- $a_{n,+g/-g}$ = Acceleration of in and out of gravity for each time
- g = Standard gravity, $9.8 m/s^2$
- G = Scale factor of axis
- $\overline{B}$ = Average offset
- A_{raw} = Raw measured acceleration
- A_{act} = Actual acceleration

3.4.2 Gyroscope calibration

Gyroscope calibration uses the most straightforward method to calculate the mean of readings because it does not involve a scale factor. In the quiescent state,

readings of tri-axis gyroscope should be $[0, 0, 0]^T$. To calibrate the gyroscope, we placed it on a horizontal surface and kept it stationary. Following this, we recorded its readings at an appropriate frequency and calculated the mean of the readings, shown as equation (3.14). The average readings subtracted from the measured raw data revealed the actual angular velocity. The gyroscope is so sensitive to noise that 1500 samples for each axis were averaged out in the calibration.

$$\begin{aligned} \overline{\omega} &= \frac{1}{N1} \sum_{n=1}^{N1} \omega_n \\ \overline{B}_g &= \overline{\omega} - 0 \\ \omega_{act} &= \omega_{raw} - \overline{B}_g \end{aligned} \tag{3.25}$$

Where:

- $\overline{\omega}$ = Mean of angular velocity readings
- ω_n = Angular velocity for each samples
- $\overline{B}_g$ = Average offset of angular velocity
- ω_{raw} = Raw measured angular velocity
- ω_{act} = Actual angular velocity

3.4.3 Magnetometer calibration

Magnetometer calibration plays a crucial role in 9-dof navigation, as the magnetometer determines the headings (yaw in Euler angle) but its original readings are

usually inaccurate. Here we mainly aim at reducing the distortion caused by the inner chip rather than by external environments. In fact, internal distortion can be divided into 2 types: hard iron and soft iron distortion [39]. Hard iron distortion is usually produced by a constant and additive magnetic field to the earth's field; thus, it is characterized by a constant offset to the ideal case. In an IMU or other smart device, adjacent components, such as the accelerometor, may lead to this magnetic interference with the magnetometer. By nature, hard iron distortion is easy to eliminate. However, the soft iron distortion, resulting from materials that influence and distort the magnetic field, is not simply additive. This distortion mimics the interaction of both rotation and scaling. Figs. 3.2, 3.3 and 3.4 show the normalized diagrams. When calibrating, we keep one axis fixed and rotate the IMU along with this axis. This suggests that the rotation method rather than linear movement produces a magnetic field on yz, xz and xy surfaces, by selecting x,y and z as the rotation axes. Consequently, calibration in the magnetometer is based on a 2D surface rather than a 1D timeline. For different distortions, we have different calibration equations.

When calibrating for hard iron distortion, the main task is to determine the fix offset of each surface. This offset is defined as the distance from the measured center of a circle away from the origins; thus, this calibration can be implemented through left and right translation. Equation (3.15) demonstrates the offset calculation when the rotation axis is Z.

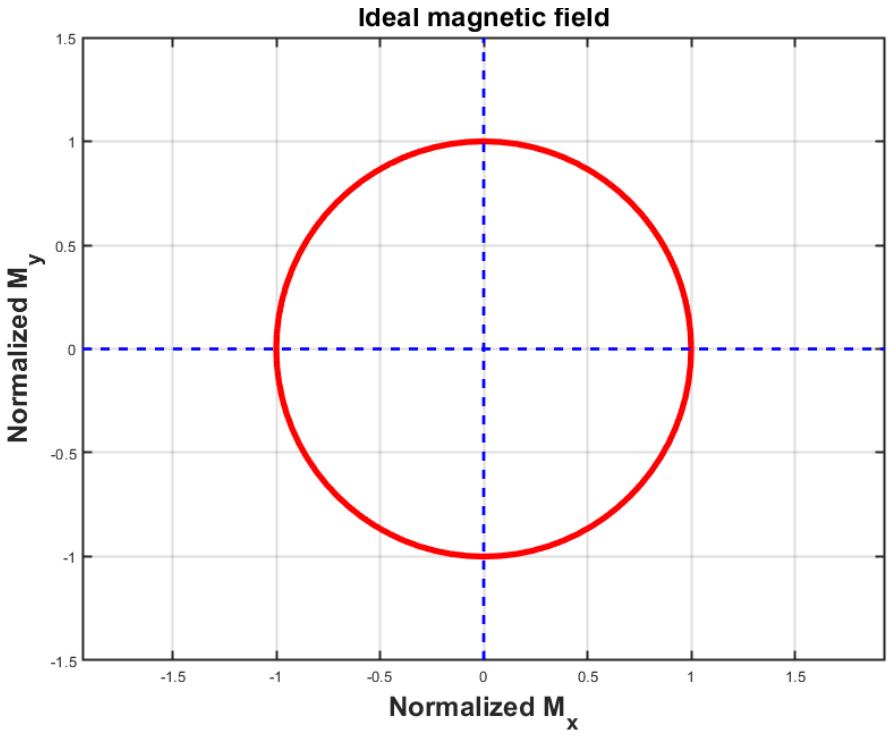


Figure 3.2: Ideal magnetic data free of distortion

$$\begin{aligned} \alpha &= \frac{x_{max} + x_{min}}{2} \\ \beta &= \frac{y_{max} + y_{min}}{2} \\ m_{x,act} &= m_{x,raw} - \alpha \\ m_{y,act} &= m_{y,raw} - \beta \end{aligned} \tag{3.26}$$

Where:

- α = Offset of x axis
- β = Offset of y axis

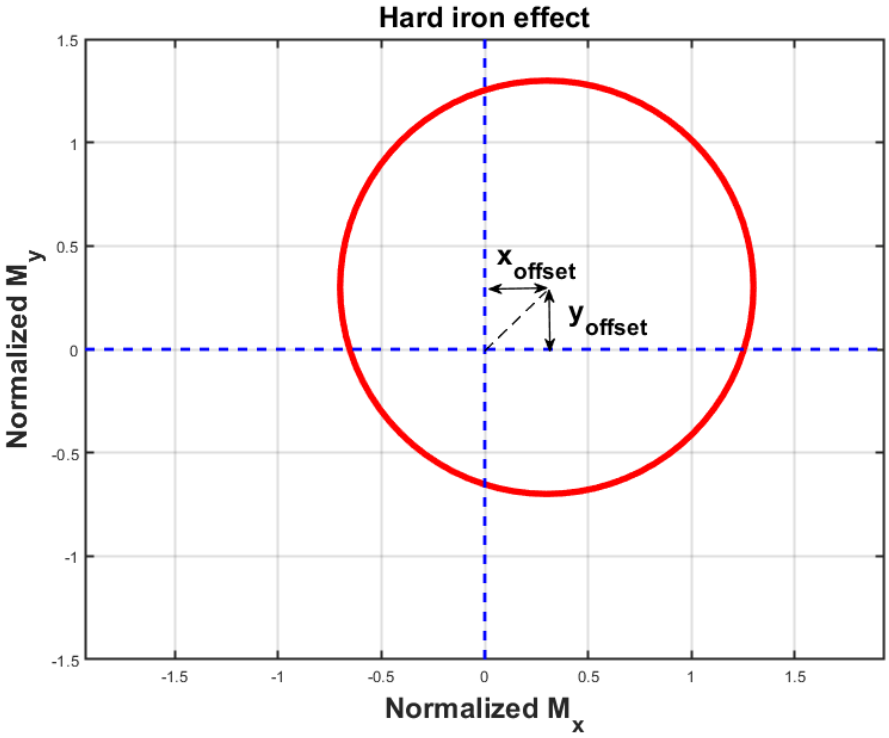


Figure 3.3: Hard iron effect magnetic data

- $m_{x/y,raw}$ = Raw measured magnetic field
- $m_{x/y,act}$ = Actual magnetic field

Unlike the calibration of hard iron distortion, the calibration of soft iron distortion requires additional parameters to modify the distorting ellipse to a unit circle. This calibration requires two steps: rotating and scaling. In the first step, the ellipse must be rotated, ensuring that it is in axial symmetry. Equation (3.16) shows the step of finding the rotation angle.

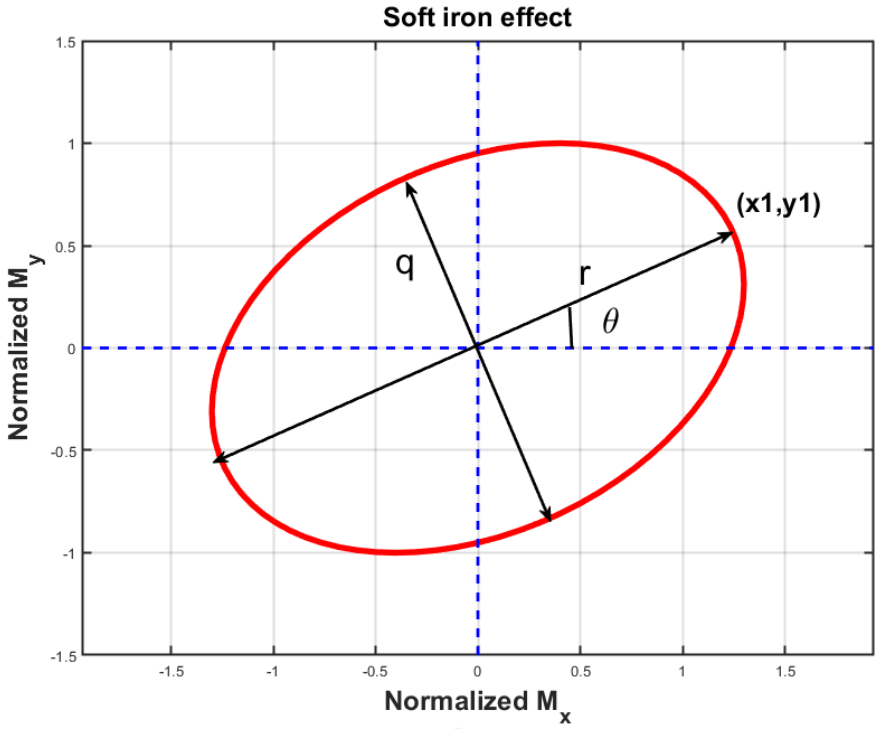


Figure 3.4: Soft iron effect magnetic data

$$
\begin{aligned}
r &= \sqrt{(x_1)^2 + (y_1)^2} \\
\theta &= arcsin(\frac{y_1}{r})
\end{aligned}
\tag{3.27}
$$

Once θ has been identified, the rotation matrix, $R(\theta)$, shown in equation (3.17) helps to transfer the magnetic curve into a symmetrical ellipse. $[x_{raw}, y_{raw}]^T$ refers to the unprocessed raw data vector, while $[x^*, y^*]^T$ refers to the rotated vector.

$$R(\theta) = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}$$
$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = R(\theta) \begin{bmatrix} x_{raw} \\ y_{raw} \end{bmatrix} \tag{3.28}$$

The next step is to scale the major axis to the minor axis so that the ellipse can be converted to a standard circle. Equation (3.18) defines the scaling factor symbolized as e, where q and r separately represents the ratio minor and major axis, respectively. Equation (3.19) shows that each x is multiplied by e to generate a circle.

$$e = \frac{q}{r} \tag{3.29}$$

$$\begin{bmatrix} x_{pro} \\ y_{pro} \end{bmatrix} = \begin{bmatrix} ex^* \\ y^* \end{bmatrix} \tag{3.30}$$

Once scaling is complete, the processed vector needs to be rotated back to the original position. In order to implement this, we apply equation (3.17) again, this time using $-\theta$, which aligns the processed ellipse and initial coordinate.

$$\begin{bmatrix} x_{act} \\ y_{act} \end{bmatrix} = R(-\theta) \begin{bmatrix} x_{pro} \\ y_{pro} \end{bmatrix} \tag{3.31}$$

Plugging equation (3.17) and (3.19) into (3.20) associates the measured raw vector, $[x_{raw}, y_{raw}]^T$, with the calibrated actual vector,$[x_{act}, y_{act}]^T$, shown as equation (3.21).

$$\begin{aligned}
\begin{bmatrix} x_{act} \\ y_{act} \end{bmatrix} &= R(-\theta) \begin{bmatrix} ex^* \\ y^* \end{bmatrix} = \begin{bmatrix} cos(-\theta) & sin(-\theta) \\ -sin(-\theta) & cos(-\theta) \end{bmatrix} \begin{bmatrix} ex^* \\ y^* \end{bmatrix} \\
&= \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} ex_{raw}cos\theta + ey_{raw}sin\theta \\ -x_{raw}sin\theta + y_{raw}cos\theta \end{bmatrix} \\
&= \begin{bmatrix} ex_{raw} + (1-e)x_{raw}sin^2\theta + (e-1)y_{raw}sin\theta cos\theta \\ ey_{raw} + (1-e)y_{raw}cos^2\theta + (e-1)x_{raw}sin\theta cos\theta \end{bmatrix}
\end{aligned} \tag{3.32}$$

3.5 Sensor fusion

As mentioned previously, sensor fusion aims at improving the reliability of positioning prediction, since the measured data is collected not merely from a single sensor but from three individual ones. In this project, the fused sensors were used to estimate the orientation of the object, which was transferred into an orientation matrix to eliminate gravity and Coriolis force. Figures 3.2 and 3.3 briefly describe the step-by-step procedure of position estimation.

Mainly, the proposed indoor positioning is comprised of two modules: processed acceleration module and integrator module. The acceleration module can

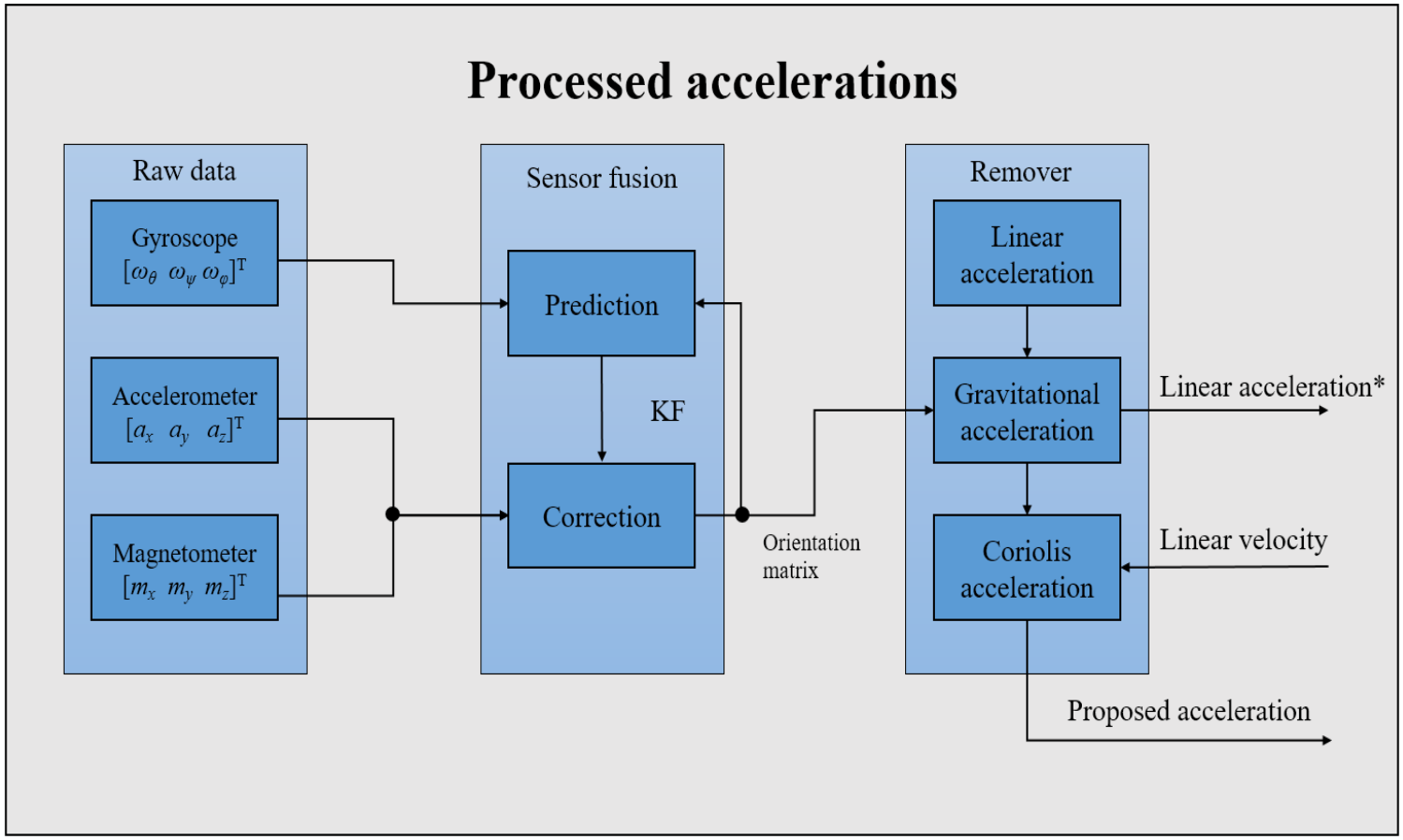


Figure 3.5: Processed acceleration removes the gravity and Coriolis force.

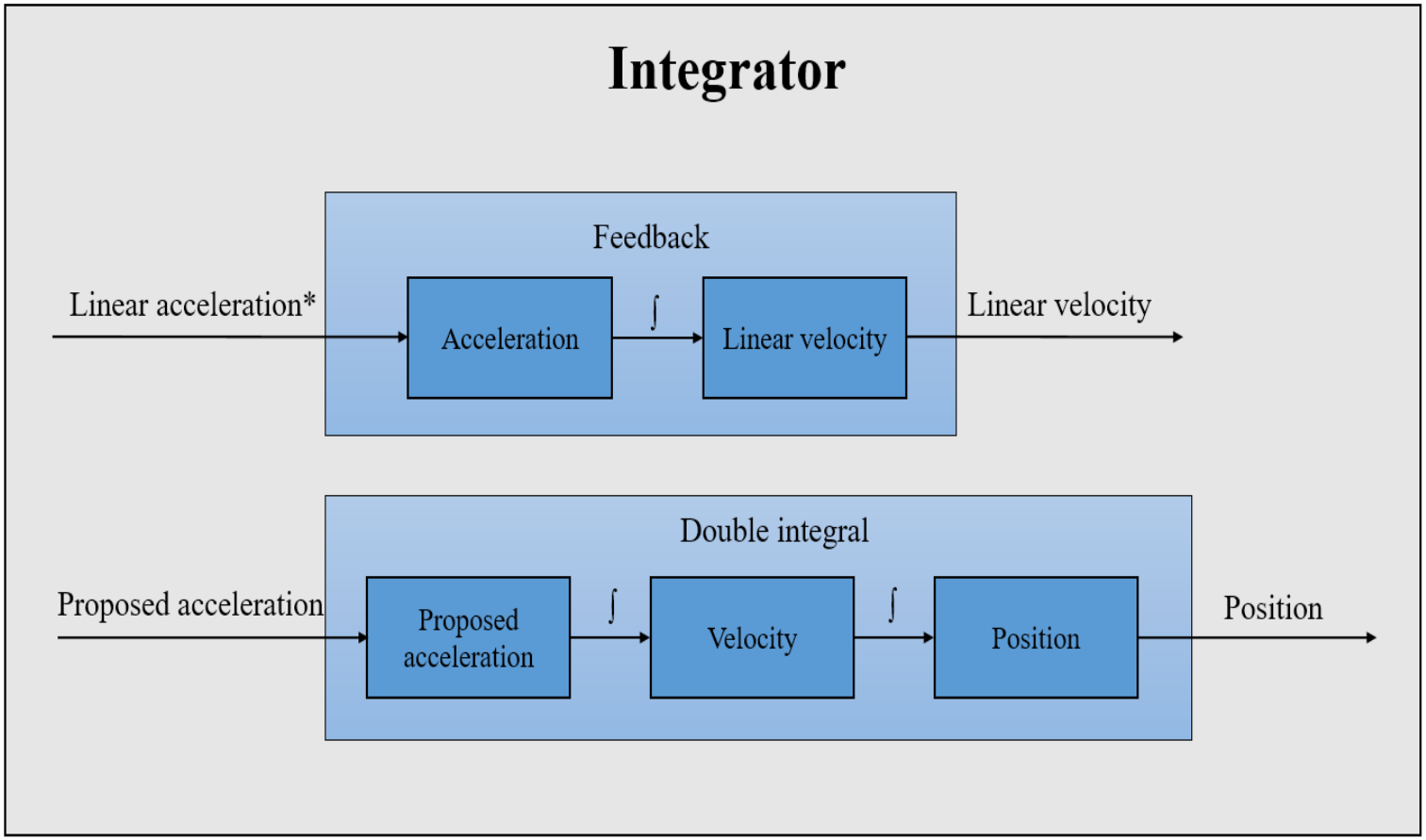


Figure 3.6: Integrator calculates the final position.

be divided into three stages: raw data, sensor fusion, and remover. The raw data stage involves data collection from the accelerometer, gyroscope, and magnetometer, while the sensor fusion enables the computation and optimization of an orientation matrix through Kalman filtering. Finally, the remover stage aims at eliminating the redundant gravitational and Coriolis accelerations from the linear acceleration, as demonstrated in the equations from (3.1) to (3.3). The final results of this module are the non-gravity linear acceleration symbolized as '$*$' and the proposed acceleration. Since the removal of Coriolis force requires linear velocity, the non-gravity linear acceleration must be integrated first.

Similarly, the integrator module is separated into feedback and double integral processes. The feedback process can produce linear velocity without gravity that can be used in the remover stage. The double integral process helps to create the final position based on calculated acceleration by integrating twice.

3.5.1 Magnetometer and yaw angle

Sensor fusion is mainly comprised of two sections: prediction and correction. Prediction is computed by a single integration of angular velocity from a gyroscope, as shown in equations (3.6) to (3.8). Correction uses accelerometer and magnetometer to jointly provide additional roll, pitch, and yaw angles. Using the Kalmen filter or the complementary filter [33, 37, 40], the predicted orientation can be constantly modified based on maximum likelihood [20]. Additionally, the accelerometer fails to provide the yaw angle because the surface of yaw is completely perpendicular

to gravity. This means there is no gravitational projection on that surface. In that case, an accelerometer may show the identical readings for all orientations regardless of yaw, as long as they separately share the same roll and pitch angles [Zhang]. This is also the drawback of 6-dof indoor navigation: it may exhibit yaw drift to some degree. The introduction of a magnetometer solves this problem since this sensor, on a yaw surface, points to the earth's north. Specifically, data from a magnetometer can derive the angle between the actual heading and the standard north, which defines the yaw angle. Equations (3.13) formulate the Euler angles computed by inverse trigonometric functions [33, 40]. At this step, all preparations are finished; next, we use the Kalman filter to fuse these datasets with another one directly computed by a gyroscope.

$$Roll, pitch, yaw = \begin{cases} \theta_a = arctan(\frac{a_y}{a_z}) \\ \phi_a = arctan(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}) \\ M_x = m_x cos(\phi_a) + m_z sin(\phi_a) \\ M_y = m_x sin(\theta_a) sin(\phi_a) + m_y cos(\theta_a) - m_z sin(\theta_a) cos(\phi_a) \\ \psi_m = arctan(\frac{M_y}{M_x}) \end{cases} \tag{3.33}$$

Where:

- $\theta_a/\phi_a/\psi_m$ = Roll, pitch and yaw from an accelerometer or a magnetometer
- $a_{x/y/z}$ = Acceleration measured by an accelerometer

- $m_{x/y/z}$ = Magnetic field measured by a magnetometer

3.5.2 The Kalman filter

The Kalmen filter is an algorithm that uses the linear system state equation to optimally estimate the system state through input and output observation data. Since the observation data includes systematic noise and interference, this optimal estimation can be regarded as a filtering process. This filtering estimates the state of a dynamic system from the raw data with noise when the measurement variance is known. In light of its convenience in computer programming and real-time updating mechanisms, the Kalman filter has been frequently applied in many fields, such as navigation, vehicle control, and aerospace [41].

The Kalman filter works in a two-step process that includes prediction and correction. Equation (3.14) describes prediction and correction modules [42]. The complete configuration of the Kalman filter for indoor navigation is also shown in Figure 3.4. With regard to the prediction module, the Kalman filter estimates the current system state with noise. In this project, the gyroscope provides the angular velocity input and measurement noise(such as the drift mentioned previously). Based on this information, the time update in prediction determines the initial error covariance matrix. Plugging in the error covariance, linear acceleration, and magnetic field vector as another the input of correction inputs helps in computing the Kalman gain. The next step is to update the estimation and error covariance, using the Kalman gain. At this point, the corrections for the current state are

complete. For the next state, K+1, the current estimation and error covariance matrix will inherit the K-1 state in the prediction module. Therefore, the Kalman filtering is a recursive system, and the final result will be close to the true value due to this constant circulation.

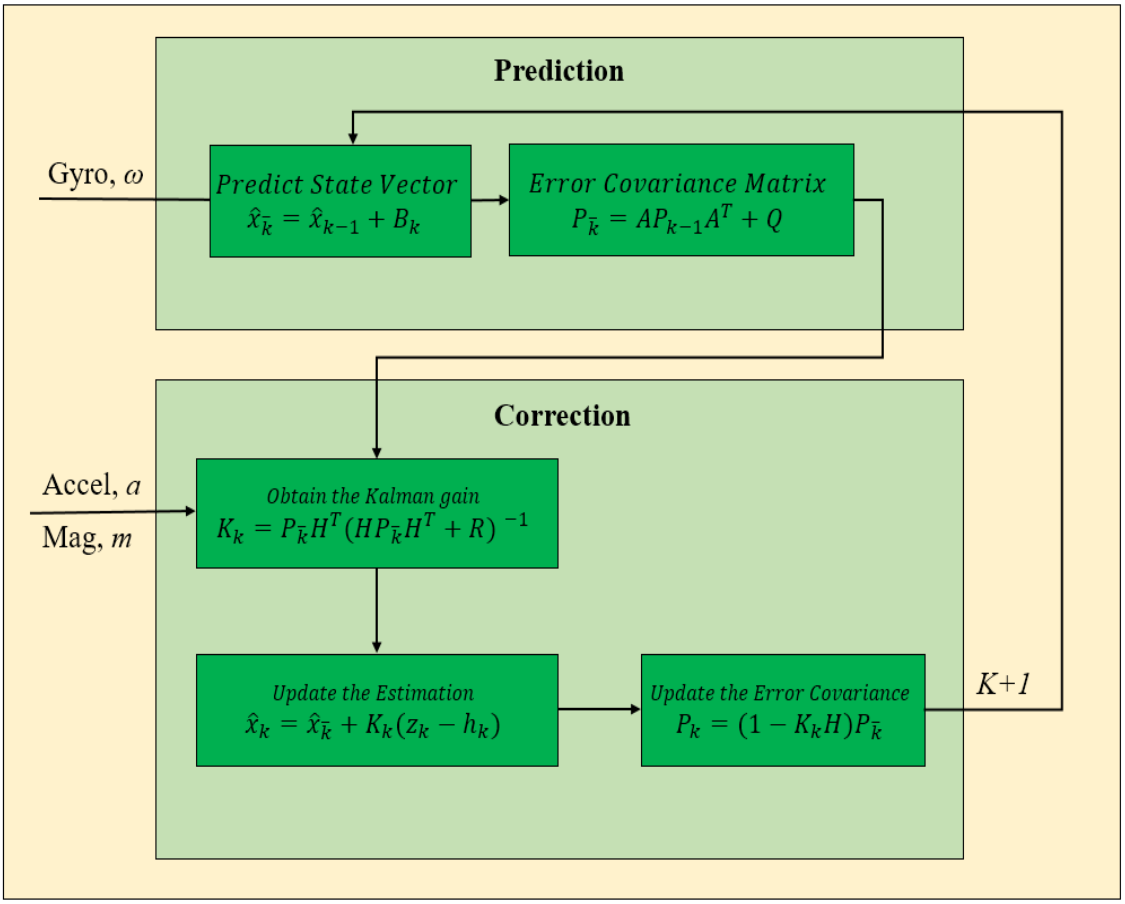


Figure 3.7: Kalman filter

$$
\begin{aligned}
Prediction = & \begin{cases} \hat{x}_{\overline{k}} = \hat{x}_{k-1} + B_k \\ P_{\overline{k}} = AP_{k-1}A^T + Q \end{cases} \\
Correction = & \begin{cases} K_k = P_{\overline{k}}H^T(HP_{\overline{k}}H_T + R)^{-1} \\ \hat{x}_k = \hat{x}_{\overline{k}} + K_k(z_k - h_k) \\ P_k = (I - K_kH)P_{\overline{k}} \end{cases}
\end{aligned} \tag{3.34}
$$

Where:

- $\hat{x}_{\overline{k}}$ = Predicted state vector including 3-axis coordinates with heading
- $\hat{x}_k$ = Corrected state vector
- B_k = State space model matrix
- $P_{\overline{k}}$ = Predicted error covariance matrix comprising: (i)Q = Predicted error noise. (ii)A = State matrix
- P_k = Corrected error covariance
- K_k = Kalman gain
- H = Measurements

3.6 SLAM

Simultaneous Localization and Mapping (SLAM), a navigation algorithm fusing both inertial sensors and image depth, has been deployed in numerous platforms

on account of its accuracy and robustness since its origination in the 1980s [43, 44]. Unlike train-to-form algorithm based on processed data, SLAM seeks to form a global and consistent estimate of an object's real-time path when reconstructing surroundings [45]. According to its definition, SLAM contains two tasks, namely localization and mapping. The goal of localization is to determine the amount of movement by collecting information such as camera images. With regard to mapping, it has a focus on generating a map of obstacles or typical beacons in an object' surroundings [45]. To implement these two goals, SLAM requires not only reliable built-in IMUs for path calculation but also external cameras or laser sensors to simultaneously detect around obstacles.

The main dection methods of SLAM can be broadly classified into two categories: visual SLAM and Light Detection and Ranging SLAM. Visual SLAM (also called vSLAM) concentrates on using RGB or depth images acquired from camera and other image sensors. Different types of cameras are found to be competent, for example, the simple cameras (fish-eye or wide-angle), compound eye cameras (stereo camera), and RGB depth cameras have been successfully applied to vSLAM navigation [45]. It is because vSLAM's has powerful compatibility with cameras that it can be implemented with inexpensive image sensors. A typical device using vSLAM usually includes a binocular camera, an IMU, and a vision processing unit (VPU) to run the algorithms. In addition, vSLAM algorithms can be subdivided into sparse methods and dense methods. Sparse methods seek to match of feature points in images, while dense methods use overall image brightness.

Unlike vSLAM, Light Detection and Ranging SLAM (also called LiDAR SLAM)

primarily employs the lasers or other distance sensors. Thanks to precise laser, the application scenarios of LiDAR SLAM can even include high-speed moving vehicles like self-driving cars. In detail, LiDAR SLAM outputs 2D or 3D point cloud data to precisely measure distance and construct a local map. By matching this cloud data, we can sequentially estimate an object's movement [45]. However, simple road conditions such as minor obstructions can cause this point cloud data to fail to provide high-density images for sufficient feature matching. Also, point cloud requires high-performance processor to optimize and rapidly process large data streaming, which invisibly increases its cost.

Chapter 4: Experimental Results

4.1 Hardware Components

In general, our experimental hardware components mainly fall into two categories: IMUs and development boards. All IMUs in this project are 9-dof sensors containing a 3-axis accelerometer, a gyroscope and a magnetometer. Their sepecification are only sightly different, and therefore, their theoretical performances are close. The development board is used for data reading from IMU sensors and connection with laptops. The device equipping visual camera is also indispensable to conduct the SLAM algorithm.

4.1.1 IMUs

In this project, three different IMUs which commonly work for indoor navigation are selected: LSM9DS1, ICM20948, and BNO055. LSM9DS1 is a 9-dof sensor containing a tri-axial accelerometer, a magnetometer, and a gyroscope. Its 3-axis accelerometer has a sensing range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$, while the gyroscope has three gears of $\pm 245/\pm 500/\pm 2000$dps to adjust the rotation sensitivity. For its magnetometer, LSM9DS1 also has different ranges of ± 4, ± 8, ± 12 and ± 16gauss. In addition, LSM9DS1 wires up I^2C data on SCL and SDA pin, and it can also work in the range of 3 to 5V.

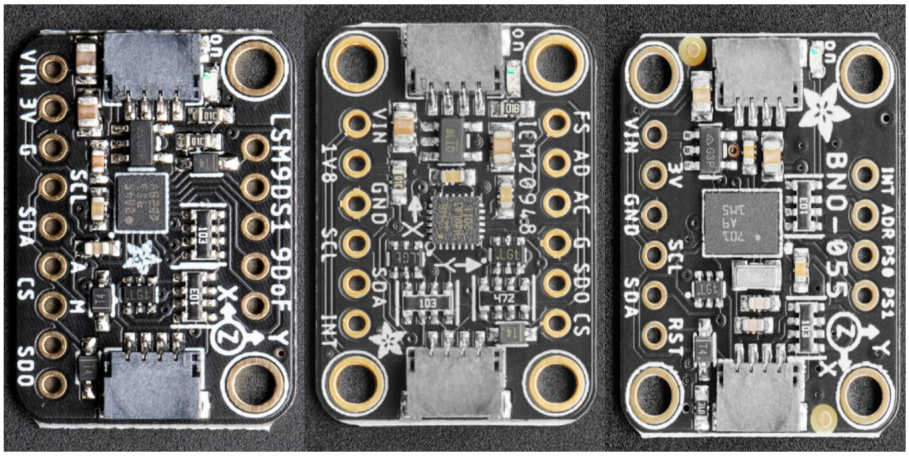


Figure 4.1: IMUs: LSM9DS1(left), ICM20948(middle) and BNO055(right)

ICM20948 also integrates a 3 axis micro-electro-mechanical system (MEM) accelerometer, magnetometer and gyroscope. As with LSM9DS1, the accelerometer in ICM20948 has a full scale range (FSR) of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$. Its 3-axis gyroscope has better range than LSM9DS1 with Programmable FSR of $\pm 250/\pm 500/\pm 1000/\pm 2000dps$. The range of its magnetometer can also vary from $-4900\mu T$ to $+4900\mu T$, where $1000\mu T$ equals 10 guass ($4900\mu T$=49 guass). ICM20948 also support I^2C protocol, and runs on both 1.8V and 3V, consuming less power than LSM9DS1.

Among these IMUs, BNO055 is the most powerful, as the 9-dof absolute orientation sensor. Its accelerometer shares the specification of the previous two IMUs,with the range of $\pm 2g$, $\pm 4g$, $\pm 8g$ and $\pm 16g$. Besides, it includes a switchable ± 125-to-$\pm 2000dps$ gyroscope. Its magnetometer has different specifications on different axes, that is, $\pm 1300\mu T$ on the x and y axes and $\pm 2500\mu T$ on the z

axis. With I^2C support, its safe logic voltage is from 3 to 5V. Meanwhile, BNO055 is a highly specialized IMU that can directly output four-point quaternions and linear velocity free of gravity, significantly reducing the workload.

4.1.2 Development Boards and Screens

Development boards are designed to read and save data for MEM sensors. A development board usually contains a flash memory module and RAM module with a microprocessor unit (MPU), which means the calibrated offset compensation can be directly saved and subtracted from the original readings. For this project, we selected two different development boards: STM32F405 and TTGO ESP32.

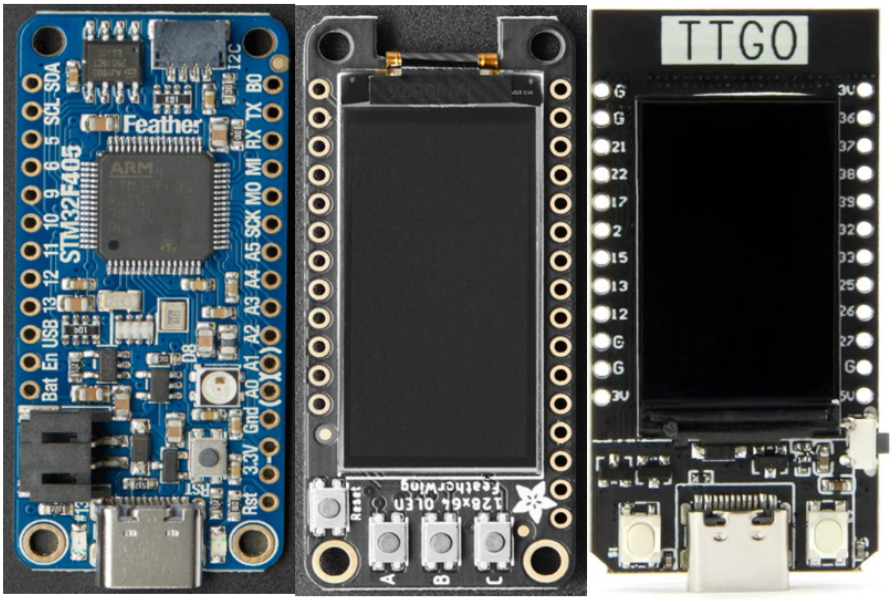


Figure 4.2: STM32F405(left), Feather OLED(middle) and TTGO(right)

As its name suggests, STM32F405 is a member of the STM32 series that focuses on high-performance, low-cost, and low-power microprocessor units (MCU) using ARM architecture. This board is equipped with a Cortex M4 MPU at 168MHz with a 1MB flash, and can operate on both 3.3V and 5V. Additionally, it supports I^2C serial bus and UART protocol, both of which are used in this project. It also includes a STEMMA-QT connector for the fast I^2C connectivity. Through this connector, the IMUs that also have this connector can be linked with the STM32F405 board without soldering. However, this development board does not include a visual screen or Bluetooth antenna; thus, it needs to be connected with a laptop during measuring. This create inconvenience for data collection.

The other development board is from the ESP32 series: TTGO ESP32. It is equipped with a 240MHz Xtensa dual-core MCU. Compared with the STM32 board, ESP32 has much larger flash memory, up to 4MB, and its working voltage ranges from 2.7V to 4.2V. Likewise, ESP32 also supports I^2C and UART protocols for user-friendly data transmission and programming. Meanwhile, the specialized version of ESP32, TTGO, has a 1.14 inch IPS screen with 135×240 resolution that can show real-time readings after programming, a feature that brings a significant convenience to the adjustment and testing process.

In order to implement visual readings like TTGO, we chose an OLED screen called FeatherWing to match the STM32F405 board. This monochrome OLED has 128×32 resolution with 3 programmable buttons to control the screen. However, the display is bi-colored and small, merely about 1 inch diagonal, but provides highly readable readings due to its high contrast.

4.1.3 Tracking Camera

As mentioned before, a visual device is needed to run the SLAM algorithm. Considering both cost and perfomance, Intel RealSense T256 was chosen for this research. This tracking camerca is equipped with two fisheye lens sensors with a BMI055,6-dof IMU, and Movidius Myriad2 video processing unit (VPU), all of which supports low-latency and power-efficient vSLAM algorithms.

Figure 4.3: Intel RealSense T265 tracking camera

Notably, T256 is not equipped with a depth camera but does have a high view black and white one: it sacrifices the optimized depth accuracy for field of view [46]. Likewise, T265 has no Bluetooth or similar wireless module; thus, its connection to the laptop must be maintained during data measuring.

4.2 Sensor Testing

4.2.1 Accelerometer Calibration

Through the 6-sided calibration shown in equation (3.24), the calibrated gravity for each axis is demonstrated in Figures (4.4) to (4.6). Notably, the precise standard gravity in Oregon is $9.75 m/s^2$; thus, g in equation (3.24) equals that number rather than 9.8.

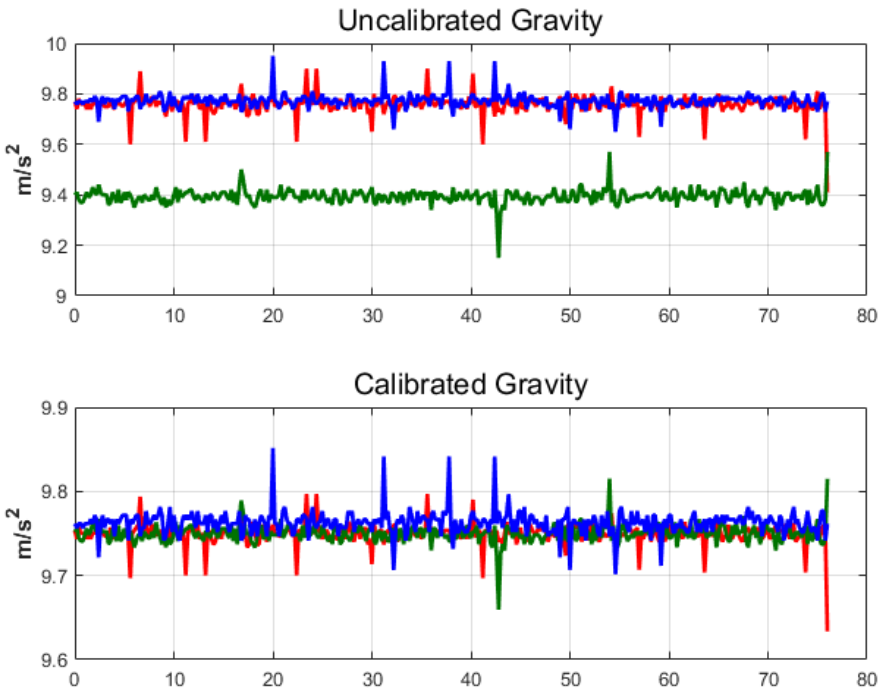
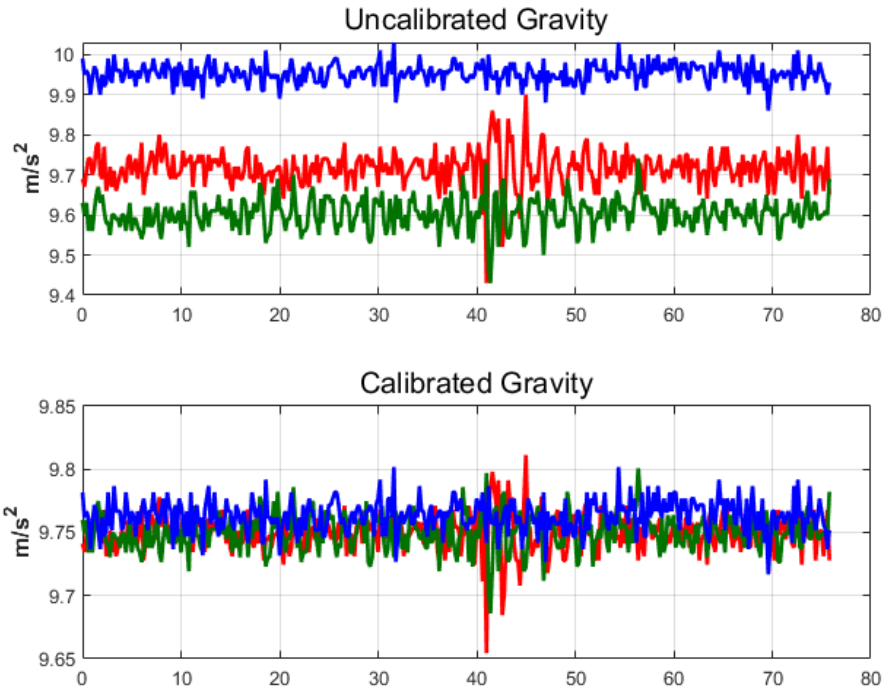
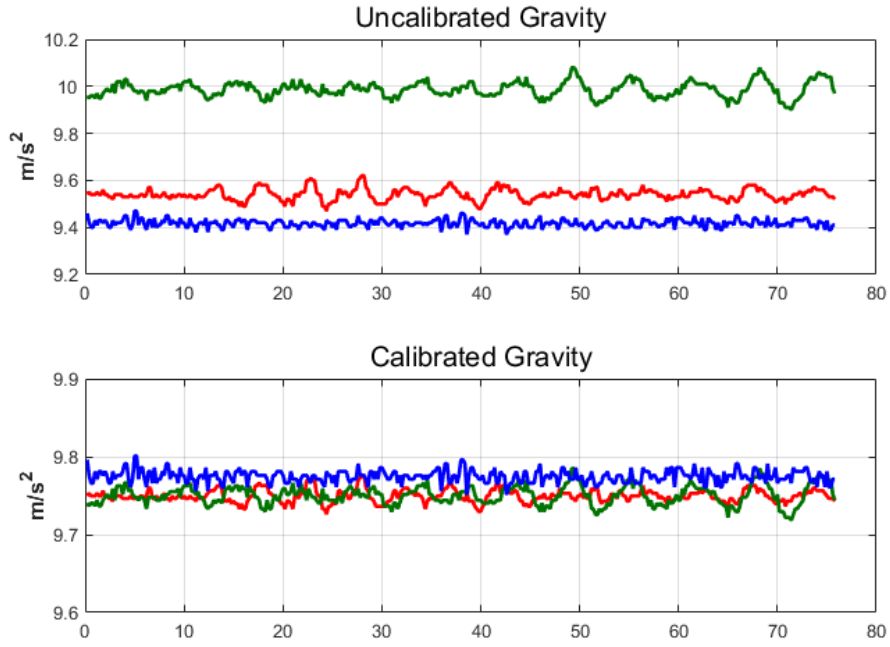


Figure 4.4: Accelerometer calibration for LSM9DS1

After calibrating, the average tri-axial readings are all close to 9.75. Since calibration cannot remove the Gauss noise in the accelerometer, electric noise still



(a) ICM20948



(b) BNO055

Figure 4.5: Accelerometer calibration for ICM20948 and BNO055

exists. As the figures show, relatively speaking, BNO055 has the best performance against internal electric noise, while LSM9DS1 has acceptable performance, but the loudest noises still cause spurts. ICM20948 had a moderate noise performance when measured.

4.2.2 Gyroscope Calibration

No scaling factor consideration makes the calibration of the gyroscope simple. Based on equation (3.25), Figures 4.4 to 4.6 demonstrate both uncalibrated and calibrated gyroscope readings for three IMUs. In these figures, the red, green, and blue waves respectively represent angular velocity at x, y, and z directions.

As Figures 4.5 and 4.6 show, ICM20948 had a tri-axial offset vector [0.0032, -0.0185, $-0.0122]^T$, while BNO055's offset vector was $[0.0037, , -0.0154, -0.01225]^T$ before calibrating. The offset of LSM9DS1 drifted most with vector $[0.0771, 0.0842, 0.1717]^T$. Figure 4.4 indicates that all offset in LSM9DS1 was positive, which may be caused by its different process corner. After compensation, the average readings of all three axes have returned zero. Notably, calibration cannot help to eliminate the Gauss noise, leading to this fluctuating waveform.

4.2.3 Magnetometer Calibration

Figures (4.9) to (4.14) compare highlight difference between uncalibrated and calibrated magnetic fields. Due to the interference-free environment, no apparent soft

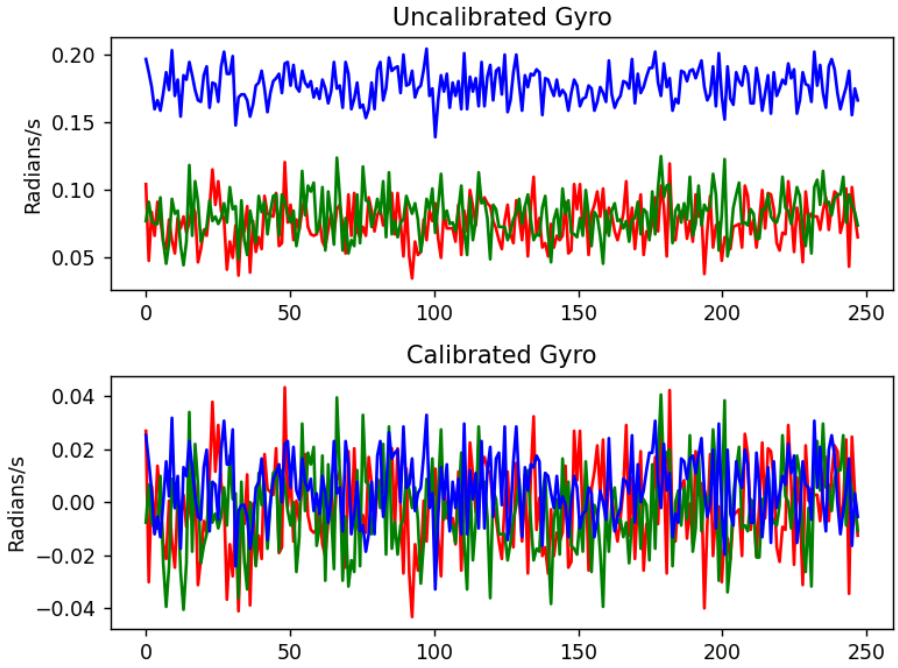
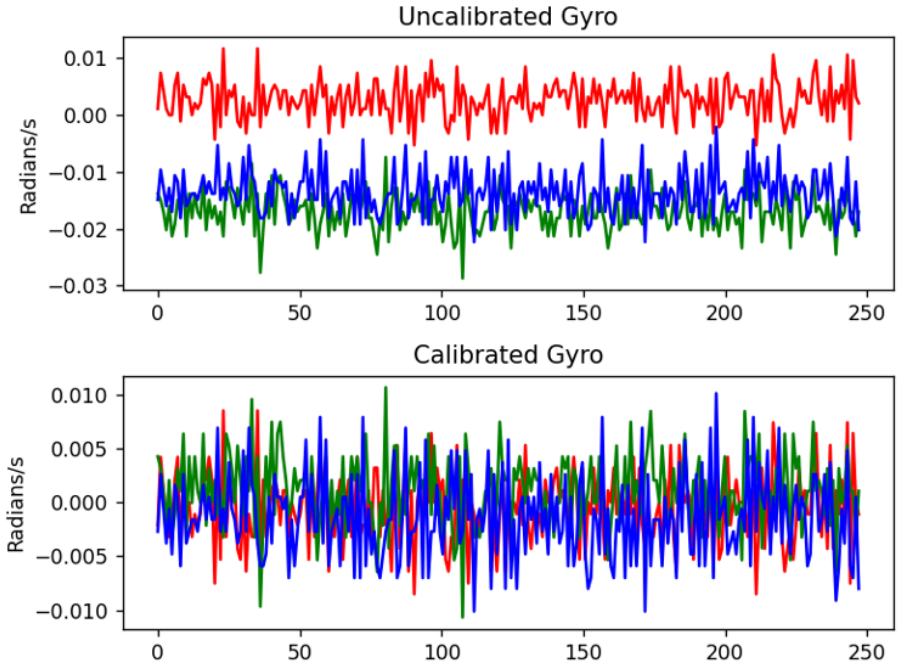


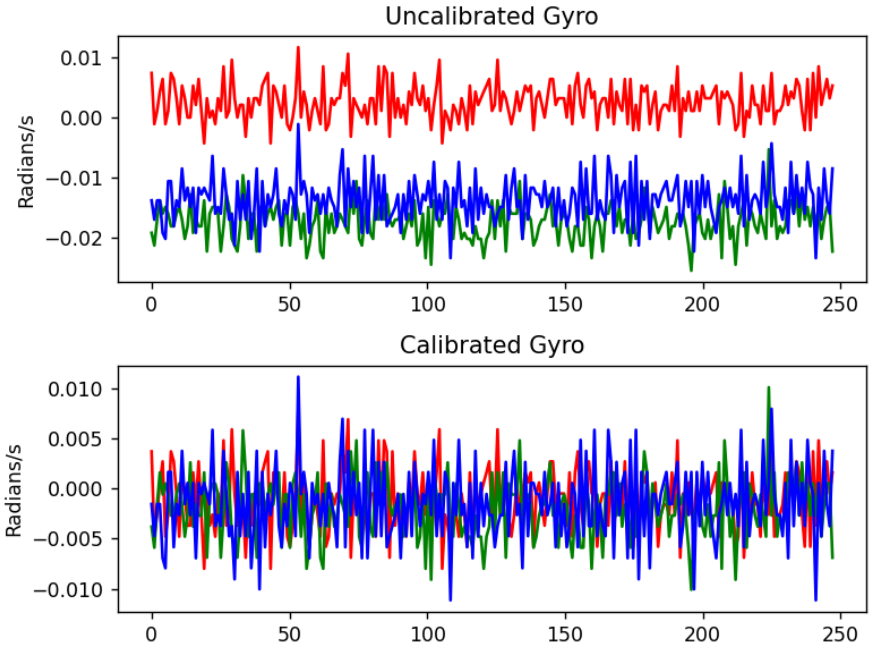
Figure 4.6: Gyroscope calibration for LSM9DS1

iron effect was detected: thus, all the calibrations aimed at eliminating the hard iron effect.

According to equation (3.26), the magnetic fields in all IMUs are calibrated and re-centered at the origin. Before the compensation, all IMUs have an obvious offset or two possible reasons. One reason suggests the offset be caused by processing when manufacturing, while another involves the development board. Since all the IMUs are connected with the board for data transmission, the electric component nearby may cause this fixed interference, indicating that the testing environment should not include a strong electrical source .

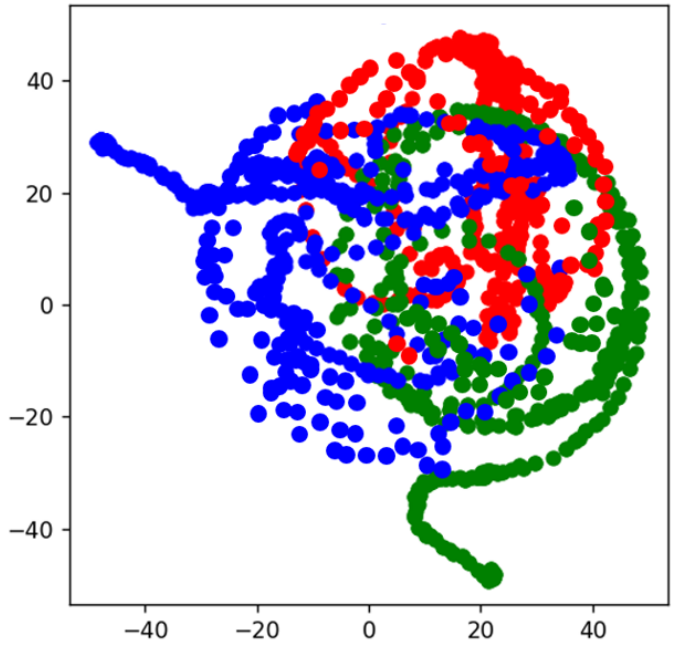


(a) ICM20948

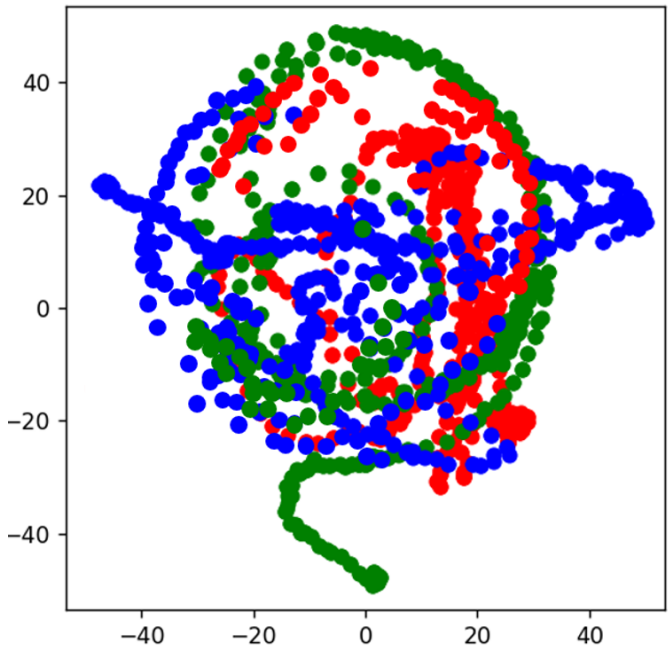


(b) BNO055

Figure 4.7: Gyroscope calibration for ICM20948 and BNO055

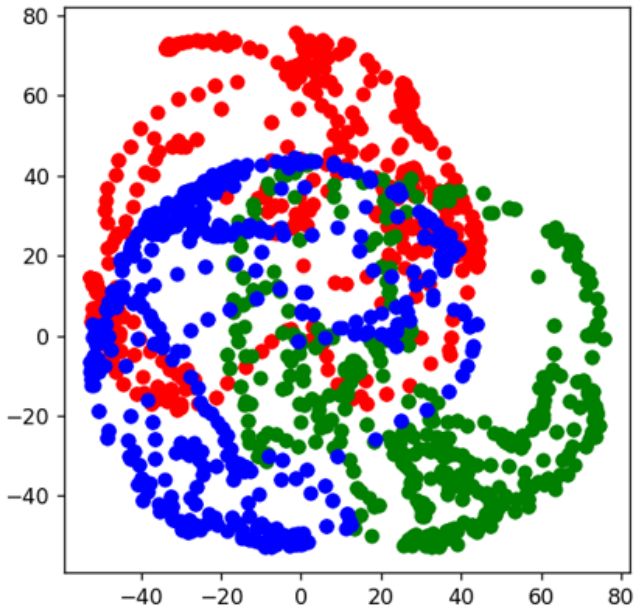


(a) Uncalibrated

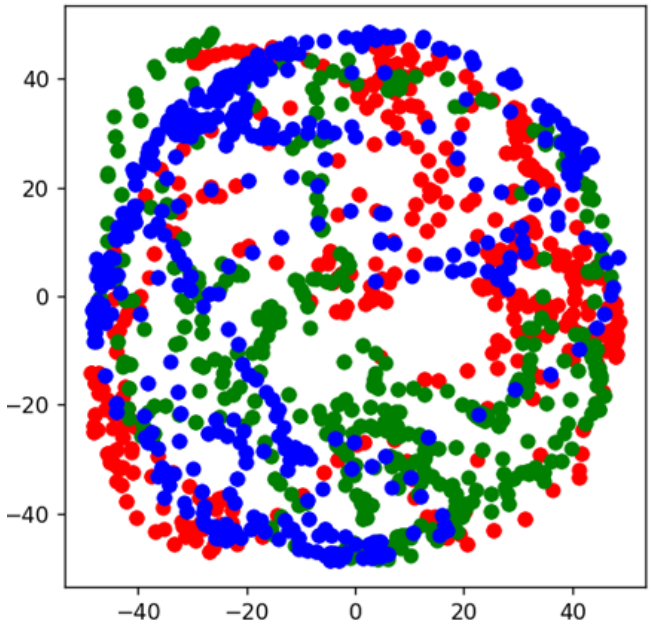


(b) Calibrated

Figure 4.8: Magnetometer calibration in LSM9DS1

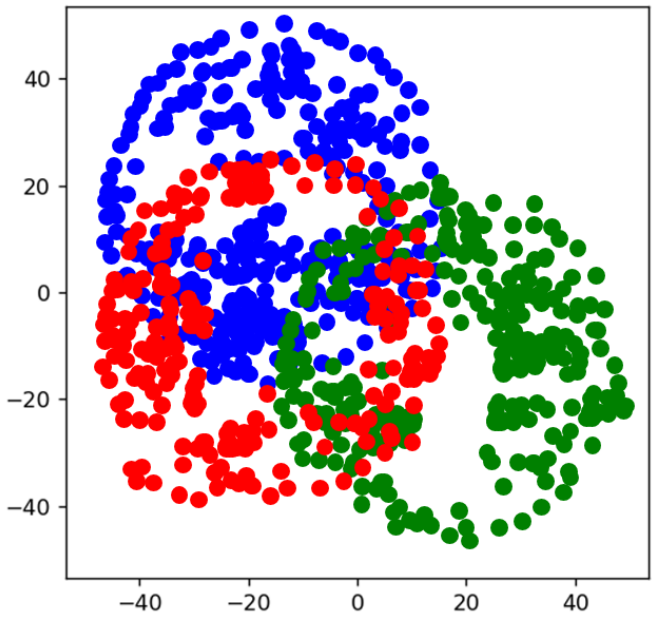


(a) Uncalibrated

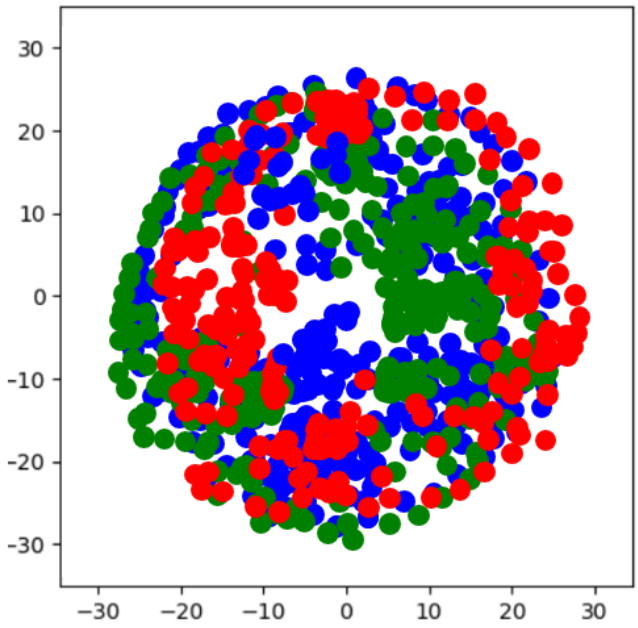


(b) Calibrated

Figure 4.9: Magnetometer calibration in ICM20948



(a) Uncalibrated



(b) Calibrated

Figure 4.10: Magnetometer calibration in BNO055

4.3 Trajectory Test

The total trajectory test is segmented into 2D and 3D sections to simulate the human's activities in an indoor environment. The first trajectory records the motion when the user leaves from the balcony and picks up an object off the floor. In fact, the stop time when the user picks up an object so brief that it can be ignored; its purpose is whose purpose is, to facilitate the investigation of a turn scenario. Alnother 3D experiment simulates a scenario of the user walking down the stairs.

4.3.1 2D Trajectory

Figure 4.11 shows the top view of our house with the proposed walking trajectory. It indicates the existence of three checkpoints for the user to turn around and slow down. The longitudinal length is nearly 5.5m, while the transverse length is estimated as 2.4m. The sample rate of IMUs is set to be 100Hz, while T265 provides a fixed 60Hz output. After the measured data from four devices has been processed, the final trajectory is shown in Figure 4.12(a).

In general, all devices match the user's actual movement from the balcony to the front door. Since T265 can provide the greatest precision and noise-free tracking, it acts as the ground truth to compare with IMUs. Among all three IMUs, BNO055 has the best performance for movement tracking. However, gyro-drift makes LSM9DS1 deviate from the proposed direction when the user passes by the second check point. The maximum deviation occurs in ICM20948, which happens

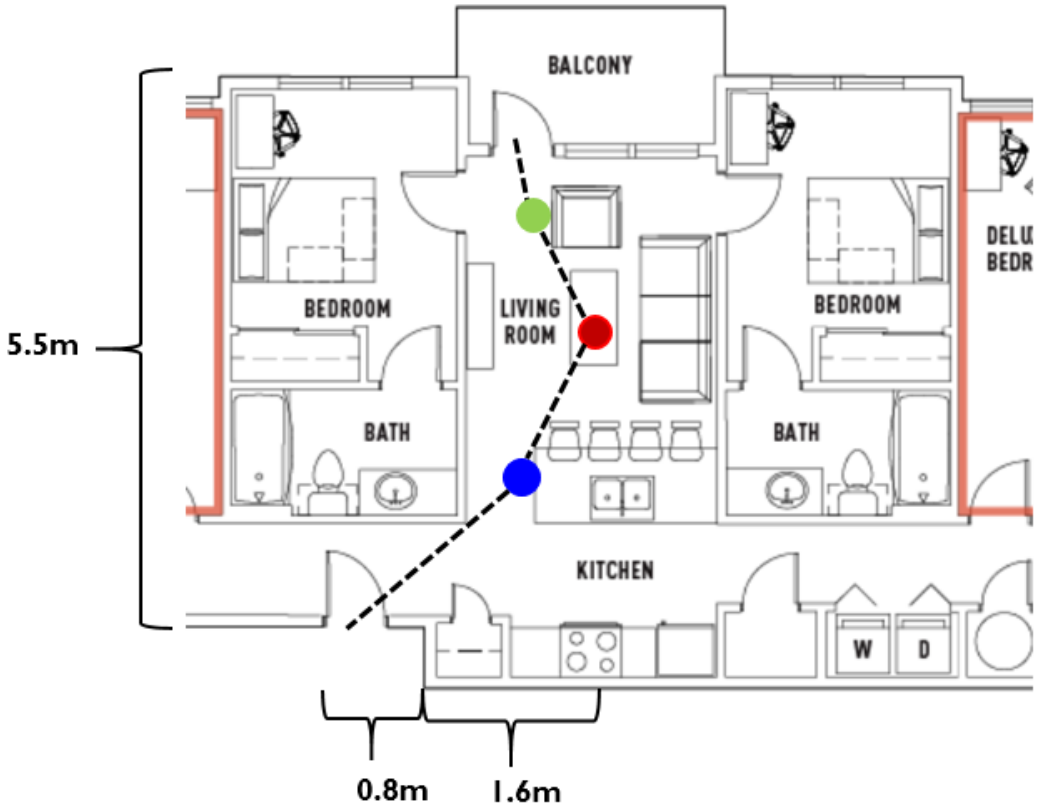
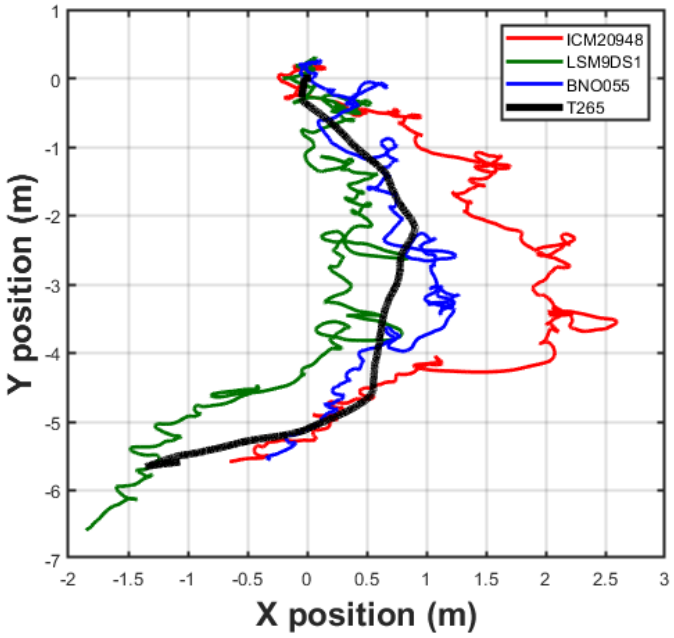
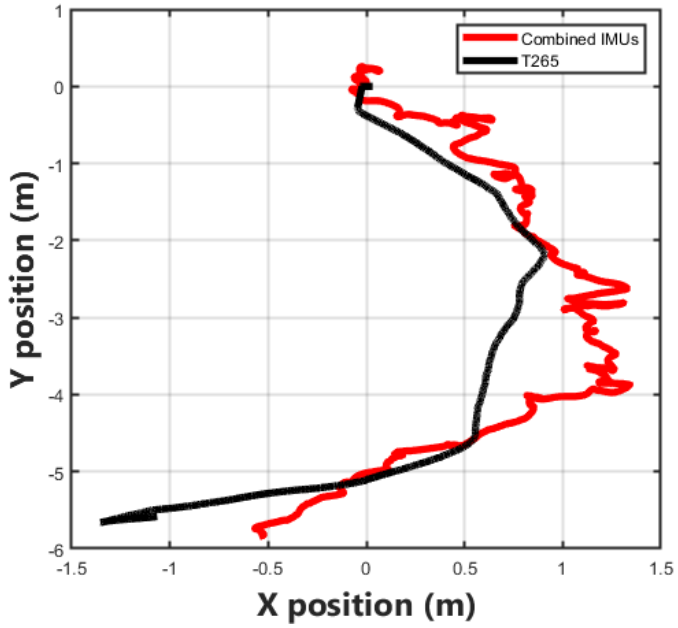


Figure 4.11: Top-view of 2D trajectory

nearly at the beginning of our measurement. The accumulative errors caused by high frequency noise also lead to uneven trajectories. To overcome this problem, all sensor readings are combined to average out accumulative noise. Figure 4.12(b) presents the processed result based on the fusion of IMUs. Compared with the single trajectory from 3 independent IMUs, the fused result looks more smooth and reliable, but drift caused by the gyroscope still exists.



(a) Separate tracking



(b) Combined tracking

Figure 4.12: 2D Trajectory

4.3.2 3D Trajectory

The 3D trajectory simulates the downstairs motion of a human, and Figure 4.13 shows the associated location. The motions in x and y directions are straight, so no checkpoint has been set in this section. The 3D trajectory aims at exploring how vertical vibration (in the z direction) affects the final reading. Figure 4.14 demonstrates the estimated results of the 3 IMUs and the T265 camera. Like 2D simulation, BNO055 offers the best estimation compared with T265, while ICM20948 and LSM9DS1 appears to have varying degrees of deviation on xy plane. Due to the straight route, there appears to be less deviation problem on the 3D Trajectory than on the 2D trajectory. Going in the z direction, all three IMUs correctly estimate the height of stair. One potential reason is that gravity accounts for a large proportion of the z-directional raw data, suppressing gyro-drift. Also, the sample rate has been lowered to reduce the accumulative errors. However, vertical vibration magnifies high noise in the accelerometer. Figure 4.14(b) compares results from the IMU fusion and the SLAM algorithm. It also indicates that accelerometer noise has been averaged out. In fact, the estimated route in Figure 4.14 inferiors to the expected straight line on the xy plane. More drastic changes to the center of gravity create a problem with drift .

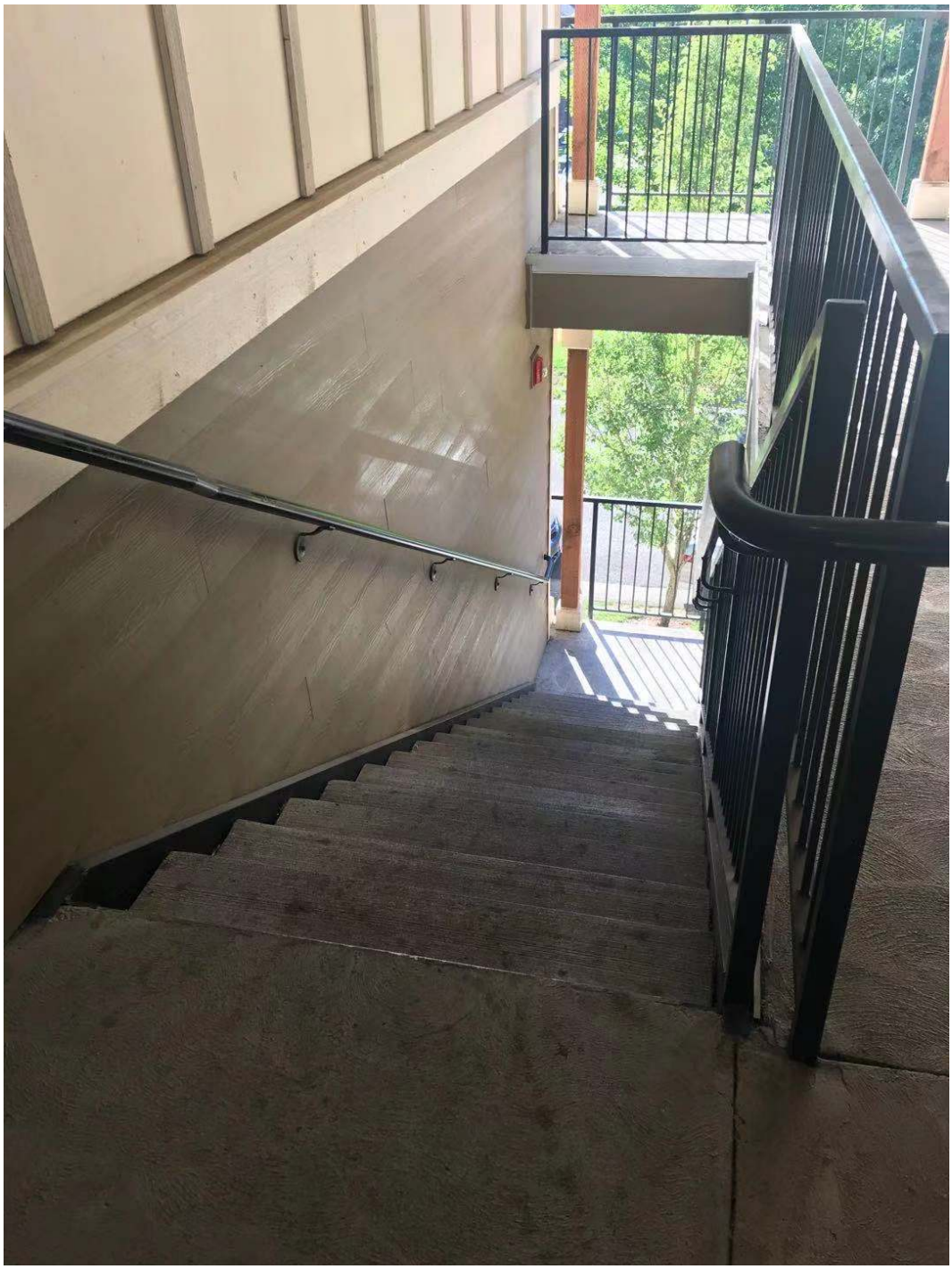
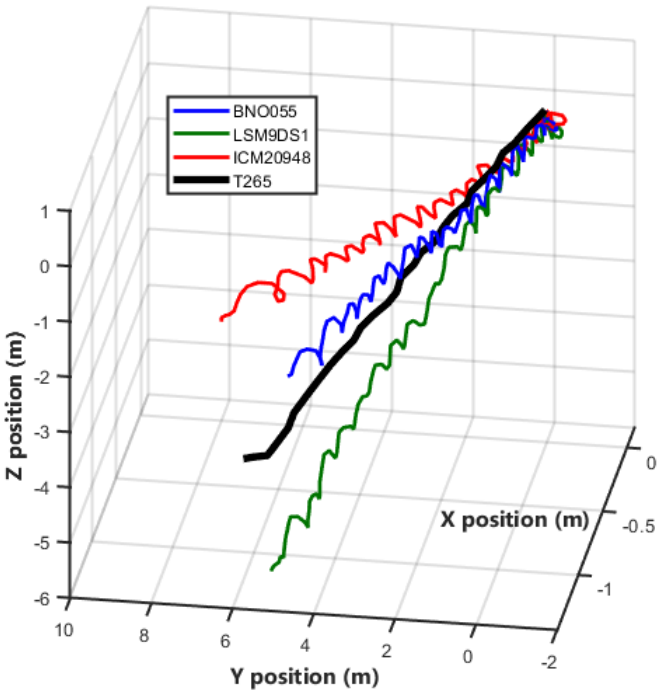
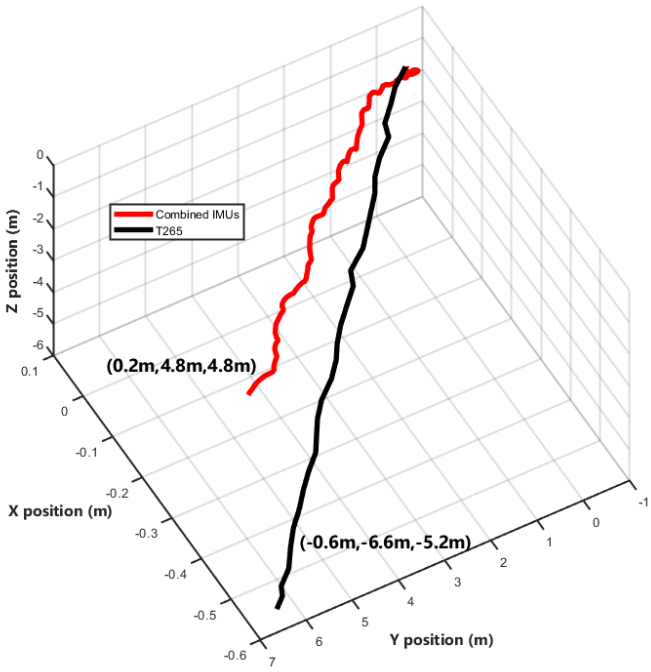


Figure 4.13: Experimental stairs



(a) Separate tracking



(b) Combined tracking

Figure 4.14: 3D Trajectory

Chapter 5: Discussion and Conclusion

5.1 Discussion

This report aims to explore and investigate the possibility of a highly efficient and accurate 9 DOF algorithm based on IMUs. In the experimental results, we preliminarily implement the planar indoor navigation for a short distance. However, our results also highlight some problems which commonly exist in inertial navigation.

One apparent issue is the device noise. In the 2D trajectory experiment, this noise affected the final results to some degree. Specifically, gyroscope drift and incompletely removed gravity jointly led to measuring errors [37]. Gyroscope drift is an almost unavoidable problem in practical measurement, and both bias stability and high frequency angle random walk (ARW) can cause the undesired drift. Bias stability can be defined as how stable the bias is over a certain period [47]. Lower bias stability or higher instability benefits users by reducing errors especially those integrated over time. Bias stability is determined by a gyroscope's comparative cost. With regard to high-frequency noise, relevant research suggests a gyroscope's sensitivity to acceleration is due to the asymmetry of its mechanical design; micromachining inaccuracies can also cause this problem [48]. Meanwhile, simple and compact mechanical system design fails to reject the vibration of sensors, exacerbating this issue.

Another problem involves the Kalman filter algorithm, which offers more room for improvement to achieve better results. In fact, a standard Kalman filter aims at reducing the errors in a linear system; thus, a dilution exists when the filter is applied to a non-linear system, such as practical inertial navigation. Focusing on these scenarios, the extended Kalman filter (EKF) helps to linearize a non-linear system by expanding the system's Taylor series. Like a harmonic, EKF extracts its first order to modeling the system, transferring that to a linear system. Besides, [37, 49] confirmed that a complementary filter can also be a potential solution to support IMUs. Unlike prediction and correction in the Kalman filter, the principle of complementary algorithm separately filters noise in accelerometers and gyroscopes by respectively using low pass and high pass filters. In fact, accelerometers are sensitive to instantaneous acceleration but provides reliable data for long-term measurement, indicating a low-frequency characteristic. Conversely, gyroscope readings are precise in short time intervals, but prolonged integral errors can greatly affect the measured data, indicating a high-frequency characteristic. Therefore, using a complementary filter can effectively remove the undesired signals as well as fully preserve the proposed information. In [50], researchers introduce the possibility of fusing both Kalman and complementary filter algorithms for indoor navigation, implementing accurate results [50]. The detailed steps are as follows: first, the complementary algorithm is used to filter frequency noise in both the accelerometer and the magnetometer; then, these processed datasets are jointly fused with the gyroscope in the Kalman filter to obtain the proposed orientation matrix. This method produces more accurate results, at the expense of

fast computation, but it also requires a strong support from both hardware and software [50].

Another difficulty in this project involves predicting a human's pose. When measured, a human's center of gravity, g_c, changes with walking, running and other daily activities. Even for a 2D simulation, an alteration of g_c (on Z direction) will disturb sensors' readings on x and y directions. To address this situation, a PDR algorithm or another step length estimation model could be introduced for training [51]. Compared a single Kalman filter, however, these form of algorithms need vast amounts of data to support precise prediction, which requires significant preliminary work. Similarly, hardware support for this algorithm fusion is also indispensable.

5.2 Conclusion

Undoubtedly, the storm called inertial indoor navigation has been sweeping the market of indoor positioning; based on its low cost, usability, and practicality. Differing from other visual devices, inertial navigation merely depends on data processing from the accelerometer, gyroscope, and magnetometer integrated into an IMU. However, certain hardware problems, such as gyro-drift and sensitivity to acceleration, reduce the reliability of measured data. Therefore, our research is of paramount importance, as it provides a potential solution to modify this inaccuracy and uncertainty. This research uses the Kalman filter algorithm to implement relatively accurate short-range human motion tracking. This report also

integrates, deduces and corrects many existing formulas involving linear acceleration, calibration, and so forth, providing reliable and detailed theoretical supports for future research. The final result shows that room for upgrades still exists in our algorithm, and deeper optimizations are needed to further improve its positioning accuracy. Separately from the other algorithm fusions, another ultimate goal is to introduce deep learning models for training, thereby reconciling a human's pose with modeling noise. With continuous research and algorithm updates, it can be predicted that such intelligent navigation algorithms will be empowered to fill the current void in the market.

Bibliography

[1] L.R.; Andrews A.P. Grewal, M.S.; Weill. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley Sons, Ltd, 2007.

[2] Chenshu Wu, Zheng Yang, Yunhao Liu, and Wei Xi. Will: Wireless indoor localization without site survey. *IEEE Transactions on Parallel and Distributed Systems*, 24(4):839–848, 2013.

[3] H. Al Nuaimi, K.; Kamel. A survey of indoor positioning systems and algorithms. *In Proceedings of the 2011 International Conference on Innovations in Information Technology.*, pages 185–190, 2011.

[4] Robert Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys Tutorials*, 15(3):1281–1293, 2013.

[5] Jussi Collin, Pavel Davidson, Martti Kirkko-Jaakkola, and Helena Leppäkoski. *Inertial Sensors and Their Applications*. Springer International Publishing, Cham, 2019.

[6] Faisal Jamil and Do Hyeun Kim. Improving accuracy of the alpha–beta filter algorithm using an ann-based learning mechanism in indoor navigation system. *Sensors*, 19(18), 2019.

[7] Hani Ramadhan, Yoga Yustiawan, and Joonho Kwon. A constrained k-nearest neighbor approach for semantic indoor trajectory extraction. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 13–16, 2020.

[8] Jiali Shen and Huosheng Hu. Visual based localization for mobile robots with support vector machines. In *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, pages 4176–4181, 2006.

[9] Carl Witt, Marc Bux, Wladislaw Gusew, and Ulf Leser. Predictive performance modeling for distributed batch processing using black box monitoring and machine learning. *Information Systems*, 82:33–52, 2019.

[10] Filippo Mortari, Eliseo Clementini, Sisi Zlatanova, and Liu Liu. An indoor navigation model and its network extraction. *Applied Geomatics*, 11(4):413–427, 6 2019.

[11] Yanying Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications Surveys & Tutorials*, 11:13–32, 2009.

[12] Christoph Fuchs, Nils Aschenbruck, Peter Martini, and Monika Wieneke. Indoor tracking for mission critical scenarios: A survey. *Pervasive and Mobile Computing*, 7(1):1–15, 2011.

[13] Bogdan Muset and S. Emerich. Distance measuring using accelerometer and gyroscope sensors. 2012.

[14] Zeyang Dai and Lei Jing. Lightweight extended kalman filter for marg sensors attitude estimation. *IEEE Sensors Journal*, 21(13):14749–14758, 2021.

[15] Sheng Shen, Mahanth Gowda, and Romit Roy Choudhury. Closing the gaps in inertial motion tracking. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, page 429–444, New York, NY, USA, 2018. Association for Computing Machinery.

[16] Lenka Tejmlova, Jiri Sebesta, and Petr Zelina. Artificial neural networks in an inertial measurement unit. In *2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)*, pages 176–180, 2016.

[17] Tsun-Hsu Lee, Shuyue Weng, and John Sanford. Indoor radio triangulation using only rssi data. In *2020 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*, pages 1097–1098, 2020.

[18] Dengpan Yuan, Maizhen Ning, Runguo Xu, Shuming Zhou, Wentao Shi, Kai Zheng, and Xin Huang. Model checking indoor positioning system with triangulation positioning technology. In *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, pages 862–866, 2018.

[19] Shixiong Xia, Y. Liu, Guan Yuan, Mingjun Zhu, and Zhaohui Wang. Indoor fingerprint positioning based on wi-fi: An overview. *ISPRS Int. J. Geo Inf.*, 6:135, 2017.

[20] Mayu Ohtani, Hisato Iwai, and Hideichi Sasaoka. Improvement of position estimation accuracy using multiple access points in terminal position estimation based on position fingerprint. In *2014 International Symposium on Antennas and Propagation Conference Proceedings*, pages 399–400, 2014.

[21] Min Li, Yutong Zhang, Yuxuan Zhang, and Jiwei Liu. Improvements and new algorithms for laterateion-based positioning and fingerprinting-based positioning. In *2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration*, pages 184–188, 2015.

[22] Ali Khalajmehrabadi, Nikolaos Gatsis, and David Akopian. Modern wlan fingerprinting indoor positioning methods and deployment challenges. *IEEE Communications Surveys Tutorials*, 19(3):1974–2002, 2017.

[23] Jose Maria Cabero, Fernando De la Torre, Aritz Sanchez, and Iñigo Arizaga. Indoor people tracking based on dynamic weighted multidimensional scaling. MSWiM '07, page 328–335, New York, NY, USA, 2007. Association for Computing Machinery.

[24] Haroon Rashid and Ashok Kumar Turuk. Dead reckoning localisation technique for mobile wireless sensor networks. *IET Wireless Sensor Systems*, 5(2):87–96, 2015.

[25] Alwin Poulose and D. Han. Indoor localization using pdr with wi-fi weighted path loss algorithm. *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 689–693, 2019.

[26] W. C. S.S. Simões and V. F. de Lucena. Hybrid indoor navigation as sistant for visually impaired people based on fusion of proximity method and pattern recognition algorithm. In *2016 IEEE 6th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, pages 108–111, 2016.

[27] Maria Fazio, Alina Buzachis, Antonino Galletta, Antonio Celesti, and Massimo Villari. A proximity-based indoor navigation system tackling the covid-19 social distancing measures. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2020.

[28] M. Fazio, M. Paone, A. Puliafito, and M. Villari. Huge amount of heterogeneous sensed data needs the cloud. In *International Multi-Conference on Systems, Signals Devices*, pages 1–6, 2012.

[29] Faisal Jamil, Naeem Iqbal, Shabir Ahmad, and Do-Hyeun Kim. Toward accurate position estimation using learning to prediction algorithm in indoor navigation. *Sensors*, 20(16), 2020.

[30] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. 2017.

[31] Sven Rönnbäck. Developement of a ins/gps navigation loop for an uav, 2000. Validerat; 20101217 (root).

[32] Xuemeng Chen, Xinxin Li, Zhaohui Song, Shusen Huang, Yuelin Wang, Jiwei Jiao, Heng Yang, and Ying Zhang. A micromachined gyroscope with piezoresistance for both high performance coriolis-effect detection and torsional vibration monitoring. In *The 13th International Conference on Solid-State Sensors, Actuators and Microsystems, 2005. Digest of Technical Papers. TRANSDUCERS '05.*, volume 1, pages 117–120 Vol. 1, 2005.

[33] Long Tran. Data fusion with 9 degrees of freedom inertial measurement unit to determine object's orientation. 06 2017.

[34] D. M. Henderson. Euler angles, quaternions, and transformation matrices for space shuttle analysis. 04 1977.

[35] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 03 1960.

[36] Ken Shoemake. Animating rotation with quaternion curves. 19(3):245–254, July 1985.

[37] Kaisheng Zhang. Exploring imu attitude and position estimation for improved location in indoor environments. 2019.

[38] Jiaxin Zhu, Weifeng Wang, Shiping Huang, and Wei Ding. An improved calibration technique for mems accelerometer-based inclinometers. *Sensors*, 20(2), 2020.

[39] C. Konvalin. Compensating for tilt, hard-iron, and soft-iron effects. 26, 12 2009.

[40] Changhui Hu, Rahul Khanna, Jay Nejedlo, Kangmin Hu, Huaping Liu, and Patrick Y. Chiang. A 90 nm-cmos, 500 mbps, 3–5 ghz fully-integrated ir-uwb transceiver with multipath equalization using pulse injection-locking

for receiver phase synchronization. *IEEE Journal of Solid-State Circuits*, 46(5):1076–1088, 2011.

[41] P. Zarchan and H. Musoff. *Fundamentals of Kalman Filtering: A Practical Approach*. Progress in astronautics and aeronautics. American Institute of Aeronautics and Astronautics, Incorporated, 2000.

[42] Faisal Jamil, Naeem Iqbal, Shabir Ahmad, and Do-Hyeun Kim. Toward accurate position estimation using learning to prediction algorithm in indoor navigation. *Sensors*, 20(16), 2020.

[43] Shao-Hung Chan, Ping-Tsang Wu, and Li-Chen Fu. Robust 2d indoor localization through laser slam and visual slam fusion. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1263–1268, 2018.

[44] Myriam SERVIERES, Valérie Renaudin, Alexis DUPUIS, and Nicolas ANTIGNY. Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking. *Journal of Sensors*, 2021:26 p., January 2021.

[45] Mathwork. Why SLAM Matters.

[46] Julius Schöning and Gunther Heidemann. Taxonomy of 3d sensors - a survey of state-of-the-art consumer 3d-reconstruction sensors and their field of applications. 02 2016.

[47] KVH industry. Guide to Comparing Gyro and IMU Technologies Micro-Electro-Mechanical Systems and Fiber Optic Gyros. *White paper*.

[48] Harvey Weinberg. Gyro Mechanical Performance: The Most Important Parameter.

[49] Yang Qian. Movement pattern detection through imu and barometer. 2020.

[50] Zhao Tan, Yiyong Wu, and Junbo Zhang. Fused attitude estimation algorithm based on explicit complementary filter and kalman filter for an indoor quadrotor uav. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 5813–5818, 2018.

[51] Ho Jin Ju, Min Su Lee, Chan Gook Park, Soyeon Lee, and Sangjoon Park. Advanced heuristic drift elimination for indoor pedestrian navigation. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 729–732, 2014.