sensors

MDPI

Article

Improving Accuracy of the Alpha–Beta Filter Algorithm Using an ANN-Based Learning Mechanism in Indoor Navigation System

Faisal Jamil and DoHyeun Kim *

Department of Computer Engineering, Jeju National University, Jejusi 63243, Korea; faisal@jejunu.ac.kr
* Correspondence: kimdh@jejunu.ac.kr

Received: 19 July 2019; Accepted: 6 September 2019; Published: 12 September 2019

Abstract: The navigation system has been around for the last several years. Recently, the emergence of miniaturized sensors has made it easy to navigate the object in an indoor environment. These sensors give away a great deal of information about the user (location, posture, communication patterns, etc.), which helps in capturing the user's context. Such information can be utilized to create smarter apps from which the user can benefit. A challenging new area that is receiving a lot of attention is Indoor Localization, whereas interest in location-based services is also rising. While numerous inertial measurement unit-based indoor localization techniques have been proposed, these techniques have many shortcomings related to accuracy and consistency. In this article, we present a novel solution for improving the accuracy of indoor navigation using a learning to perdition model. The design system tracks the location of the object in an indoor environment where the global positioning system and other satellites will not work properly. Moreover, in order to improve the accuracy of indoor navigation, we proposed a learning to prediction model-based artificial neural network to improve the prediction accuracy of the prediction algorithm. For experimental analysis, we use the next generation inertial measurement unit (IMU) in order to acquired sensing data. The next generation IMU is a compact IMU and data acquisition platform that combines onboard triple-axis sensors like accelerometers, gyroscopes, and magnetometers. Furthermore, we consider a scenario where the prediction algorithm is used to predict the actual sensor reading from the noisy sensor reading. Additionally, we have developed an artificial neural network-based learning module to tune the parameter of alpha and beta in the alpha–beta filter algorithm to minimize the amount of error in the current sensor readings. In order to evaluate the accuracy of the system, we carried out a number of experiments through which we observed that the alpha–beta filter with a learning module performed better than the traditional alpha–beta filter algorithm in terms of RMSE.

Keywords: indoor navigation system; inertial measurement unit; artificial neural network; alpha–beta filter; motion trackina

1. Introduction

The ability to navigate has always been of great importance when discovering new and unknown territories of the world. The evolution of various navigation techniques has helped us spread across the planet. Today, navigation remains an important part of our society. The technologies of today enable us to use the navigation in a whole new way than our ancestors could. Since smartphones were released on the market, a lot of location-based services have been developed. It is now possible to use navigation to find your way to a certain address or a point of interest, for example, the closest gasoline station or restaurant. All these functions are available because of Global positioning system (GPS) which has been integrated into those applications [1].

Nowadays, a GPS-based system is considered to be a well-known navigation system using satellites to calculate a receiver's current location on earth. These systems take the user's three-dimensional information (i.e., latitude, altitude, and longitude) [1–3]. The accuracy of GPS depends on the term of the line of sight; if the accuracy is good, then the system will easily locate the person or object within meters. Similarly, if the signal is weak, then the position of the object is unreliable, and the system is unable to get the exact position. Although GPS is considered to be very important for locating a target in an outdoor environment, it is not feasible for an indoor environment. In an indoor environment, there is always signal attenuation as compared to the outdoor environment because of weak signal or signal disturbed by impenetrable obstacles like a different object, concrete, and steel wall. These obstacles and hurdles continuously block signals coming from the satellites and, hence, it is difficult to locate an object's precise location [3,4]. Therefore, in consideration of these problems, the GPS is not reliable for an indoor positioning system (IPS) [5].

The IPS is a system used to track and locate the position of the object or a person inside a building by using sensor data, magnetic field, acoustic signal, radio waves, and WLAN nodes. During the past decade, many significant research has been done in the field of indoor localization [6]. This has lead to the development of several IPSs using different technologies for both research and commercial purposes. Many GPS chips have been manufactured in order to get the location of the object in an indoor environment, but the output is not accurate as compared to an outdoor environment. The tracking of a person within an indoor environment is the common use case that many researchers have adopted in order to evaluate the system [7]. During the past several years, many devices have been made in order to get the object location through sensor data. These devices are called inertial measurement units (IMUs) [8].

Many IMUs have been developed in the past several years which use sensors like accelerometers, gyroscopes, and magnetometers in order to calculate object localization [9]. These sensor data are used to calculate the linear acceleration and angular rate of a moving body, respectively. There are many ways to calculate the distance of the moving object [10]. One of the popular ways is to take double integration of acceleration concerning time to get the distance of the moving object [11,12]. In the case of constant acceleration, motion can be characterized by the motion equation. The combined acceleration (a), time (t), displacement (x), and velocity (v) are described as Motion. The rate of change of displacement is defined as velocity, and the rate of change of velocity is called acceleration. The velocity calculation is shown in Equation (1), in which the velocity is obtained by integrating the constant acceleration. In Equations (2)–(4) the velocity is further integrate to get position of the object.

$$v = \int a\mathrm{d}t = v_o + at \tag{1}$$

$$y = \int v\mathrm{d}t \tag{2}$$

$$y = \int (v_o + at)\mathrm{d}t \tag{3}$$

$$y = y_o + v_o t + \frac{1}{2}at^2 \tag{4}$$

Nevertheless, in these sensor values, there exists a dynamic noise of the accelerometer output, and it will increase with time by integrating the accelerometer [13]. Whenever a value is measured, there will be some error introduced by the transmission, as mentioned in Equation (5).

$$measured\_value = true\_value + noise \tag{5}$$

In order to get the precise output, different kinds of filters have been used, e.g., Wiener filter, low-pass filter, Kalman filter, Gaussian filter, Butterworth filter, alpha–beta filter, and high-pass filter, etc. These filters are responsible for removing noise from the measured value [14]. Enabling a prediction algorithm to cope with dynamic data or changing location data is a challenging task [15]. In this article, we propose a general architecture to improve the performance of the prediction algorithm using the learning module. The learning module monitors the performance of the prediction algorithm continuously by receiving the output as feedback. After analyzing the noise strength in the measured value and the output of the prediction algorithm, the learning module updates the tunable parameter or swaps the trained model of the prediction algorithm to improve its performance in terms of prediction accuracy. For experimental analysis, we have used the alpha–beta filter as a prediction algorithm, and our learning module is based on an artificial neural network.

The rest of the paper is organized as follows: A detailed overview of related work is presented in Section 2. In Section 3, we present the proposed learning to prediction model and inertial tracking in indoor navigation with conceptual design and detailed description of the chosen case study. A detailed discussion of the implementation and experimental setup is presented in Section 4. Section 5 presents the results of the proposed system. Finally, we conclude the paper in Section 6.

2. Related Work

Over the years, a lot of indoor positioning systems have been proposed to measure traveled distance. Navigation can be classified into two main categories, i.e., outdoor navigation and indoor navigation. The indoor navigation systems are further segregated into two main sub-categories, i.e., indoor positioning techniques and indoor positioning technologies. Furthermore, indoor techniques are further divided into two parts, i.e., signal properties and positioning algorithms. Location estimation and position algorithms are segregated into five categories, i.e., fingerprinting/Scene analysis, connectivity/neighborhood, triangulation, proximity, and trilateration [3]. Similarly, signal properties comprise seven types, i.e., Angle of Arrival (AoA), Time of arrival (ToA), Time of difference of arrival (TDoA), Received signal strength indication (RSSI), Hop-based, Interferometry, and Return time of flight (RToF). Finally, indoor positioning technologies are divided into ten categories, i.e., infrared, ultrasound, audible sound, magnetic, optical and vision, radio frequency, visible light, hybrid, inertial, and motion sensor. The overview of all these methods is presented in Figure 1. In this section, our main focus is to discuss the inertial and motion sensor in detail.

2.1. Inertial and Motion Sensor

In the inertial and motion sensor category, the distance of the object is calculated using the sensor's value, i.e., gyroscope, magnetometer, and accelerometer. The magnetometer is used to determine the orientation relative to the earth's magnetic field. The accelerometer is used to measure the acceleration of the object on a given axis. Similarly, the gyroscope is used to calculate the circular motion or angle of a moving object. From these sensors value, the double integration method over time yields the object's velocity in the first step, and the second step calculates velocity to get a distance as illustrated in Equation (4). The Inertial and motion sensors are also used within the dead reckoning navigation. In dead reckoning navigation, the position estimation is calculated based on continuous tracking of the object using acceleration from the origin [3].

In [16], the authors implemented two algorithms which aim to measure distance. The distance is measured using the double integration of accelerometer. However, in double integration, the error rate is more than expected. In the second algorithm, the distance traveled is measured by counting the number of steps. The distance traveled by the steps is measured by calculating the angle between legs using the accelerometer and gyroscope. In order to remove the noise, the complementary filter is used in the proposed algorithm. The main advantage of this system is to reduce the circuit cost and increase the efficiency of the system.

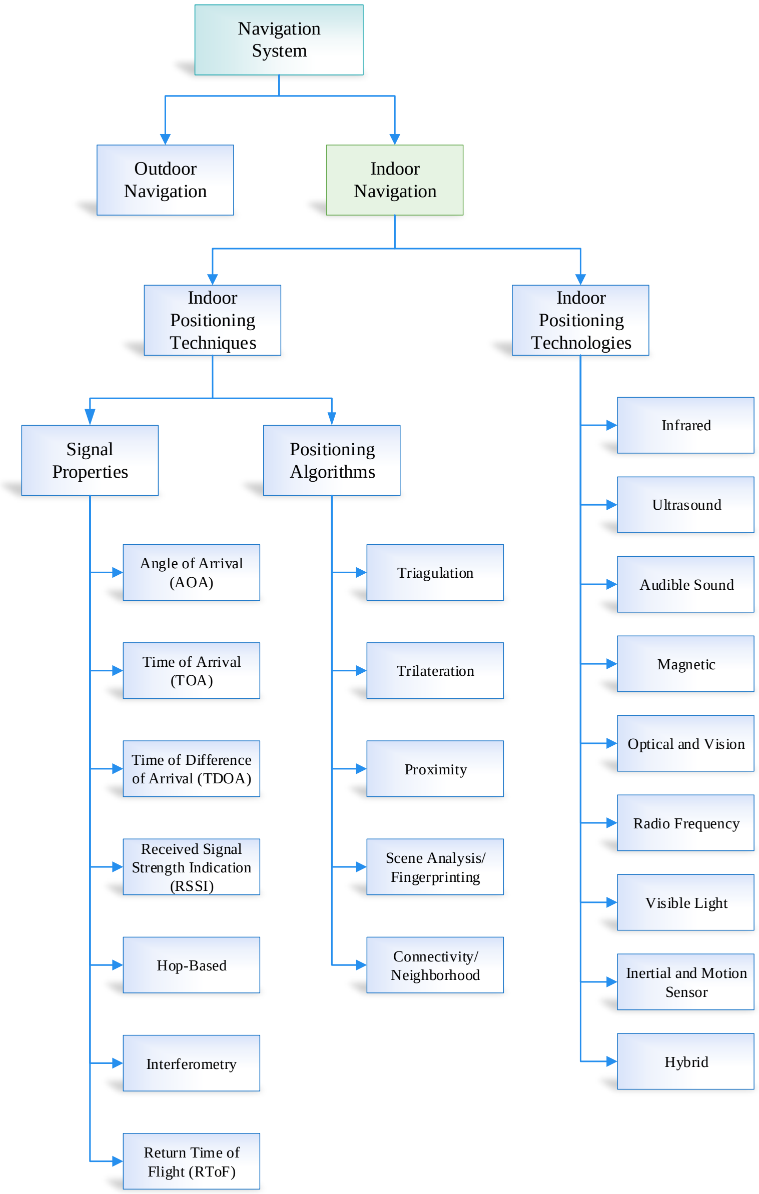

Figure 1. Taxonomy of indoor positioning algorithms.

A personal navigation system was presented in [17]. The developed system calculates the position of the pedestrian using the double integration method. The main aim of the system is to focus on three points; (i) real-time pedestrian position to get the accurate estimation, (ii) visualization of the position in 3D inside the building, and (iii) precise transition between the indoor and outdoor environments. The Kalman filter is also used to remove the sensing noise of the MTi/MT sensor in order to achieve accuracy. In [18], the authors proposed a new motion tracking system using two wearable inertial sensors. These inertial sensors are placed on upper limb joints near the wrist and elbow. An MT9B sensor is used which contains a 3-axis accelerometer, gyroscope, and magnetometer sensor in order to detect the motion of the human wrist, elbow and shoulder. In order to estimate the shoulder position, a Lagrangian-based optimization method was then adopted, integrating the translation and rotation components of the wearable inertial sensors.

The Kinematic-based model is designed to control the robotic arm using a dynamic state–space method in order to estimate the angle of the human shoulder using two wearable inertial sensors. In order to eliminate the noise, the Kalman filter has been used to implement the nonlinear state–space inertial tracker. The performance of the system is calculated in terms of RMS angle error, which is less than 8° for both shoulders and arms. Moreover, the average correlation is $r \geq 0.95$ for all movement tasks [19]. In [20], the authors presented an inertial tracking for mobile augmented reality. Real-time tracking is computed using an accelerometer, gyroscope, and silicon micro-machined. Six DoF are used to visualize the real-time movement and are capable of visualizing the movement in an indoor and outdoor environment.

Authors in [21] proposed a new method that detects the period of eating using a watch-like configuration. The sensor monitors the movement of wrist all day and detects whether the person is eating or not. The main aim of this study is to monitor the daily activity of a person in terms of energy intake.

2.1.1. Dead Reckonina

The inertial and motion sensors are used within the so-called dead reckoning navigation. Dead Reckoning (DR) is also known as pedestrian dead reckoning (PDR) and is a mechanism for estimating the user's current position using the previously known position with respect to time. DR is an alternative of radio navigation like the GPS, e.g., in case of bad weather due to signal attenuation, the GPS fails to work properly [1,22]. DR can give accurate position information, but it will give an error for a long periods of time [23]. In order to improve the accuracy of DR, the new hybrid solution is presented by the author which is more reliable than the existing solutions [24]. DR is also used with inertial navigation systems (INSs) such as a PDR in order to provide an accurate position estimation [25]. Similarly, DR is also embedded in micro-electromechanical systems (MEMS) to develop miniaturized electromechanical navigation device systems which are more reliable, accurate and have a low cost [26,27].

INS uses an inertial sensor to estimate the acceleration, position, velocity, and orientation of the object in motion with the involvement of external reference points [1,28]. This estimation of position, velocity, acceleration, and orientation is possible using DR integrated with inertial sensors, i.e., accelerometer, gyroscope, and magnetometer in order to attain an accurate estimation [29]. The common algorithm used in pedestrian navigation is Extended Kalman Filter (EKF), Particle Filter (PF), Kalman Filter (KF) integrated with INS to predict the position in indoor environment [23,25,30].

The authors in [31] presented the feasibility of using only the magnetic field for indoor positioning. The advantage of only using the magnetic field for position estimation in indoor environments is that no infrastructure is required to be deployed for the designed system which makes this approach cost-effective. Moreover, the performance of the system is directly proportional to the number of fingerprints. The magnetic field intensity data comprised of three groups, i.e., intensities in X, Y, Z direction. Furthermore, the magnetic field is unknown even with the integration of acceleration, i.e., horizontal intensity and vertical intensity.

In [32], the authors presented a VMag an infrastructure-free indoor positioning system fusion with magnetic and visual sensor. The proposed system is based on a novel approach for estimating the position in an indoor environment without relying on pre-deployed infrastructure assistance. The localization can be easily done by a user holding a smartphone. The presented system is designed using a particle filtering framework integrated with a neural network which improved the accuracy of the localization in an indoor environment. A number of experiments are carried out for different indoor settings, e.g., a laboratory, a garage, a canteen, and an office building.

Based on the comprehensive analysis of the state-of-the-art approaches in the field, limitations of available indoor techniques are described in Tables 1–3.

Table 1. Critical analysis of signal properties.

Signal	Property	Measurement	Metric
Angle of Arrival (AOA)	Angle-based	High accuracy at room level	Complex, expensive and low accuracy at wide coverage
Received Signal Strength Indication (RSSI)	Signal-based (RSS)	Medium accuracy	Low cost
Time of Arrival (TOA)	Distance-based	High accuracy	Complex and expensive
Time Difference of Arrival (TDOA)	Distance-based	High accuracy	Expensive
Hop-Based	Signal-based	High accuracy	Complex and expensive with short range coverage
Interferometry	Signal-based	Medium accuracy	Complex with low accuracy
Return Time of Flight (RToF)	Signal-based	Low accuracy	Short range coverage

Table 2. Critical analysis of positioning algorithms.

Positioning Algorithm	Signal Property	Pros	Cons
Triangulation	AOA	High accuracy at room level	Complex, expensive and low accuracy at wide coverage
Trilateration	TOA/TDOA	Medium accuracy	Complex and expensive
Proximity	RSSI	High accuracy	Complex and expensive
Connectivity/ Neighbourhood	RSSI/ Hop-based	High accuracy	Complex, expensive, short coverage
Scene analysis/fingerprinting	RSSI	High performance	Complex, expensive, medium accuracy and time consuming

Table 3. Critical analysis of positioning technologies.

Technology	Technique	Algorithm	Accuracy	Cost	Complexity	Scalability	Real-time
Infrared	Trilateration	TOA, TDOA	Medium	Low	High	Medium	Yes
Audible sound	Trilateration	TOA	Medium	Medium	Medium	Medium	Yes
Magnetic	Triangulation	AOA, TOA	High	High	High	Low	Yes
Bluetooth	Trilateration, fingerprinting	TDOA, RSSI	Low	Medium	Medium	Medium	Yes
WLAN	Trilateration, fingerprinting	TDOA, RSSI	Low	Medium	High	Medium	Yes
RFID	Fingerprinting	RSSI	Low	Medium	Medium	High	Yes
UWB	Trilateration	TOA, TDOA	High	Medium	Medium	Medium	Yes
NFC	Proximity	RSSI	High	Low	Low	High	No
WSN	Fingerprinting	RSSI	Medium	Medium	Medium	Medium	Yes
PDR/INS	DR	EKF, PF	Medium	Low	Low	Medium	Yes

Previously, there have been a lot of research proposed for increasing the performance and accuracy of the motion tracking and navigation systems using different algorithms, except for the alpha–beta filter algorithm. Nevertheless, none of these systems address the tuning of prediction algorithm with ANN. To the best knowledge of the authors, there has been no functional, positioning system for indoor navigation systems based on a learning to prediction model built so far.

3. System Architecture of Proposed Indoor Navigation

3.1. Scenario of Inertial Tracking in Indoor Navigation

The three-axis output from IMU in the form of linear acceleration and angular velocity is combined in the form of the non-linear matrix equation. The non-linear matrix has the information of both the orientation and position of the object in an indoor environment. The orientation of the object is calculated in two ways, i.e., orientation estimation from gyroscope output and orientation estimation from accelerometer output. In the case of the gyroscope, the absolute orientation cannot be calculated directly because of the drift associated with gyroscope readings. However, in the acceleration case, the orientation estimation is calculated using short-term and long-term stability. The short-term orientation stability is inaccurate as compared to long-term orientation stability because of the presence of ferromagnetic material. In other cases, the orientation matrix accuracy is disturbed due to rotational and linear acceleration. In order to fix the problem of orientation inaccuracy, we combined the advantages of long-term stability of accelerometer and short-term precision of gyroscope via an alpha–beta filter, as shown in Figure 2.

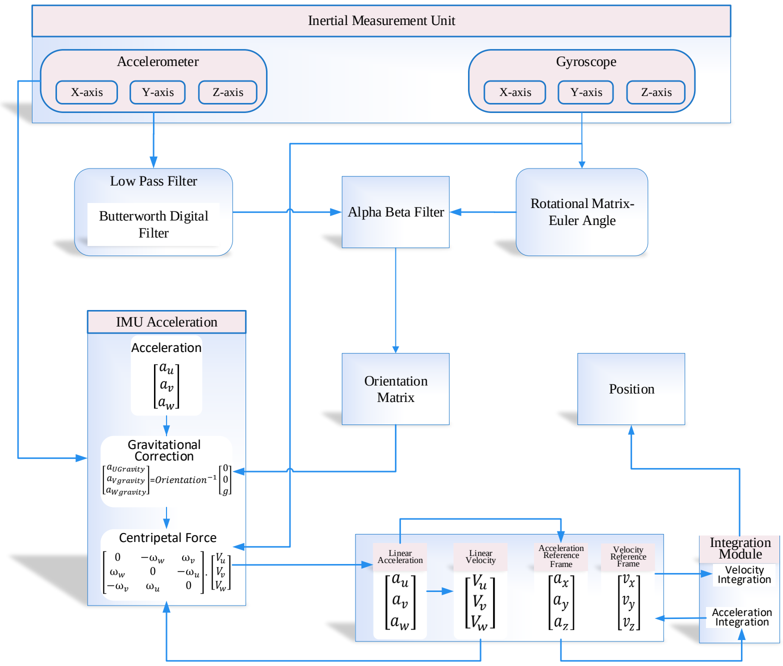
Inertial Measurement Unit
Accelerometer
X-axis
Y-axis
Z-axis
Gyroscope
X-axis
Y-axis
Z-axis
Low Pass Filter
Butterworth Digital Filter
Alpha Beta Filter
Rotational Matrix-Euler Angle
IMU Acceleration
Acceleration
$\begin{bmatrix} a_u \\ a_v \\ a_w \end{bmatrix}$
Gravitational Correction
$\begin{bmatrix} a_{UGravity} \\ a_{Vgravity} \\ a_{Wgravity} \end{bmatrix} = Orientation^{-1} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$
Centripetal Force
$\begin{bmatrix} 0 & -\omega_w & \omega_v \\ \omega_w & 0 & -\omega_u \\ -\omega_v & \omega_u & 0 \end{bmatrix} . \begin{bmatrix} V_u \\ V_v \\ V_w \end{bmatrix}$
Orientation Matrix
Position
Linear Acceleration
$\begin{bmatrix} a_u \\ a_v \\ a_w \end{bmatrix}$
Linear Velocity
$\begin{bmatrix} V_u \\ V_v \\ V_w \end{bmatrix}$
Acceleration Reference Frame
$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$
Velocity Reference Frame
$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$
Integration Module
Velocity Integration
Acceleration Integration

Figure 2. Inertial tracking scenario for proposed indoor navigation system.

3.1.1. Orientation Estimation from Gyroscope in Indoor Navigation

The orientation estimation from gyroscope output in indoor navigation is measured using the Euler angles (ψ, ϕ, and θ). The Euler angle is used to describe the orientation of an object in 3-dimensional Euclidean space. Furthermore, we also keep track of the order of the rotation in every time step, which is as follow.

$$\Delta\theta(P+1) = \Delta t \frac{\theta(P+1) + \theta(P)}{2} \tag{6}$$

$$\Delta\phi(P+1) = \Delta t \frac{\phi(P+1) + \phi(P)}{2} \tag{7}$$

$$\Delta\psi(P+1) = \Delta t \frac{\alpha(P+1) + \alpha(P)}{2} \tag{8}$$

We use the trapezium rule as numerical integration method in above three equation and these Equations (6), (7), and (8) represents the Yaw, Pitch and Roll. Where ΔP is the time index, $\Delta\ \theta$, $\Delta\ \psi$, and $\Delta\ \phi$ denotes the incremental angle around the W-axis, U-axis, and V-axis.

The rotation matrix mention in Figure 2 is denoted as R in (9), (10), and (11) around each particular axis.

$$R(W, \theta, P+1) = \begin{pmatrix} cos\Delta\theta(P+1) & -sin\Delta\theta(P+1) & 0 \\ sin\Delta\theta(P+1) & cos\Delta\theta(P+1) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{9}$$

$$R(V,\phi,P+1)=\begin{pmatrix} cos\Delta\phi(P+1) & 0 & sin\Delta\phi(P+1) \\ 0 & 1 & 0 \\ -sin\Delta\phi(P+1) & 0 & cos\Delta\phi(P+1) \end{pmatrix} \tag{10}$$

$$R(U,\psi,P+1)=\begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\Delta\psi(P+1) & -sin\Delta\psi(P+1) \\ 0 & sin\Delta\psi(P+1) & cos\Delta\psi(P+1) \end{pmatrix} \tag{11}$$

In (9), (10), and (11) are combined in form of general rotation matrix represent in (11).

$$Rotation(P+1)=R(W,\theta,P+1).R(V,\phi,P+1).R(U,\psi,P+1) \tag{12}$$

Finally, the orientation matrix is defined in (13).

$$Orientation(K+1)=Rotation(K+1).Orientation(K) \tag{13}$$

The orientation matrix contains information related to the orientation of IMU. Therefore, in order to get the orientation matrix, we process the coordinates of any vector from the IMU-Fixed Frame to the Earth-Fixed Frame.

3.1.2. Orientation Estimation from Accelerometer in Indoor Navigation

In the proposed Indoor navigation system, the orientation matrix is a 3 × 3 matrix that transforms vectors linked with IMU-Fixed Frame into an Earth-Fixed Frame. The impact of the transformation of the vector in the proposed method is to transform orientation matrix coordinates from IMU-Fixed Frame into Earth-Fixed Frame, and similarly, the inverse transforms from Earth-Fixed Frame to IMU-Fixed Frame. The inverse of the orientation matrix is defined in (14).

$$orientation^{-1}=\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{14}$$

We also calculate the acceleration magnitude, which is denoted by a, since the vector is referenced to Earth-Fixed Frame, which is parallel to the z-axis. Therefore, the accelartion vector coordinates in the Earth-Fixed Frame are (0 0 g) and, (a_{ua} a_{va} a_{wa}) is the acceleration field vector directly measured from the IMU output. In this case, the accelartion vector coordinates concerning Earth-Fixed Frame is represented in (15).

$$\begin{pmatrix} a_{ua} \\ a_{va} \\ a_{wa} \end{pmatrix}=\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}\cdot\begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \tag{15}$$

where the acceleration vector to Earth Fixed-Frame is also represented in (15), where g denotes the magnitude of acceleration. The inverse of the orientation matrix is mentioned in (15) and (16):

$$\begin{pmatrix} a_{13} \\ a_{23} \\ a_{33} \end{pmatrix}=\frac{1}{a}\cdot\begin{pmatrix} a_{ua} \\ a_{va} \\ a_{wa} \end{pmatrix} \tag{16}$$

Lastly, across a product from the third and first column, we compute the second column inverse of the orientation matrix. The proposed system is comprised of three main components, i.e., orientation matrix, linear acceleration, and position tracking. Therefore, the equation for inertial tracking is represented in (17).

$$X(P+1) = \alpha \cdot [Rotation(P+1) \cdot X(P)] + \beta \cdot [Orientation(P+1) \cdot \frac{(P+1)+(P)}{(P)}] \tag{17}$$

Equation (17) used trapezium rule as the numerical integration method where.

$$X = \begin{pmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \tag{18}$$

Equation (18) represents the position and velocity of the object in a proposed indoor environment where x, y, z are the 3-axis position and $\dot{x}$, $\dot{y}$, $\dot{z}$ are the velocity of objects.

$$\alpha = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} I_{3x3} & \Delta t.I_{3x3} \\ o_{3x3} & I_{3x3} \end{pmatrix} \tag{19}$$

$$A_{acc} = \begin{pmatrix} a_u \\ a_v \\ a_w \end{pmatrix} \tag{20}$$

$$L_{vec} = \begin{pmatrix} V_U \\ V_V \\ V_W \end{pmatrix} \tag{21}$$

Equation (20), A_{acc} represent the acceleration of the object in proposed indoor navigation and similarly (21) represents the linear velocity vector denoted as L_{vec} .

$$\beta = \begin{pmatrix} \frac{\Delta t^2}{2} \cdot I_{3x3} \\ \Delta t \cdot I_{3x3} \end{pmatrix} \tag{22}$$

$$Rotation(P+1) = \begin{pmatrix} I_{3x3} & O_{3x3} \\ O_{3x3} & Rotation(P+1) \end{pmatrix} \tag{23}$$

In Equation (23) the identity is represented by 1, and Null matrices are denoted by 0.

In the proposed inertial tracking scenario, we integrate linear acceleration to get the linear velocity in order to get the position in the proposed indoor system. However, to get an accurate position, we have to remove both the centripetal force from the data used in the linear acceleration. In Figure 2, IMU acceleration module has three-component, i.e., linear acceleration and centripetal acceleration. Therefore the IMU acceleration in Fixed-Frame is calculated using (24).

The rotation of the object can be measured using two ways, e.g., (i) self-rotation of the body and (ii) body rotation around a point in space. However, in the proposed indoor navigation system, we only consider body rotation around a point in space; therefore, we calculate the centripetal force using the linear velocity and angular velocity cross product. The result of this cross product is the linear velocity concerning IMU Fixed-Frame with the axis(U, V, W) which is represented in (26).

$$\begin{pmatrix} a_{Ua} \\ a_{Va} \\ a_{Wa} \end{pmatrix} (t + \Delta t) = Rotation^{-1}(t + \Delta t) \begin{pmatrix} a_{Ua} \\ a_{Va} \\ a_{Wa} \end{pmatrix} (t) \tag{24}$$

Similarly, using (24) we represent the orientation matrix in (25) as follow:

$$\begin{pmatrix} a_{Ua} \\ a_{Va} \\ a_{Wa} \end{pmatrix} (t) = Orientation^{-1}(t) \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \tag{25}$$

Finally, gravitational acceleration with respect to IMU Fixed-Frame is represented in (26).

$$\begin{pmatrix} a_{Ua} \\ a_{Va} \\ a_{Wa} \end{pmatrix} \tag{26}$$

$$\begin{bmatrix} a_{ucentripetal} \\ a_{vcentripetal} \\ a_{wcentripetal} \end{bmatrix} = \begin{bmatrix} 0 & -\omega_w & \omega_v \\ \omega_w & 0 & -\omega_u \\ -\omega_v & \omega_u & 0 \end{bmatrix} \cdot \begin{bmatrix} V_U \\ V_V \\ V_W \end{bmatrix} \tag{27}$$

3.2. Proposed System Architecture of Learning to Prediction Scheme

The proposed learning to prediction scheme is comprised of the prediction algorithm and learning module. Traditionally, we trained prediction algorithms using historical data, so that they can learn the hidden pattern and relationship among input and output parameters. Subsequently, trained models are used to predict the output for any given input data. The prediction algorithm will perform well when input data and the application scenario remain the same as the training data conditions. However, the existing prediction algorithm does not allow adaptation of the trained model with changing and dynamic input conditions. To overcome this limitation, we propose the learning to prediction model, as shown in Figure 3. In order to improve the accuracy of the prediction algorithm, we tune the prediction algorithm using a learning module. In the proposed system, the performance of the prediction algorithm is contentiously monitored by the learning algorithm by receiving its output as feedback. Moreover, the performance of the prediction algorithm also depends on external parameters, which are considered by the learning module. After investigating the output and the current external factors of the prediction algorithm, the learning module may update the tunable parameters of the prediction algorithm or completely replace the trained model in a prediction algorithm to improve its performance in terms of prediction accuracy when environmental triggers are observed.

The development environment of the proposed system is categorized into two parts, i.e., the learning module and prediction algorithm. The prediction algorithm is based on the alpha–beta filter, and for the learning module, we used the artificial neural network (ANN). An alpha–beta filter is a simplified form of the observer for data smoothing, control, and estimation application. An alpha–beta filter is a lightweight algorithm that does not require historical data, but only previous state information to make an intelligent prediction about the actual state of the system. In the proposed method, the alpha–beta filter is used to predict the actual IMU sensor value, i.e., accelerometer and gyroscope from the noisy IMU sensor value. Noise in accelerometer sensor readings is due to the gyro bias. However, in the case of the learning module, we choose the ANN algorithm which takes three input, i.e., acceleration, gyroscope and previously predicted value (feedback) as shown in Figure 4.

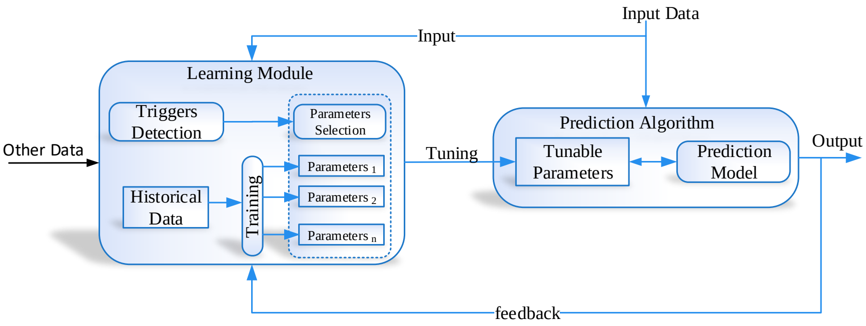

Figure 3. Conceptual view of learning to prediction model.

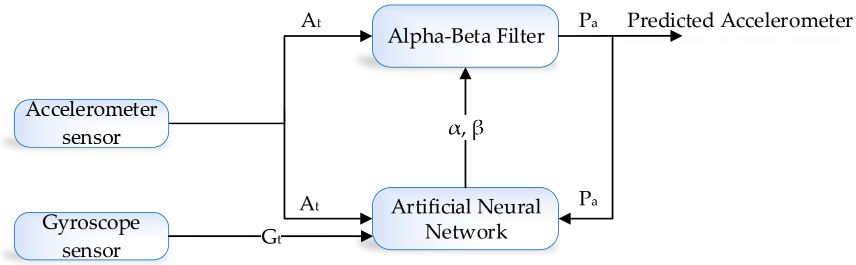

Figure 4. Accelerometer and gyroscope prediction using ANN-based learning module.

Similarly, in the case of gyroscope prediction, the noise in gyroscope is due to the influence of accelerometer value. For the learning module, we take three input, i.e., gyroscope, acceleration, and previously predicted value(feedback) as shown in Figure 5.

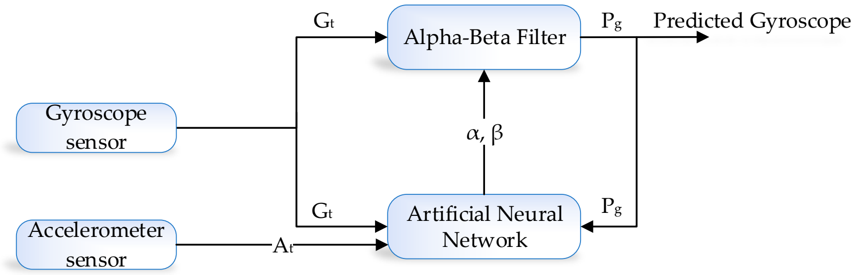

Figure 5. Gyroscope prediction using ANN-based learning module.

The alpha–beta filter algorithm gets readings from the IMU sensor at time t i.e., A_t and G_t, and will predict accelerometer and gyroscope P_a and P_a data by removing noise. The performance of the alpha–beta filter algorithm's is mainly controlled through a tunable parameter known as α and β that is calculated at each iteration. The residual (r) in the sensors reading is computed by the learning module so that α and β can be updated intelligently. In the next subsection, we will explain the detail of the alpha–beta filter before going into detailed architecture.

3.3. Alpha-Beta Filter Algorithm

An alpha–beta filter is a simplified form of an observer for data smoothing, estimation, and control application. It consists of two internal states in which the first state is obtained by integrating the value of the second state over time. The output value of the system corresponds to the observation of the first state and disturbance. The α and β gain is the most important parameter of the alpha–beta filter's design, which is the main key point behind the performance of the algorithm. An alpha–beta algorithm updates the value of gain depending on the situation to control weights given to the system's own predicted state or sensor readings. The detailed working with all the components of the alpha–beta filter are present in Figure 6.

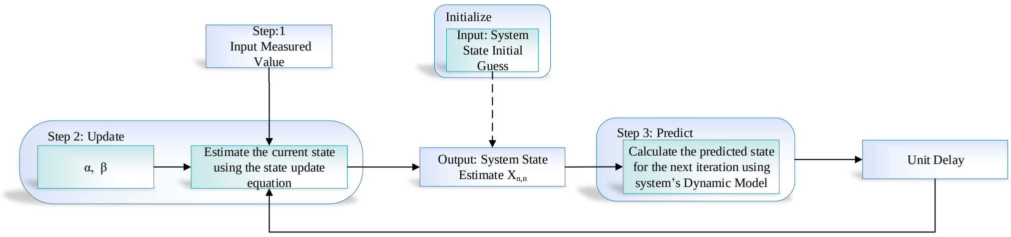
Step:1
Input Measured Value
Initialize
Input: System State Initial Guess
Step 2: Update
α, β
Estimate the current state using the state update equation
Output: System State Estimate X$_{n,n}$
Step 3: Predict
Calculate the predicted state for the next iteration using system's Dynamic Model
Unit Delay

Figure 6. Working of the alpha–beta filter algorithm.

The noise in the sensor value depends on environmental factors, which can seriously affect sensor readings in that environment. In this study, we consider an IMU sensor reading (i.e., accelerometer and gyroscope) having noise, and let us assume P_a and P_a and an accelerometer and gyroscope at time t. The alpha–beta filter algorithm includes the process model that can make an internal prediction about the system state, i.e., estimated accelerometer and estimated gyroscope, and then, it is compared with the current sensor reading to decide predicted accelerometer and predicted gyroscope T_{t+1} at time $t+1$. The α β filter is commonly used as an efficient tracking filter. It may be viewed as the steady-state of the second-order Kalman filter. Next, we briefly explain the step-by-step working of alpha–beta filter algorithm, that is, how it removes the noise from the IMU sensor data.

$$\hat{X}_{K|K-1} = F\hat{X}_{K-1|K-1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \hat{X}_K - 1|K-1 \tag{28}$$

In (28) $\hat{X}_{K|K}$ and $\hat{X}_{K|K-1}$ represents the estimated and predicted vector state at time KT.

$$\hat{X}_{K|K} = F\hat{X}_{K|K-1} + K_{V_K} = \hat{X}_{K|K-1} + \begin{bmatrix} \alpha \\ \beta \\ \bar{T} \end{bmatrix} V_K \tag{29}$$

where $\bar{T}$ represent the sampling period, in (29), the steady-state Kalman filter gain is represented by K, where V_K is the innovation process which is further defined in (30).

$$V_K = Z_K - \hat{Z}_K \tag{30}$$

In Equation (30), Z_K is the Kth measurement vector and Z_K represents its predicted value, as mentioned by (31).

$$\hat{Z}_K = H\hat{X}_{K|K-1} \tag{31}$$

where H represent the measurement matrix, as mentioned in (32)

$$H = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{32}$$

Finally the α, β coefficients is calculated as follow:

$$\beta = \frac{\alpha^2}{2-\alpha} \tag{33}$$

However, the λ is represented as tracking index. The λ function is the variance and the measurement noise, receptively.

$$\lambda = \frac{\rho_v}{\rho_w} T^2 \tag{34}$$

The ρ_v in (34) is represented as standard deviations of the system noise and ρ_w is the measurement noise respectively. The ideal value of α and β is calculated using the (35)

$$\alpha = -\frac{1}{8}(\lambda^2 + 8\lambda - (\lambda + 4)\sqrt{\lambda^2 + 8\lambda}) \tag{35}$$

$$\beta = 2(2 - \alpha) - 4\sqrt{1 - \alpha} \tag{36}$$

3.4. ANN-Based Learning to Prediction for the Alpha–Beta Filter

ANN algorithms are general-purpose learning algorithms and are actively used in solving a wide range of problems, including regression, classification, clustering, pattern recognition, forecasting, and time-series data processing. In the proposed indoor navigation system, the alpha–beta filter algorithm [33] is controlled using the ANN-based learning module. Figure 6 illustrates the flow diagram of the operation of the alpha–beta filter, which works fine with the optimal value of α and β. In the proposed system, we predict the actual value of accelerometer and gyroscope from the noisy sensor reading. The conventional alpha–beta filter fails to predict the actual sensor value under dynamic conditions. In Figure 7, we proposed the detailed learning to prediction model, which is based on an artificial neural network algorithm taking three inputs, i.e., current sensor value (i.e., accelerometer and gyroscope) and previously predicted sensor values. The output of the ANN algorithm is the optimal value of α and β. The updated value of α and β is passed to the filter algorithm to tune its prediction accuracy by adjusting the α β value. The number of neurons defineed in the hidden layer are ten; we defined ten hidden layers because it was the best compromise between accuracy and efficiency; the number of neurons in the output layer is two. The tang-sigmoid function has been used as an activation function. The ANN provides α, and β values as output to the filter algorithm used in (29).

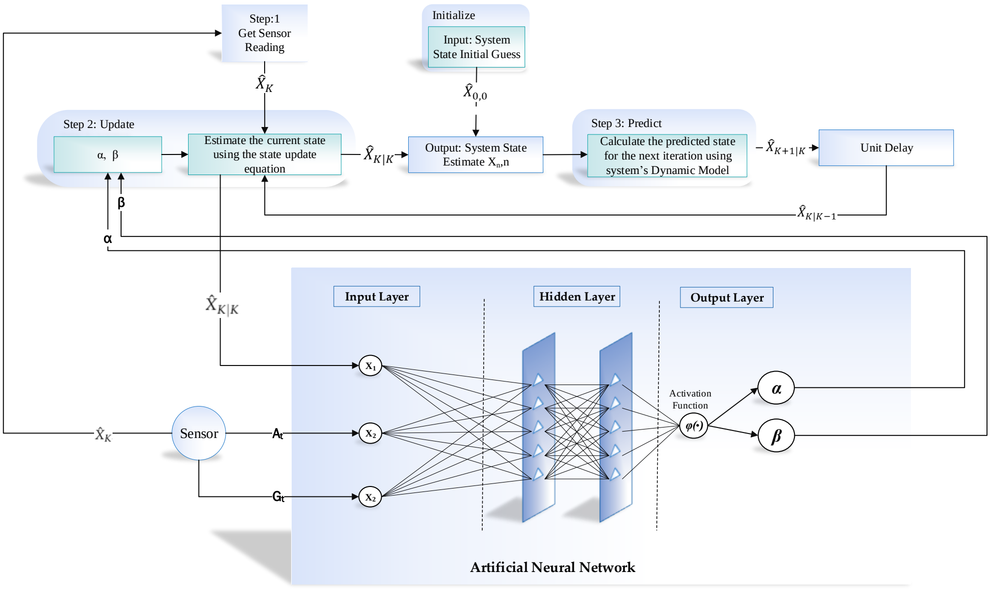

Figure 7. Detailed diagram of accelerometer and gyroscope prediction using the alpha–beta filter with learning module.

The proposed learning to prediction model enables the alpha–beta filter to estimate the actual accelerometer and gyroscope accurately from the noisy sensor reading with a dynamic error rate.

4. Implementation for ANN-Based Learning Mechanism in Indoor Navigation

4.1. Development Environment

The development environment of the proposed system is categorized into two parts, as illustrated in Figures 2 and 3, i.e., inertial tracking in indoor navigation and the learning to prediction model. For inertial tracking in the indoor navigation system, we have used NGIMU [34], which is a data acquisition platform that combines on-board sensors and data processing algorithms. The NGIMU contains a triple-axis accelerometer and gyroscope sensor [34]. The detailed characteristics of NGIMU are described in Table 4.

Table 4. Characteristic of NGIMU.

Sensor	Description	
Gyroscope	Range	± 2000°/s
	Resolution	0.06°/s
	Sample Rate	400 Hz
Accelerometer	Range	±16 g
	Resolution	490 µg
	Sample Rate	400 Hz
Magnetometer	Range	±1300 µT
	Resolution	∼0.3 µT
	Sample Rate	∼20 Hz

The NGIMU sensor in the proposed system is used to collect data for inertial tracking of an object in an indoor environment. Therefore, the data were taken from two different locations while making the object navigate in each respective location. First, the data were taken from the movement of an object from the main corridor to room No. D243 in the Engineering building-4. Afterwards, the same pattern was followed while the object was moving in the conference room in Ocen Science building of Jeju National University, South Korea. The time duration for collecting one sample is approximately 60 s in which the first 10 s remained stationary so that the algorithm could converge on a stable state. The detailed development environment of inertial tracking in the indoor navigation system is summarized in Table 5.

Table 5. Development environment for the proposed inertial tracking in indoor navigation.

Component	Description
IDE	MATLAB R2018a
Operating System	Window 10
CPU	Intel(R) Core(TM) i5-8500 CPU@3.00GHz
Memory	8GB
Signal Processing Filter	Butterworth Digital Filter
Data Smoothing Algorithm	Alpha-Beta filter

Similarly, in the case of the learning to prediction model, we have used ANN for learning module and alpha–beta filter as a prediction algorithm. The Tool and technologies for implementing learning to prediction model mentioned in Table 6. All the implementation and experimental work of this study were carried out on Window 10 with an Intel Core i5-8500@3.00GHz processor and 8 GB memory. Moreover, we use MATLAB R2018a as a development environment for both inertial tracking and learning to the prediction model. In order to make the system friendly and efficient, we use the third party tool, i.e., NGIMU application programming interface (API) to extract data from the NGIMU sensor.

Table 6. Development environment for the proposed Learning to Prediction Model.

Component	Description
IDE	MATLAB R2018a
Operating System	Window 10
CPU	Intel(R) Core(TM) i5-8500 CPU@3.00GHz
Memory	8GB
Artificial Neural Network	Feed Forward Backpropagation
Neuron in Hidden Layer	10
Neuron in output Layer	2
Number of Input	3
Prediction Algorithm	Alpha-Beta filter

4.2. Implementation

We implemented the proposed system for the evaluation of the alpha–beta filter algorithm with the learning module in MATLAB R2018a. The implementation of the proposed system is comprised of two modules, i.e., Learning to prediction module and inertial tracking for indoor navigation. In inertial tracking for indoor navigation, the experiments were performed on a real dataset containing an accelerometer and gyroscope data. The data were taken from two different locations while making the object navigate in each respective location. First, the data were taken from the movement of object from the main corridor to room No. D243 in engineering building-4. Afterwards, the same pattern was followed while the object was moving in the conference room in Ocen Science building-5 of Jeju National University. The time duration of collecting a sample is 60 s. We loaded the data into the proposed application data structure from the Excel file. The data contain three inputs (i.e., accelerometer, gyroscope, and previously predicted value of sensor). First, we retrieved sensor data along with current time. After that, we calculated the magnitude of the acceleration and applied a low pass Butterworth digital filter to remove noise from the sensor data. Similarly, in the case of a gyroscope, we also computed magnitude first and then the rotational Matrix–Euler angle. The absolute value was then passed to the alpha–beta filter for further signal smoothing. After the orientation matrix was calculated, it was used to perform a rotation. Finally, when all bias from the data was removed, we used the traditional position estimation method, which is a double integration method. In a double integration method, first, we calculate the linear velocity through the integration of linear acceleration, then we calculate the position by taking the integration of linear velocity. The detailed flow chart of inertial tracking for indoor navigation is described in Figure 8.

In Learning to prediction module, first, we computed the root mean squared error (RMSE) for the sensor readings by comparing its values with the original sensor data, i.e., accelerometer and gyroscope data. The RMSE for the sensor reading is very high, i.e., 5.32.

Next, we used the alpha–beta filter algorithm to predict actual accelerometer and gyroscope readings from the noisy sensor reading. The implementation interface provides manual tuning of the alpha–beta filter internal parameter, i.e., time, current system state, a current derivate of the system state. Experiments were conducted with different values of input parameters, and the corresponding results were collected. The RMSE for the predicted accelerometer and gyroscope value using Alpha-Beta filter with $\alpha = 0.5$ and $\beta = 0.1$ was 2.59, which was much better than RMSE of the sensor reading, i.e., 53.32% reduction of the error. However, it still needs improvement. We have to use MATLAB for the implementation of the ANN-based learning module to predict and tune the α and β to improve the prediction accuracy of the alpha–beta filter algorithm. The ANN algorithm has three neurons in the input layer for current sensor value (i.e., accelerometer and gyroscope) and predicted sensor values and two neurons in the output layer for prediction the error in sensor readings. Input and output data were normalized using the Equation mentioned in (37).

$$\tilde{n}_i = \frac{n_i - n_{min}}{n_{max} - n_{min}} \tag{37}$$

The $\tilde{n}_i$ is the normalized value for the ith data point of the input and the output parameters, i.e., current sensor reading (i.e., accelerometer and gyroscope) and predicted sensor values, and the predicted α and β value. The minimum and maximum values in the available dataset is represented by n_{min} and n_{max} respectively. Usually the ANN network is trained with the normalized data, therefore, we need to de-normalize the output of the neural network to get the corresponding predicted error using (38).

$$error_i = er\tilde{r}or_i x(error_{max} - error_{min}) + error_{min} \quad (38)$$

In the proposed system, we considered different configurations (e.g., the activation functions, changing neurons number in the hidden layer, learning rates) for the training of the ANN algorithm. We carried out multiple experiments for the training of every individual configuration of ANN; average results are reported to factor out the stochastic element in ANN network weights' initialization. Moreover, we have used the four-fold cross-validation technique to evade the bias in the training process. The four-fold cross-validation is used for every individual configuration in the proposed system. In this case, we divided our dataset into four subsets of equal size (i.e., 2490 instances in each subset). The training and testing dataset used for each model in our four-fold cross-validation process is illustrated in Figure 9.

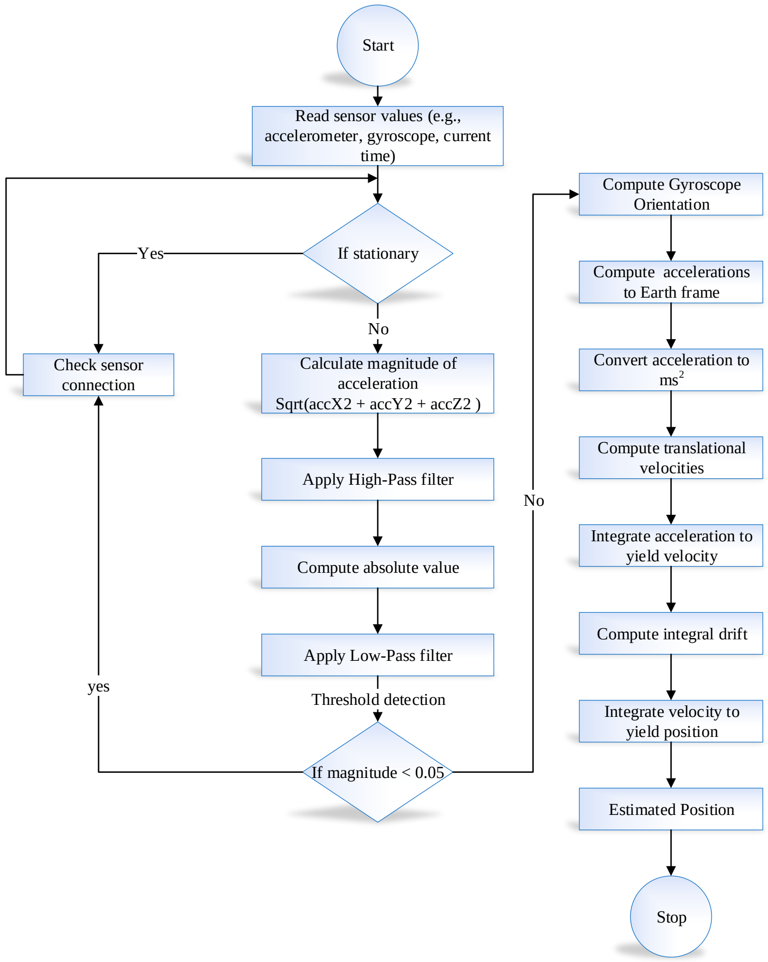
Start
Read sensor values (e.g., accelerometer, gyroscope, current time)
If stationary
Yes
No
Check sensor connection
Calculate magnitude of acceleration Sqrt(accX2 + accY2 + accZ2)
Apply High-Pass filter
Compute absolute value
Apply Low-Pass filter
Threshold detection
If magnitude < 0.05
yes
No
Compute Gyroscope Orientation
Compute accelerations to Earth frame
Convert acceleration to ms2
Compute translational velocities
Integrate acceleration to yield velocity
Compute integral drift
Integrate velocity to yield position
Estimated Position
Stop

Figure 8. Flow chart of inertial tracking in indoor navigation.

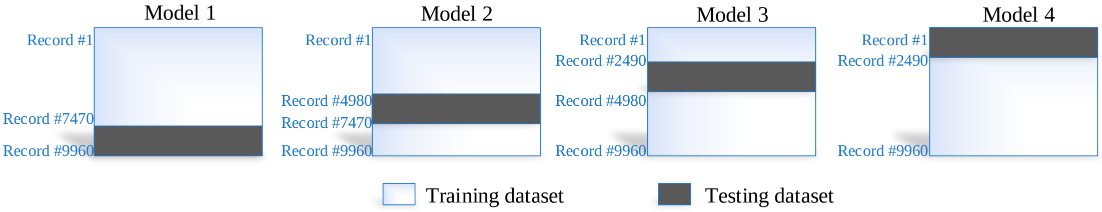

Figure 9. Training and testing dataset in four-fold cross-validation.

According to this technique, 25% of the dataset is used for testing, and the remaining 75% is used for training the ANN algorithm. The ANN training algorithm was based on the Levenberg–Marquardt algorithm, which is considered to be the best and fastest method for moderately sized neural networks. The maximum number of epochs used to train the ANN was 100. The selected configuration for ANN along with the corresponding prediction accuracy in terms of RMSE for training and the testing dataset are summarized in Table 7.

Table 7. Prediction accuracy in terms of RMSE for training and testing dataset using four-fold cross validation in indoor navigation system.

Experiment ID	ANN Configuration			Model 1		Model 2		Model 3		Model 4		Model Average (Test Cases)	Experiments Average (Test Case)
	Activation Function	Hidden Layers	Learning Rate	Trainina	Test	Trainina	Test	Trainina	Test	Trainina	Test		
1	Sigmoid	10	0.1	0.25	0.22	0.12	0.12	0.25	0.25	0.21	0.24	0.20	0.23
2	Sigmoid	10	0.1	0.26	0.23	0.25	0.26	0.28	0.28	0.50	0.35	0.28	
3	Sigmoid	10	0.1	0.29	0.24	0.15	0.17	0.10	0.12	0.32	0.35	0.22	
1	Linear	10	0.1	4.56	5.22	5.06	3.20	4.48	5.07	4.58	4.90	4.59	4.59
2	Linear	10	0.1	4.56	5.22	5.06	3.20	4.48	5.07	4.58	4.90	4.59	
3	Linear	10	0.1	4.56	5.22	5.06	3.20	4.48	5.07	4.58	4.90	4.59	
1	**Sigmoid**	**10**	**0.2**	**0.19**	**0.15**	**0.08**	**0.09**	**0.20**	**0.22**	**0.24**	**0.27**	**0.18**	**0.19**
2	**Sigmoid**	**10**	**0.2**	**0.24**	**0.25**	**0.18**	**0.20**	**0.19**	**0.20**	**0.18**	**0.19**	**0.21**	
3	**Sigmoid**	**10**	**0.2**	**0.24**	**0.19**	**0.15**	**0.16**	**0.24**	**0.24**	**0.22**	**0.23**	**0.20**	
1	Linear	10	0.2	4.44	5.18	5.03	3.17	4.49	5.04	4.55	4.87	4.56	4.56
2	Linear	10	0.2	4.44	5.18	5.03	3.17	4.49	5.04	4.55	4.87	4.56	
3	Linear	10	0.2	4.44	5.18	5.03	3.17	4.49	5.04	4.55	4.87	4.56	
1	Sigmoid	15	0.1	1.15	0.91	0.27	0.34	0.34	0.33	0.24	0.27	0.46	0.33
2	Sigmoid	15	0.1	0.13	0.11	0.23	0.25	0.23	0.20	0.31	0.31	0.21	
3	Sigmoid	15	0.1	0.57	0.45	0.34	0.36	0.22	0.22	0.19	0.23	0.31	
1	Linear	15	0.1	4.45	5.19	5.04	3.18	4.50	5.05	4.56	4.88	4.57	4.57
2	Linear	15	0.1	4.45	5.19	5.04	3.18	4.50	5.05	4.56	4.88	4.57	
3	Linear	15	0.1	4.45	5.19	5.04	3.18	4.50	5.05	4.56	4.88	4.57	
1	Sigmoid	15	0.2	0.27	0.23	0.56	0.91	0.19	0.22	0.40	0.40	0.44	0.30
2	Sigmoid	15	0.2	0.24	0.20	0.22	0.25	0.26	0.29	0.20	0.23	0.24	
3	Sigmoid	15	0.2	0.25	0.19	0.20	0.24	0.21	0.22	0.21	0.24	0.22	
1	Linear	15	0.2	4.45	5.19	5.04	3.18	4.50	5.05	4.56	4.88	4.57	4.57
2	Linear	15	0.2	4.45	5.19	5.04	3.18	4.50	5.05	4.56	4.88	4.57	
3	Linear	15	0.2	4.45	5.19	5.04	3.18	4.50	5.05	4.56	4.88	4.57	

Table 7 shows that the ANN prediction accuracy in terms of RMSE is affected by changing the learning rate or changing neuron number in the hidden layer. Nevertheless, a significant difference in prediction accuracy can be seen for each model in the four-fold cross-validation process. In the ANN algorithm, the sigmoid activation function is commonly used for prediction accuracy. In the proposed indoor navigation system, the sigmoid activation function performs better, and significant improvement can be observed as compared to the linear activation function. The highlighted column shows the best case achieved for the ANN algorithm using a sigmoid activation function with $learningrate = 0.2$ and $neuron = 10$. The best case configuration is further used for tuning the performance of the alpha–beta algorithm.

5. Results and Discussions

We have used an open-source NGIMU API for collecting real-time accelerometer and gyroscope data to calculate the inertial tracking in an indoor environment. Furthermore, in order to analyze the performance of the proposed system, we compared the proposed learning to prediction model with conventional alpha–beta filter to observe the improvement in the prediction accuracy of the alpha–beta filter algorithm results. For the traditional filter, the result was collected with varying the value of α and β. Hence, the proposed system is comprised of two modules (i.e., the inertial tracking in indoor navigation module and learning to prediction module); therefore, in this section, we first demonstrate and discuss the inertial tracking in indoor navigation module results and then learning to prediction.

The inertial tracking required sensor data (i.e., accelerometer and gyroscope) which is taken using NGIMU in Jeju National University, South Korea. Figure 10 investigated the accelerometer data which is collected form NGIMU sensor. The 3-axis accelerometer data with respect to time are shown along with the filtered and stationary data. The stationary data depict the accelerometer magnitude less than 0.05 to check the state object. Similarly, we use the Butterworth filter to filter the accelerometer data using the specified cut off frequency.

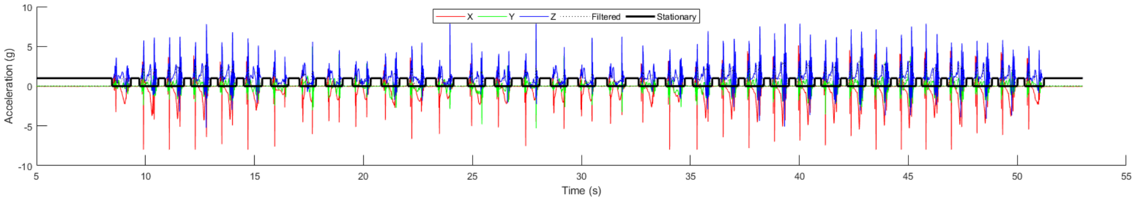

Figure 10. Acceleration.

Figure 11 shows the angular velocity of the object in an indoor environment calculated using gyroscope data. The angular velocity is used as the object moves through an angle. It is the change in perspective of a moving object divided by time. The angular velocity is calculated using the following formula.

$$\omega = \frac{(\theta_f - \theta_i)}{t} \tag{39}$$

In Equation (39), ω represent the angular velocity, θ_f is final angle of the object, θ_i is the initial angle of the object, t represent the time, and $\Delta\theta$ is the change of angle.

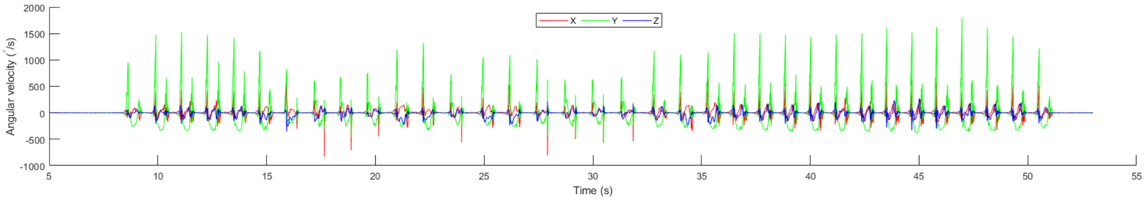

Figure 11. Angular velocity.

The 3-axis acceleration of the object in an indoor environment is illustrated in Figure 12. Acceleration is the rate of change of velocity divided by time. Therefore, in indoor navigation, the acceleration is calculated using the following formula.

$$a = \frac{v_f - v_i}{t} \tag{40}$$

$$a = \frac{\Delta v}{t} \tag{41}$$

In Equations (40) and (41), the a represent the acceleration in m/s_2, v_f is the final velocity, v_i is the initial velocity of the object, t represent the time in second, and Δv is the change in velocity in m/s.

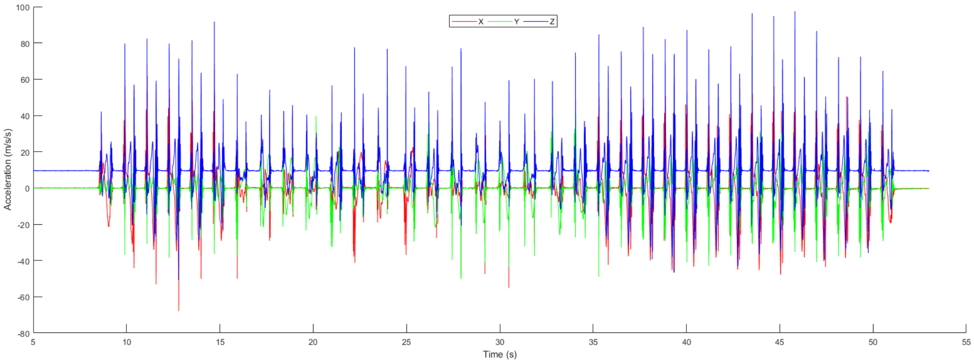

Figure 12. Acceleration m/s^2.

In Figure 13, we calculate the 3-dimensional velocity of the object in an indoor environment. The velocity is used to measure how fast the object is moving; therefore in the proposed system, we calculate the velocity using the following formula.

$$v = \frac{x_f - x_i}{t} \tag{42}$$

$$v = \frac{\Delta x}{t} \tag{43}$$

In Equations (42) and (43), the v represent the velocity in m/s, x_f is final position of the object, x_i is the initial position, t is a time(s) in which change occur, and Δx represent the change in position.

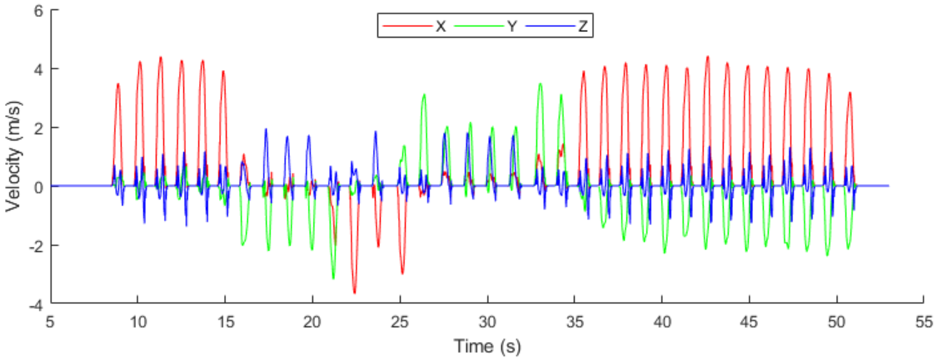

Figure 13. Velocity m/s.

Figure 14 shows the position data plot of the 60 s of straight walk-in MCL Lab as a map in Figures 15 and 16. We have concluded from this graph that drifts has significantly been reduced by the proposed system. However, there exists an error because the displacement on the axis should meet at the origin, but they didn't. The x, y, and z represent the three-axis position.

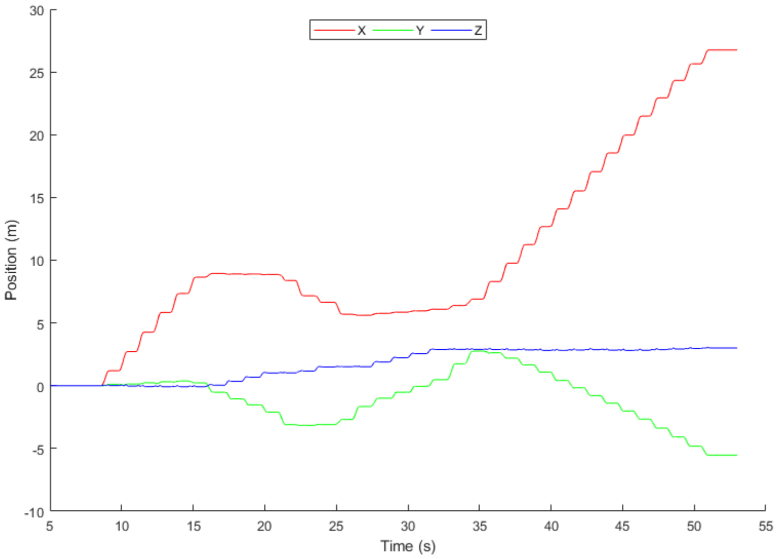

Figure 14. Position.

Figures 15 and 16 show the trajectory constrained in the X–Y plane from the top view. The Person starts walking from the start point and stops at the endpoint. As we saw in the graph, there is a drift in the start as it presents the person in stationary mode. In Figure 15, our lab has dimension of 20 × 29, where 12 is the length and 29 is the width. We have set our reference point on origin which is (0,0) therefore, our starting point is (9,2) with respect to reference point of the lab, which is (0,0). All the coordinates of the tracking line are referenced with respect to our reference point (0,0).

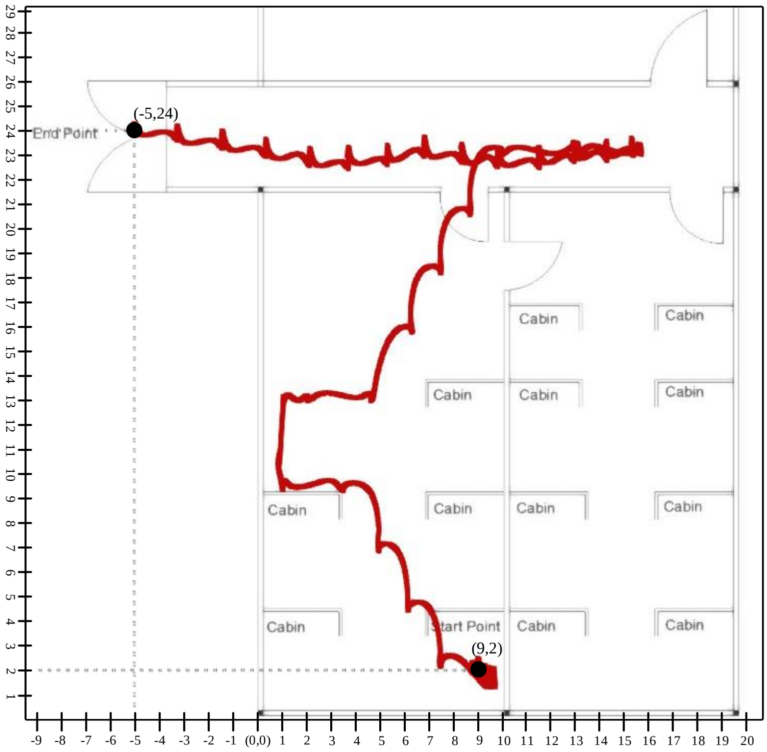

Figure 15. Person tracking scenario in Engineering building-4 of Jeju National University.

Similarly, in Figure 16, the conference has a dimension of 20 × 30, where 20 is the length and 30 is the width. We have set the reference point with respect to the origin, where starting point is (11,23) and the endpoint is (20,5).

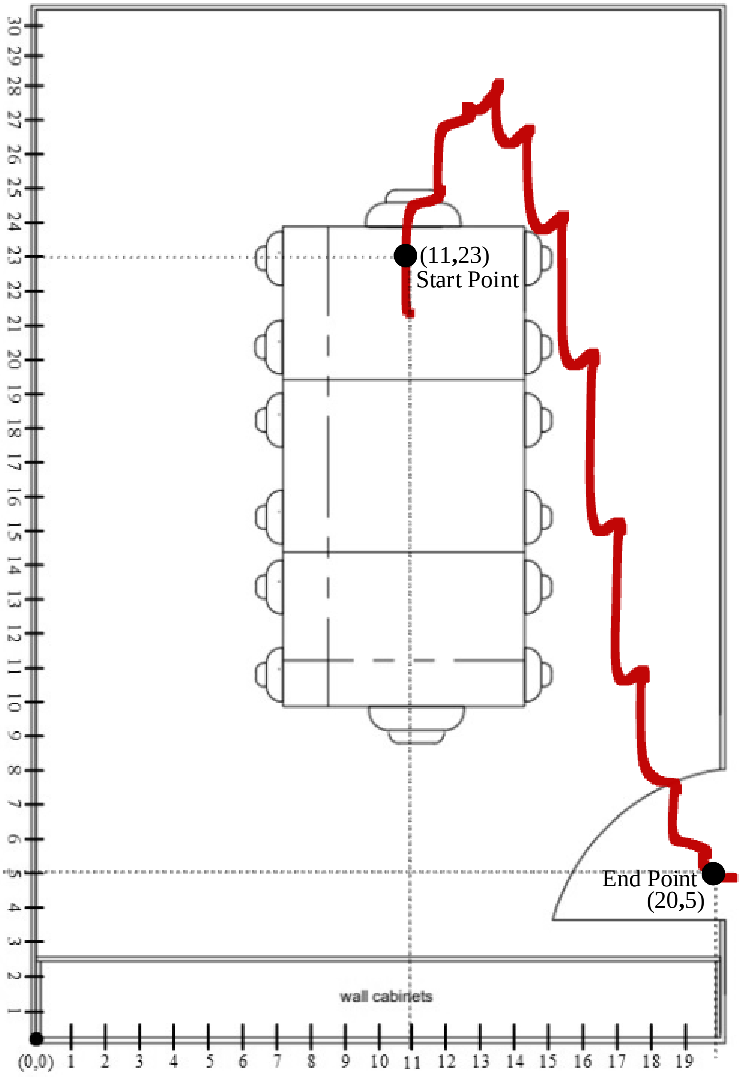

Figure 16. Person tracking scenario in ocean sciences building-5 of Jeju National University.

Figures 17 and 18 show the results of the alpha–beta filter with selected values of alpha and beta. The optimal values of alpha and beta are not fixed, and depend on the available dataset. It is challenging to choose the optimal values of alpha and beta in the alpha–beta filter manually. Therefore, multiple experiments were conducted with different values of alpha and beta; however, using $alpha = 0.75$ and $beta = 0.05$, we predict the required value of accelerometer and gyroscope for the noisy sensor reading.

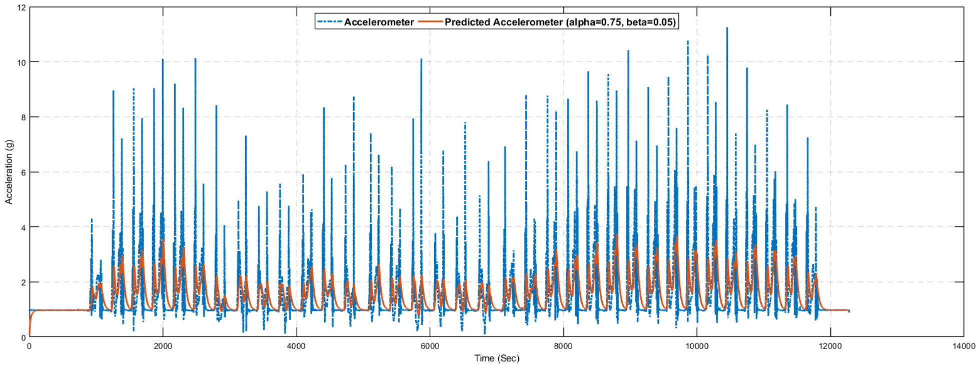

Figure 17. Accelerometer prediction results using the alpha–beta filter algorithm with selected values of alpha and beta.

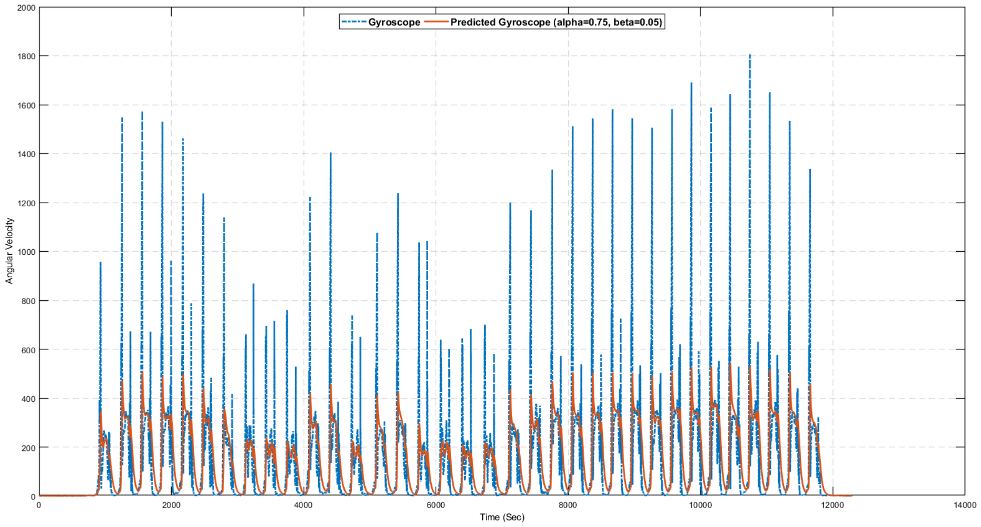

Figure 18. Gyroscope prediction results using the alpha–beta filter algorithm with selected values of alpha and beta.

Next, we present the results of the alpha–beta filter tuned with the proposed learning to prediction model. After tuning the ANN learning module, we used the trained model to improve the performance of the alpha–beta filter algorithm by appropriately tuning its parameters alpha and beta. In order to find the alpha-beta gain from the predicted error, we need to choose an appropriate value (i.e., R, the proportionality constant) called the error factor as represented in Equation (44).

$$gain = \frac{error_i}{R} \tag{44}$$

Hence, experiments were conducted by varying the values of the error factor R. Figures 19 and 20 show the prediction results of the alpha–beta filter algorithm using the learning module, varying the values of the error factor R. It is very difficult to comprehend the results presented in Figures 19 and 20, as the difference between the results is not so obvious visually. Therefore, we used various statistical measures to summarize these results in the form of a single statical value for the quantifiable comparative analysis.

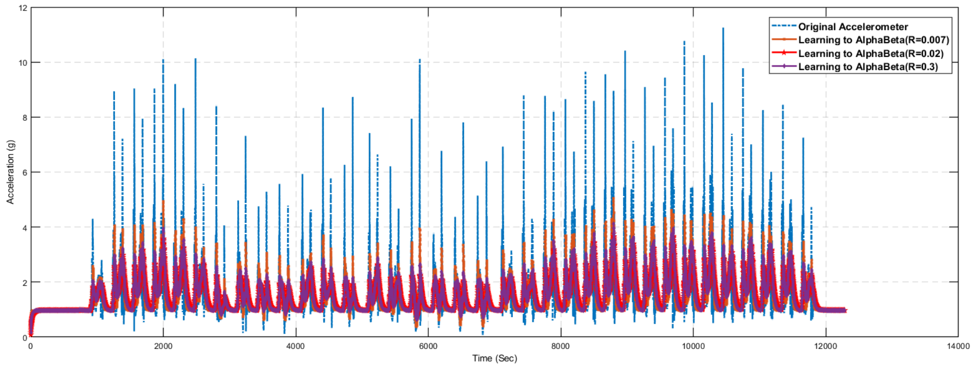

Figure 19. Accelerometer prediction results using the proposed learning to Alpha-Beta filter algorithm with selected error factor *R*.

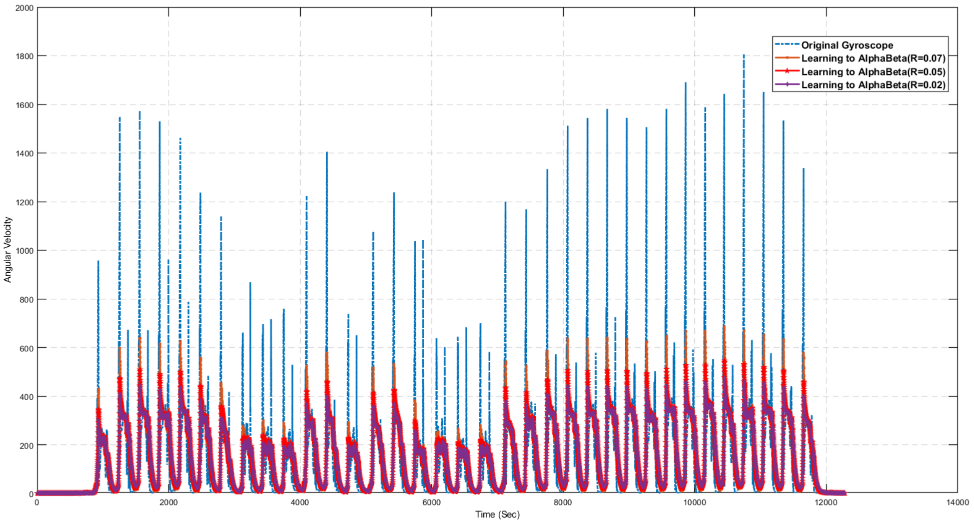

Figure 20. Gyroscope prediction results using the proposed learning to Alpha-Beta filter algorithm with selected error factor *R*.

In Table 8, we computed the RMSE of the position with and without proposed learning to the prediction model. According to the results, we improved the accuracy of position estimation by 18%. The proposed learning to prediction model corrects the bias error by removing the noise and improving the accuracy of the system.

Table 8. Estimating error in position with and with proposed learning to prediction model.

Experiment ID	Position Error	Position Error with Proposed Learning to Prediction Model
1	0.130 mm	0.102 mm
2	0.115 mm	0.098 mm
3	0.135 mm	0.112 mm

We have used three statistical measures that were used for performance comparison in terms of accuracy, i.e., Mean absolute deviation (MAD), Mean Squared Error (MSE), Root mean squared error (RMSE), and Mean absolute error (MAE). The formulas of these statistical measures are presented in Equations (45)–(47).

$$MAD = \frac{\sum_{n}^{i=1} |Actual_i - \widehat{Predicted}_i|}{n} \tag{45}$$

MAD is used to compute an average deviation found in the predicted values from the actual values. The calculation is done by dividing the sum of the absolute difference between the actual accelerometer and actual gyroscope $Actual_i$ and predicted accelerometer and gyroscope $Pred\hat{i}cted_i$ by the alpha–beta filter with the total number of the data items, i.e., n.

$$MSE = \frac{\sum_{n}^{i=1}(Actual_i - Pred\hat{i}cted_i)^2}{n} \tag{46}$$

Similarly, MSE is considered the most widely used statistical measure in the performance evaluation of the prediction algorithms. Squaring the error magnitude not only removes the negative and positive error problems, but it also gives more penalty for higher misdirections as compared to low errors.

$$RMSE = \sqrt{\frac{\sum_{n}^{i=1}(Actual_i - Pred\hat{i}cted_i)^2}{n}} \tag{47}$$

Finally, the mean absolute error (MAE) is the absolute error to measure the difference between two continuous variables.

$$\frac{1}{n}\sum_{n}^{i=1}|(Actual_i - Pred\hat{i}cted_i)| \tag{48}$$

The problem with MSE is that it magnifies the actual error, which sometimes makes it difficult to realize and comprehend the actual error amount. This problem is resolved by the RMSE measure, which is obtained by simply taking the square root of MSE.

Table 9 presents the statistical summary of the results for the alpha–beta filter algorithm with and without a learning module. Comparative analysis shows that the alpha–beta filter with the proposed learning to prediction model results in an error factor $R = 0.02$ (highlighted in bold), outperforming all other settings on all statistical measures. The best results for the alpha–beta filter without the learning module were obtained with $alpha = 0.75$ and $beta = 0.05$, which results in prediction accuracy of 2.49 in terms of RMSE. Similarly, the best results for the alpha–beta filter with the learning module were obtained with R = 0.02, which results in prediction accuracy of 2.38 in terms of RMSE. Figures 17–20 show sample results for best cases of an alpha–beta filter with and without the ANN-based learning module. The relative improvement in prediction accuracy of the proposed learning to prediction model (best case), when compared to the best and worst-case result of the alpha–beta filter without the learning module, was 4.41% and 11.19% in terms of RMSE metric, respectively. Significant improvement in prediction accuracy gives us the confidence to further explore the application of the proposed learning to prediction model to improve the performance of other prediction algorithms.

Table 9. Statistical summary of the alpha–beta filter prediction results with and without the ANN-based learning module.

Metric	Alpha-Beta Filter		Alpha-Beta with Learning Module	
	$\alpha = 0.75, \beta = 0.05$	$\alpha = 1.45, \beta = 0.75$	R = 0.02	R = 0.1
RMSE	**2.494**	2.527	**2.388**	2.481
MAD	**0.163**	0.166	**0.156**	0.165
MSE	**6.222**	6.388	**5.701**	6.155
MAE	**0.997**	1.137	**0.931**	1.2156

6. Conclusions

In this article, we presented a novel learning to prediction model to improve the performance of prediction algorithms under dynamic conditions. The proposed model enabled conventional prediction algorithms to adapt to dynamic conditions through continuous monitoring of its performance and tuning of its internal parameters. To evaluate the effectiveness of the proposed learning to the prediction model, we developed an ANN-based learning module to improve the prediction accuracy of the alpha–beta filter algorithm as a case study. The proposed learning to prediction scheme improved the performance of the alpha–beta filter prediction by dynamically tuning its internal parameter α and β, i.e., estimated error in measurement. The ANN-based learning to prediction takes three input parameters (i.e., the current sensor reading (i.e., accelerometer and gyroscope) and alpha–beta predicted reading) in order to predict the estimated noise in sensor readings. Afterwards, the estimated error in the measurement parameter, i.e., α, β in the alpha–beta filter is updated by dividing the estimated error with a noise factor R.

Author Contributions: Data curation, F.J.; Formal analysis, F.J.; Funding acquisition, D.K.; Investigation, F.J.; Methodology, D.K.; Software, F.J.; Supervision, D.K.; Validation, D.K; Visualization, F.J.

Funding: This research received no external funding.

Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (No. NRF–2018R1A5A1025137), and this research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP–2019–2014–1–00743) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation). Any correspondence related to this paper should be addressed to Dohyeun Kim.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Grewal, M.S.; Weill, L.R.; Andrews, A.P. *Global Positioning Systems, Inertial Navigation, and Integration*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
2. Koyuncu, H.; Yang, S.H. A survey of indoor positioning and object locating systems. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **2010**, *10*, 121–128.
3. Fuchs, C.; Aschenbruck, N.; Martini, P.; Wieneke, M. Indoor tracking for mission critical scenarios: A survey. *Pervasive Mob. Comput.* **2011**, *7*, 1–15. [CrossRef]
4. Rantakokko, J.; Händel, P.; Fredholm, M.; Marsten-Eklöf, F. User requirements for localization and tracking technology: A survey of mission-specific needs and constraints. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2010), Zurich, Switzerland, 15–17 September 2010; pp. 1–9.
5. Khudhair, A.A.; Jabbar, S.Q.; Sulttan, M.Q.; Wang, D. Wireless indoor localization systems and techniques: survey and comparative study. *Indones. J. Electr. Eng. Comput. Sci.* **2016**, *3*, 392–409. [CrossRef]
6. Brena, R.F.; García-Vázquez, J.P.; Galván-Tejada, C.E.; Muñoz-Rodriguez, D.; Vargas-Rosales, C.; Fangmeyer, J. Evolution of indoor positioning technologies: A survey. *J. Sens.* **2017**, *2017*, 2630413. [CrossRef]
7. Gu, Y.; Lo, A.; Niemegeers, I. A survey of indoor positioning systems for wireless personal networks. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 13–32. [CrossRef]
8. Kok, M.; Hol, J.D.; Schön, T.B. Using inertial sensors for position and orientation estimation. *arXiv* **2017**, arXiv:1704.06053.
9. Faramondi, L.; Inderst, F.; Pascucci, F.; Setola, R.; Delprato, U. An enhanced indoor positioning system for first responders. In Proceedings of the 4th International Conference on Indoor Positioning and Indoor Navigation (IPIN 2013), Montbeliard-Belfort, France, 28–31 October 2013; pp. 1–8.
10. Filippeschi, A.; Schmitz, N.; Miezal, M.; Bleser, G.; Ruffaldi, E.; Stricker, D. Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion. *Sensors* **2017**, *17*, 1257. [CrossRef]
11. Arraigada, M.; Partl, M. Calculation of displacements of measured accelerations, analysis of two accelerometers and application in road engineering. In Proceedings of the 6th Swiss Transport Research Conference (STRC 2006), Monte Verità, Ascona, Switzerland, 15–17 March 2006; p. 30.
12. Seifert, K.; Camacho, O. Implementing positioning algorithms using accelerometers. *Free. Semicond.* **2007**, 1–13.
13. Abellanosa, C.B.; Lugpatan, R.P.J.; Pascua, D.A.D. Position estimation using inertial measurement unit (IMU) on a quadcopter in an enclosed environment. *Int. J. Comput. Commun. Instrum. Eng.* **2016**, *3*, 332–336.
14. Pastell, M.; Frondelius, L.; Järvinen, M.; Backman, J. Filtering methods to improve the accuracy of indoor positioning data for dairy cows. *Biosyst. Eng.* **2018**, *169*, 22–31. [CrossRef]

15. Bozkurt, S.; Elibol, G.; Gunal, S.; Yayan, U. A comparative study on machine learning algorithms for indoor positioning. In Proceedings of the 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), Madrid, Spain, 2–4 September 2015; pp. 1–8.
16. Muset, B.; Emerich, S. Distance measuring using accelerometer and gyroscope sensors. *Carpathian J. Electron. Comput. Eng.* **2012**, *5*, 83.
17. Feliz Alonso, R.; Zalama Casanova, E.; Gómez García-Bermejo, J. Pedestrian tracking using inertial sensors. *J. Phys. Agents* **2009**, *3*, 35–43. [CrossRef]
18. Zhou, H.; Stone, T.; Hu, H.; Harris, N. Use of multiple wearable inertial sensors in upper limb motion tracking. *Med. Eng. Phys.* **2008**, *30*, 123–133. [CrossRef]
19. El-Gohary, M.; McNames, J. Shoulder and elbow joint angle tracking with inertial sensors. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 2635–2641. [CrossRef] [PubMed]
20. Lang, P.; Kusej, A.; Pinz, A.; Brasseur, G. Inertial tracking for mobile augmented reality. In Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference, Anchorage, AK, USA, 21–23 May 2002; pp. 1583–1587.
21. Dong, Y.; Scisco, J.; Wilson, M.; Muth, E.; Hoover, A. Detecting periods of eating during free-living by tracking wrist motion. *IEEE J. Biomed. Health Inform.* **2013**, *18*, 1253–1260. [CrossRef] [PubMed]
22. Luzum, B. Navigation Principles of Positioning and Guidance. *Eos Trans. Am. Geophys. Union* **2004**, *85*, 110–110. [CrossRef]
23. Harle, R. A survey of indoor inertial positioning systems for pedestrians. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1281–1293. [CrossRef]
24. Sharp, I.; Yu, K. Sensor-based dead-reckoning for indoor positioning. *Phys. Commun.* **2014**, *13*, 4–16. [CrossRef]
25. Diaz, E.M.; Gonzalez, A.L.M.; de Ponte Müller, F. Standalone inertial pocket navigation system. In Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS 2014), Monterey, CA, USA, 5–8 May 2014; pp. 241–251.
26. Godha, S.; Cannon, M. GPS/MEMS INS integrated system for navigation in urban areas. *Gps Solut.* **2007**, *11*, 193–203. [CrossRef]
27. Zhang, H.; Yuan, W.; Shen, Q.; Li, T.; Chang, H. A handheld inertial pedestrian navigation system with accurate step modes and device poses recognition. *IEEE Sens. J.* **2014**, *15*, 1421–1429. [CrossRef]
28. Bird, J.; Arden, D. Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors [emerging opportunities for localization and tracking]. *IEEE Wirel. Commun.* **2011**, *18*, 28–35. [CrossRef]
29. Goyal, P.; Ribeiro, V.J.; Saran, H.; Kumar, A. Strap-down pedestrian dead-reckoning system. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2011), Guimaraes, Portugal, 21–23 September 2011; pp. 1–7.
30. Gusenbauer, D.; Isert, C.; Krösche, J. Self-contained indoor positioning on off-the-shelf mobile devices. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2010), Zurich, Switzerland, 15–17 September 2010; pp. 1–9.
31. Li, B.; Gallagher, T.; Dempster, A.G.; Rizos, C. How feasible is the use of magnetic field alone for indoor positioning? In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2012), Sydney, NSW, Australia, 13–15 November 2012; pp. 1–9.
32. Liu, Z.; Zhang, L.; Liu, Q.; Yin, Y.; Cheng, L.; Zimmermann, R. Fusion of magnetic and visual sensors for indoor localization: Infrastructure-free and more effective. *IEEE Trans. Multimed.* **2016**, *19*, 874–888. [CrossRef]
33. Penoyer, R. The alpha–beta filter. *C User J.* **1993**, *11*, 73–86.
34. Next Generation Inertial Measurement Unit x-io Technologies Limited. Available online: https://x-io.co.uk/ngimu/ (accessed on 20 August 2019).