

Article

3D Indoor Position Estimation Based on a UDU Factorization Extended Kalman Filter Structure Using Beacon Distance and Inertial Measurement Unit Data [†]

Tolga Bodrumlu ^{1,*}  and Fikret Caliskan ²¹ Mechatronics Engineering Department, Istanbul Technical University, Istanbul 34025, Turkey² Control and Automation Engineering Department, Istanbul Technical University, Istanbul 34025, Turkey; caliskan@itu.edu.tr

* Correspondence: bodrumlu@itu.edu.tr

[†] This work is an extended version of the paper Enhancing Indoor Position Estimation Accuracy: Integration of IMU, Raw Distance Data, and Extended Kalman Filter with Comparison to Vicon Indoor Positioning System Data, Online, 15–30 November 2023.

Abstract: The development of the GPS (Global Positioning System) and related advances have made it possible to conceive of an outdoor positioning system with great accuracy; however, for indoor positioning, more efficient, reliable, and cost-effective technology is required. There are a variety of techniques utilized for indoor positioning, such as those that are Wi-Fi, Bluetooth, infrared, ultrasound, magnetic, and visual-marker-based. This work aims to design an accurate position estimation algorithm by combining raw distance data from ultrasonic sensors (Marvelmind Beacon) and acceleration data from an inertial measurement unit (IMU), utilizing the extended Kalman filter (EKF) with UDU factorization (expressed as the product of a triangular, a diagonal, and the transpose of the triangular matrix) approach. Initially, a position estimate is calculated through the use of a recursive least squares (RLS) method with a trilateration algorithm, utilizing raw distance data. This solution is then combined with acceleration data collected from the Marvelmind sensor, resulting in a position solution akin to that of the GPS. The data were initially collected via the ROS (Robot Operating System) platform and then via the Pixhawk development card, with tests conducted using a combination of four fixed and one moving Marvelmind sensors, as well as three fixed and one moving sensors. The designed algorithm is found to produce accurate results for position estimation, and is subsequently implemented on an embedded development card (Pixhawk). The tests showed that the designed algorithm gives accurate results with centimeter precision. Furthermore, test results have shown that the UDU-EKF structure integrated into the embedded system is faster than the classical EKF.

Keywords: indoor positioning; extended Kalman filter; UDU factorization; sensor fusion

Citation: Bodrumlu, T.; Caliskan, F. 3D Indoor Position Estimation Based on a UDU Factorization Extended Kalman Filter Structure Using Beacon Distance and Inertial Measurement Unit Data. *Sensors* **2024**, *24*, 3048. <https://doi.org/10.3390/s24103048>

Academic Editor: José Luis Lázaro-Galilea

Received: 11 December 2023

Revised: 24 December 2023

Accepted: 4 January 2024

Published: 11 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the advancement of technology in recent years, positioning systems have advanced significantly; however, these systems typically take place outdoors, where GPS signals are readily available. For indoor environments, GPS signals are insufficient for positioning accuracy because they are not at the desired level. This situation creates problems for indoor unmanned aerial (UAVs) and ground vehicles (UGVs). Therefore, studies have begun on position estimation for indoor environments. There are lots of different approaches related to indoor positioning, such as those that are motion-captured-based, lidar-based, Wi-Fi-based, Bluetooth-based, ultra-wideband (UWB)-based, and IMU-based [1–3]. Motion-capturing-based systems like VICON and OptiTrack employ multiple high-speed cameras to determine the position of an object relative to one another; however, these systems have a drawback in that they require a complex setup and challenging calibration. Lidar-based

positioning approaches, such as Gmapping, Hector, and Cartographer [4–6], are also an option, but they only provide the two-dimensional position due to weight limitations, especially for UAVs.

Wi-Fi- and Bluetooth-based positioning techniques are widely used, with an accuracy of a few meters [7]; however, they have constraints regarding the required number of access points, costs, and energy consumption. Bluetooth Low Energy (BLE) and Wi-Fi operate on the same frequency. BLE is intended for short-range and energy-efficient communication using brief messages [8]. BLE-based location is usually achieved by placing proximity beacons at specific locations. Receivers determine their position by measuring the RSSI (receiver's distance from the sender) of the closest beacons.

UWB positioning is lightweight, has a straightforward design, provides stable positioning results, and can achieve a centimeter level of accuracy; however, relying solely on UWB does not fulfill the demands of high-precision indoor operations. Although an IMU is a popular sensor for determining orientation, its position estimates can accumulate errors over time due to drift [9,10].

Therefore, an algorithm is designed to perform more accurate indoor positioning estimation. This paper presents an algorithm that uses the EKF with UDU factorization to combine the data from an IMU and raw distance data to calculate an accurate position estimation. The algorithm begins by using the RLS method with a trilateration algorithm to obtain an initial position estimation. This estimation is then fused with data from the IMU's acceleration sensor.

The main contributions of this work are listed below:

- The use of a sensor fusion algorithm, utilizing the EKF structure, in conjunction with acceleration data from IMU sensors, to enhance the accuracy of position information obtained from distance data provided by beacon sensors via the RLS algorithm.
- The incorporation of a UDU factorization structure to reduce computation costs in embedded systems, in addition to the utilization of the EKF structure for sensor fusion.
- The ability of the designed algorithm to produce a solution even under suboptimal conditions, such as the use of only three beacon sensors instead of the ideal four.

In this article, detailed information will be provided on the position estimation algorithm, followed by a description of the data collection process from the real sensor, and, finally, the results of the real-time tests of the designed algorithm will be conveyed.

2. Position Estimation Algorithm

In this section, the structure of algorithms designed for position estimation will be discussed. Figure 1 illustrates the general framework of the algorithm. Initially, information related to the geometric approach will be provided, and, subsequently, detailed information about the sensor fusion aspect will be conveyed. The designed algorithm is intended to be applied in real-time systems. For this reason, the EKF structure has been decomposed into UDU factorization to reduce the computation cost. Hence, the UDU-EKF structure is employed instead of the classical EKF structure in this study.

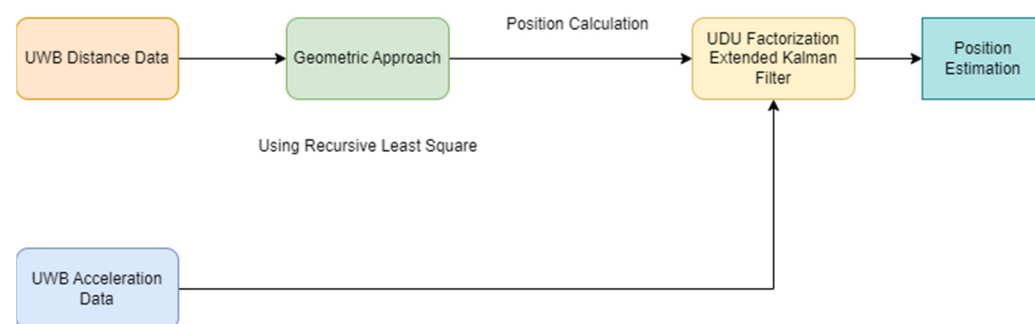


Figure 1. General structure of position estimation.

2.1. Geometric Approach

2.1.1. Solution Based on Three Reference Points

This study is based on a geometric method, as depicted in Figure 2, which involves three reference points, B_1 , B_2 , and B_3 , represented by their coordinates: (x_1, y_1, z_1) , (x_2, y_2, z_2) , and (x_3, y_3, z_3) , respectively. Additionally, the study also includes distance measurements, d_1 , d_2 , and d_3 , up to point A. The determination of the coordinates (x, y, z) of point A is equivalent to finding the solutions to the following system of quadratic equations [11–13].

$$\begin{aligned}(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= d_1^2 \\(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 &= d_2^2 \\(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 &= d_3^2\end{aligned}\quad (1)$$

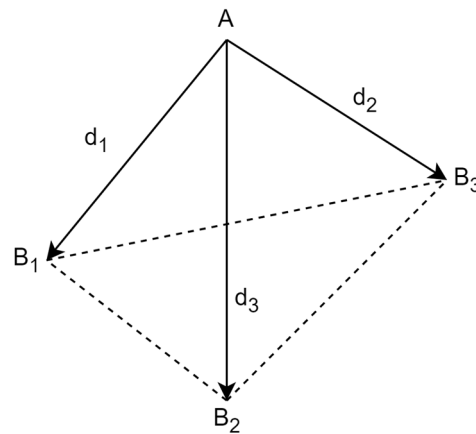


Figure 2. Reference points and interval measurements.

The equations in this system can be expressed as follows [11–13]:

$$\begin{aligned}(x^2 + y^2 + z^2) - 2x_1x - 2y_1y - 2z_1z &= d_1^2 - x_1^2 - y_1^2 - z_1^2 \\(x^2 + y^2 + z^2) - 2x_2x - 2y_2y - 2z_2z &= d_2^2 - x_2^2 - y_2^2 - z_2^2 \\(x^2 + y^2 + z^2) - 2x_3x - 2y_3y - 2z_3z &= d_3^2 - x_3^2 - y_3^2 - z_3^2\end{aligned}\quad (2)$$

The system of equations can be represented in matrix form as follows [11–13]:

$$A_0 u = b_0 \quad (3)$$

where

$$A_0 = \begin{bmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ 1 & -2x_3 & -2y_3 & -2z_3 \end{bmatrix}, u = \begin{bmatrix} x^2 + y^2 + z^2 \\ x \\ y \\ z \end{bmatrix}, b_0 = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ d_3^2 - x_3^2 - y_3^2 - z_3^2 \end{bmatrix} \quad (4)$$

The solution set of this system has two possible interpretations: In one interpretation, points B_1 , B_2 , and B_3 are not in a straight line. In the other interpretation, they are in a straight line [11–13].

Case 1. B_1 , B_2 , and B_3 are not in a straight line.

In this situation, the following propositions hold true:

- The rank of matrix A_0 is 3.
- The dimension of the null space of A_0 is 1.

The general solution to Equation (3) can be represented as follows:

$$u = u_p + t.u_h \quad (5)$$

where t is a real number coefficient, u_p is a particular solution of (3), and u_h is a solution of the homogenous system $A_0.u = 0$. Both u_p and u_h can be calculated using the Gaussian elimination method:

$$u_p = (u_{p0}, u_{p1}, u_{p2}, u_{p3})^T, u_h = (u_{h0}, u_{h1}, u_{h2}, u_{h3})^T, u = (u_0, u_1, u_2, u_3)^T \quad (6)$$

If we substitute the expression for u_p , u_h , and u into Equation (4), the following expressions are obtained:

$$\begin{aligned} u_0 &= u_{p0} + tu_{h0}, & u_1 &= u_{p1} + tu_{h1}, \\ u_2 &= u_{p2} + tu_{h2}, & u_3 &= u_{p3} + tu_{h3} \end{aligned} \quad (7)$$

By using the constraint $u \in E$:

$$u_{p0} + tu_{h0} = (u_{p1} + tu_{h1})^2 + (u_{p2} + tu_{h2})^2 + (u_{p3} + tu_{h3})^2 \quad (8)$$

and then

$$t^2(u_{h1}^2 + u_{h2}^2 + u_{h3}^2) + u_{p1}^2 + u_{p2}^2 + u_{p3}^2 - u_{p0} + t(2u_{p1}u_{h1} + 2u_{p2}u_{h2} + 2u_{p3}u_{h3} - u_{h0}) = 0 \quad (9)$$

This equation is a quadratic in the form of $at^2 + bt + c = 0$, and has two solutions:

$$t_{1/2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (10)$$

The solutions of the equation system can be presented as follows:

$$u_1 = u_p + t_1 u_h, \quad u_2 = u_p + t_2 u_h \quad (11)$$

Case 2. B_1, B_2 , and B_3 are in a straight line.

In this situation, the following propositions hold true:

- The rank of matrix A_0 is 2.
- The dimension of the null space of A_0 is 2.

The general solution to Equation (3) can be expressed as follows:

$$u = u_p + t.u_{h1} + k.u_{h2} \quad (12)$$

where u_p is a particular solution of Equation (3), and u_{h1} as well as u_{h2} are two linearly independent solutions of the homogeneous system described by $A_0.u = 0$.

2.1.2. Solution Based on More Than Three Reference Points

If there is an additional distance of $d_4, d_5, \dots, d_n$ to the reference points, $B_4, B_5 \dots B_n$ can be extend as follows [11–13]:

$$\begin{bmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ 1 & -2x_3 & -2y_3 & -2z_3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -2x_n & -2y_n & -2z_n \end{bmatrix} \begin{bmatrix} x^2 + y^2 + z^2 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ d_3^2 - x_3^2 - y_3^2 - z_3^2 \\ \vdots \\ d_n^2 - x_n^2 - y_n^2 - z_n^2 \end{bmatrix} \quad (13)$$

Expressed in a known form, this system can be written as follows:

$$Au = b \quad (14)$$

$$A = \begin{bmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ 1 & -2x_3 & -2y_3 & -2z_3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -2x_n & -2y_n & -2z_n \end{bmatrix}, u = \begin{bmatrix} x^2 + y^2 + z^2 \\ x \\ y \\ z \end{bmatrix}, b = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ d_3^2 - x_3^2 - y_3^2 - z_3^2 \\ \vdots \\ d_n^2 - x_n^2 - y_n^2 - z_n^2 \end{bmatrix}$$

The general solution, denoted as $\hat{u}$, can be obtained using the least squares method as follows:

$$\hat{u} = (A^T A)^{-1} A^T b \quad (15)$$

The projection of point p on the column space of matrix A is as follows:

$$p = A (A^T A)^{-1} A^T b \quad (16)$$

In this case, the coordinates of point p in the column space of $\text{Col}(A)$ correspond to the solution, $\hat{u}$. Moreover, vector b consists of distances between the unknown point, A , and all the reference points; however, if the measurements are uncorrelated but have varying levels of uncertainty, the weighted least squares (WLS) method is utilized. The solution, $\hat{u}$, can then be determined using the following equation:

$$\hat{u} = (A^T V^{-1} A)^{-1} A^T V^{-1} b \quad (17)$$

V is the covariance matrix of random errors. These errors refer to the variability or uncertainty in the measurements of the distances between points (represented by the vector u) and the reference points. It might be measurement noise (e.g., signal interference) or propagation effects (e.g., signal reflection or refraction). The RLS method uses u_0 as the initial solution. It is continually updated with each new distance measurement, resulting in the creation of a new solution, u_1 . This approach enables the simultaneous calculation of both distance measurements and positioning, allowing for the initiation of position assignment even when not all distances are known, thus reducing waiting time and improving the speed of the positioning calculation. Generally, for indoor positioning, a limited number of participating reference stations does not cause a big challenge to the computational resources for multilateration. This presented algorithm utilizes a linear algebraic approach with low computational complexity, making it suitable for real-time applications.

The combination of distance data and the RLS algorithm is used to calculate the position. The following section will go into further detail on how to improve position estimation using the EKF.

2.2. Sensor Fusion Algorithm

Several algorithms are used for sensor fusion, including Bayesian networks, convolutional neural networks, Dempster–Shafer theory, and the Kalman filter [14]. This system employs the Kalman filter to achieve more precise position estimation; however, the Kalman filter is optimized for linear systems, while real-world systems often exhibit non-linearity. Thus, the EKF is utilized to enhance performance [15]. The implementation of the EKF used in this study will be discussed in further detail. The state vector is defined as follows [13]:

$$X[k] = \begin{bmatrix} p(w)[k] \\ v(w)[k] \\ a(w)[k] \end{bmatrix} \quad (18)$$

The vector $p(w)$ represents the coordinates of the object along the x , y , and z axes in the world coordinate system, and is given as $[p_x[k], p_y[k], p_z[k]]$. The velocity vector of the object along the x , y , and z axes in the world coordinate system is represented by $v(w) = [v_x[k],$

$v_y[k], v_z[k]$. Additionally, the accelerometer vector of the object in the world coordinate system is represented by $a(w) = [a_x[k], a_y[k], a_z[k]]$.

The world coordinate system here (w) can also be specified as the coordinate system of the UWB sensor (Marvelmind). At this point, Δt is defined as the sampling time interval, and $(\Delta t)w[k]$ can be defined as the process noise of acceleration, representing uncertainties in the acceleration process. $\frac{\Delta t^2}{2}w[k], \frac{\Delta t^3}{6}w[k]$ can be represented as the process noise of velocity and process noise of positions, respectively. Equations of motion in the $(k + 1)$ time interval can be expressed as follows [13]:

$$p[k + 1] = p[k] + v[k](\Delta t) + a[k]\frac{\Delta t^2}{2} + \frac{\Delta t^3}{6}w[k] \quad (19)$$

The equation states that the new position at time step $k + 1$ is determined by the current position, the velocity multiplied by the time step, half of the acceleration multiplied by the square of the time step, and a noise term. The noise term is added to account for uncertainties or external disturbances that may affect the motion of the system:

$$v[k + 1] = v[k] + a[k](\Delta t) + \frac{\Delta t^2}{2}w[k] \quad (20)$$

The equation states that the new velocity at time step $k + 1$ is determined by the current velocity, the acceleration multiplied by the time step, and a noise term:

$$a[k + 1] = a[k] + (\Delta t)w[k] \quad (21)$$

The equation states that the new acceleration at time step $k + 1$ is determined by the current acceleration and a noise term.

The choice of noise term $w[k]$ depends on the characteristics of the system and the sources of uncertainty or disturbances. As mentioned earlier, it is common to assume $w[k]$ to be white noise. White noise has equal power at all frequencies, making it a simple and convenient choice in many applications.

The state equation can be represented in matrix form, as follows:

$$X[k + 1] = AX[k] + Gw[k] \quad (22)$$

The matrices A and G represent the transition matrix and noise process matrix, respectively. Process noise vector and Q covariance can be defined as follows [16–18]:

$$w[k] = [w_x[k]; w_y[k]; w_z[k]], \quad Q = \text{diag}\left(\begin{bmatrix} \sigma_{ax}^2 & \sigma_{ay}^2 & \sigma_{az}^2 \end{bmatrix}\right)$$

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \Delta t^2/2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 1 & 0 & \Delta t^2/2 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \Delta t^2/2 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} \frac{\Delta t^3}{6} & 0 & 0 \\ 0 & \frac{\Delta t^3}{6} & 0 \\ 0 & 0 & \frac{\Delta t^3}{6} \\ \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \quad (23)$$

The observation vector, $o[k]$, contains the distance values used in the geometric approach and an additional noise vector. Since the observation vector typically contains noisy or imperfect measurements of the true state of the system, the measurement vector, $H[k]$, represents the predicted measurements based on the current estimate of the system's

state. The observation vector contains the real-world measurements obtained from sensors, while the measurement vector represents the predicted measurements based on a current estimate of the system's state:

$$o[k] = \begin{bmatrix} d_1[k] + n_1[k] \\ d_2[k] + n_2[k] \\ d_3[k] + n_3[k] \\ d_4[k] + n_4[k] \end{bmatrix} \approx H[k]x[k] + n[k] \quad (24)$$

$d_1, \dots, d_4$ represents the true distance measurements. In addition to this, n_1, n_2, n_3 , and n_4 represent the noise associated with each measurement.

The measurement vector is represented by $H[k]$, and the vector of noise, with a mean of zero and a covariance matrix, is represented by $n[k]$. As seen in Figure 3, in the general structure, five Marvelmind ultrasonic sensors are used for the system, with one being mobile and four being stationary. The distances from the four stationary sensors to the mobile sensor can be defined as r_1, r_2, r_3 , and r_4 . The matrix, R , a covariance matrix, is defined by the covariances of these distance data as follows:

$$R = \text{diag} \left(\begin{bmatrix} \sigma_{r1}^2 & \sigma_{r2}^2 & \sigma_{r3}^2 & \sigma_{r4}^2 \end{bmatrix} \right) \quad (25)$$

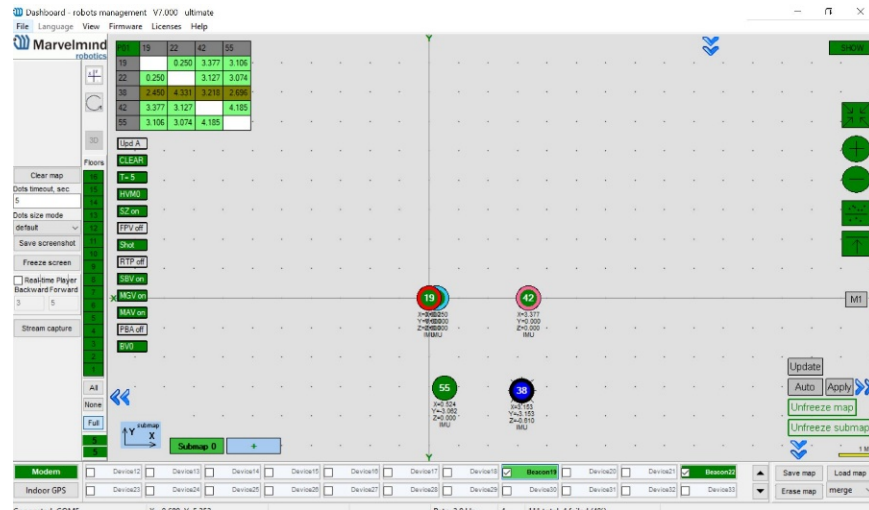


Figure 3. Displaying used sensors on the interface.

The equations described in the structure of the applied EKF are utilized. The initialization stage of this EKF is one of the crucial factors, where the initial values of the covariance matrices are assigned. Subsequently, the prediction and update steps are executed in sequence [15,19].

- (1) Prediction of state:

$$\bar{x}[k] = f(A\hat{x}[k-1], u[k]) \quad (26)$$

where f is a nonlinear transition function and $u[k]$ represents control inputs if applicable.

- (2) Prediction of state covariance:

$$\bar{P}[k] = A\hat{P}[k-1]A^T + Q \quad (27)$$

where A_{k-1} is the Jacobian matrix of the nonlinear transition function, f , with respect to the state, X .

- (3) Gain calculation of the EKF:

$$K[k] = \bar{P}[k]H[k]^T [H[k]\bar{P}[k]H[k]^T + R[k]]^{-1} \quad (28)$$

(4) State correction:

$$\hat{x}[k] = \bar{x}[k] + K[k](O[k] - h(\bar{x}[k])) \quad (29)$$

where h is the Jacobian matrix and represents the nonlinear measurement function.

(5) State covariance correction:

$$\hat{P}[k] = (I - K[k]H[k])\bar{P}[k] \quad (30)$$

In EKF, the complexity of computation is increased by calculating the Jacobian matrix. To enhance the efficiency and accuracy, and to decrease the computational cost of the EKF computations, UDU factorization is applied. The details of this algorithm are given in the next section.

2.3. UDU Factorization

UDU factorization employs the above equations. It substitutes the covariance matrix, $P[k]$, with an upper triangular matrix with ones on the diagonal ($U[k]$) and a diagonal matrix, $D[k]$, and can be expressed as follows [20]:

$$P[k] = U[k]D[k]U^T[k] \quad (31)$$

2.4. UDU Measurement Update

Reword the sentence by expressing the process of factorizing the covariance, P , into an LDL form, where instead of utilizing an upper triangular matrix a lower triangular matrix is used [21].

$$P[k] = L[k]\Delta[k]L^T[k] \text{ and } \bar{P}[k] = \bar{L}[k]\bar{\Delta}[k]\bar{L}^T[k] \quad (32)$$

where Δ is the diagonal matrix and U and D are the inverses of L^T and Δ , respectively [21].

$$\begin{aligned} P[k]^{-1} &= L[k]^{-T}\Delta[k]^{-1}L[k]^{-1} = U[k]D[k]U[k]^T \\ \bar{P}[k]^{-1} &= \bar{L}[k]^{-T}\bar{\Delta}[k]^{-1}\bar{L}[k]^{-1} = \bar{U}[k]\bar{D}[k]\bar{U}[k]^T \end{aligned} \quad (33)$$

Therefore, the measurement update becomes as follows:

$$U[k]D[k]U[k]^T = \bar{U}[k]\bar{D}[k]\bar{U}[k]^T + H[k]^T R[k]^{-1} H[k] \quad (34)$$

Then, factorize the $m \times m$ matrix R_k into the LDL form as described in [21]:

$$\begin{aligned} R[k] &= L_{R[k]}\Delta_{R[k]}L_{R[k]}^T \\ R[k]^{-1} &= L_{R[k]}^{-T}\Delta_{R[k]}^{-1}L_{R[k]}^{-1} = U_{R[k]}D_{R[k]}U_{R[k]}^T \end{aligned} \quad (35)$$

Therefore, (34) becomes as follows:

$$U[k]D[k]U[k]^T = \bar{U}[k]\bar{D}[k]\bar{U}[k]^T + H[k]^T U_{R[k]}D_{R[k]}U_{R[k]}^T H[k] \quad (36)$$

and the term $U_{R[k]}H$ can be expressed as follows [21]:

$$U_{R[k]}^T H_R = \begin{bmatrix} v_1^T \\ \vdots \\ v_m^T \end{bmatrix} \quad (37)$$

Each v_i is an $n \times 1$ vector. The factor $H[k]^T R[k]^{-1} H[k]$ can be expressed as follows [21]:

$$\begin{aligned} H[k]^T R[k]^{-1} H[k] &= H[k]^T U_{R[k]} D_{R[k]} U_{R[k]}^T H[k] \\ &= \begin{bmatrix} v_1^T \\ \vdots \\ v_m^T \end{bmatrix}^T \begin{bmatrix} \frac{1}{d_{1R}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{d_{mR}} \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_m^T \end{bmatrix} = \sum_{i=1}^m \frac{1}{d_{iR}} v_i v_i^T \end{aligned} \quad (38)$$

Therefore, the measurement update is expressed as follows:

$$U[k] D[k] U[k]^T = \bar{U}[k] \bar{D}[k] \bar{U}[k]^T + \sum_{i=1}^m \frac{1}{d_{iR}} v_i v_i^T \quad (39)$$

In addition to this, $\hat{z}[k]$ and $\bar{z}[k]$, which are related to $\hat{x}_k$ and $\bar{x}_k$, are defined below [20,21]:

$$\begin{aligned} \hat{z}[k] &\triangleq P[k]^{-1} \hat{x}[k], \quad \bar{z}[k] \triangleq \bar{P}[k]^{-1} \bar{x}[k] \\ \hat{z}[k] &= P[k]^{-1} (I - K[k] H[k]) \bar{x}[k] + P[k]^{-1} K[k] m[k] \\ \hat{z}[k] &= \bar{z}[k] + H[k]^T R[k]^{-1} m[k] \end{aligned} \quad (40)$$

where $m[k]$ is the measurement vector.

2.5. UDU Time Update

The Kalman filter uses a standard method. It involves inverting the information matrix to obtain the covariance, then propagating the covariance, and finally inverting the propagated covariance matrix to prepare for the measurement update step. This proposed algorithm utilizes this method for covariance propagation and factorizing $Q[k]$ with UDU parameterization [20,21]:

$$Q[k] = U_{Q[k]} \Delta_{Q[k]} U_{Q[k]}^T \quad (41)$$

where $\Delta_{Q[k]}$ is a diagonal $p \times p$ matrix, and $G_{Q[k]}$ is defined as follows:

$$G_{Q[k]} \triangleq G[k] U_{Q[k]} \quad (42)$$

so $\bar{P}[k]$ becomes the following:

$$\bar{P}[k+1] = \phi[k] P[k] \phi[k]^T + G_{Q[k]} \Delta_{Q[k]} G_{Q[k]}^T \quad (43)$$

Using the matrix inversion lemma, we obtain the following:

$$\begin{aligned} (Q + WEF)^{-1} &= Q^{-1} - Q^{-1} W (E^{-1} + FQ^{-1}W)^{-1} FQ^{-1} \\ Q &= \phi[k] P[k] \phi[k]^T; \quad E = \Delta_{Q[k]}; \quad W = G_{Q[k]}; \quad F = G_{Q[k]}^T \\ Q^{-1} &= M[k] \phi[k]^{-1} P[k]^{-1} \phi[k]^{-1} \end{aligned} \quad (44)$$

The inversion of the propagated covariance matrix is as follows [20,21]:

$$\begin{aligned} \bar{P}[k+1]^{-1} &= M[k] - M[k] G_{Q[k]} \left[G[k]^T M[k] G_{Q[k]} + \Delta_{Q[k]}^{-1} \right]^{-1} G_{Q[k]}^T M[k] \\ \bar{G}[k] &= \phi[k]^{-1} G_{Q[k]}, \quad D_{Q[k]} = \Delta_{Q[k]}^{-1} \end{aligned} \quad (45)$$

Then, the expression for $P[k+1]^{-1}$ is given by the following:

$$P[k+1]^{-1} = \phi[k]^{-T} \left\{ P[k]^{-1} - P[k]^{-1} \bar{G}[k] \left[\bar{G}[k]^T P[k]^{-1} \bar{G}[k] + D_{Q[k]} \right]^{-1} \bar{G}[k]^T P[k]^{-1} \right\} \phi[k]^{-1} \quad (46)$$

and the Kalman (K_{UDU}) gain is defined as follows:

$$K_{UDU[k]} \triangleq P[k]^{-1} \bar{G}[k] \left[\bar{G}[k]^T P[k]^{-1} \bar{G}[k] + D_{Q[k]} \right]^{-1} \quad (47)$$

Then, if we define brackets in (47) as $\mathcal{P}[k]^{-1}$, we obtain the following:

$$\begin{aligned} \mathcal{P}[k]^{-1} &\triangleq P[k]^{-1} - P[k]^{-1} \bar{G}[k] \left[\bar{G}[k]^T P[k]^{-1} \bar{G}[k] + D_{Q[k]} \right]^{-1} \bar{G}[k]^T P[k]^{-1} \\ \mathcal{P}[k]^{-1} &= \left[I - K_{UDU[k]} \bar{G}[k]^T \right] P[k]^{-1} \end{aligned} \quad (48)$$

Equation (48) is an analog to Equations (28)–(30), with the following:

$$\mathcal{P}[k]^{-1} \rightarrow \hat{P}[k], P[k]^{-1} \rightarrow \bar{P}[k], \bar{G}[k] \rightarrow H[k]^T, K_{UDU[k]} \rightarrow K[k], D_{Q[k]} \rightarrow R[k] \quad (49)$$

and since $D_{Q[k]}$ is the diagonal matrix, the UDU factorization of $\mathcal{P}[k]^{-1}$ is solved directly by using Carlson's rank-one update [22]:

$$\begin{aligned} U[k] \mathcal{D}[k] U[k]^T &= U[k] D[k] U[k]^T \\ &\quad - U[k] D[k] U[k]^T \bar{G}[k] \left[\bar{G}[k]^T U[k] D[k] U[k]^T \bar{G}[k] \right. \\ &\quad \left. + \Delta_{Q[k]}^{-1} \right] \bar{G}[k]^T U[k] D[k] U[k]^T \end{aligned} \quad (50)$$

and time-propagated UDU factors of $\bar{P}[k+1]^{-1}$ are defined as below:

$$\bar{P}[k+1]^{-1} = \bar{U}[k+1] \bar{D}[k+1] \bar{U}[k+1]^T = \phi[k]^{-T} U[k] \mathcal{D}[k] U[k]^T \phi[k]^{-1} \quad (51)$$

The time update for $\bar{z}[k+1]$ is obtained as follows.

$$\bar{P}[k+1]^{-1} \bar{x}[k+1] = \bar{P}[k+1]^{-1} \phi[k] P[k] P[k]^{-1} \hat{x}[k] \quad (52)$$

then it becomes

$$\bar{z}[k+1] = \bar{P}[k+1]^{-1} \phi[k] P[k] \hat{z}[k] \quad (53)$$

and it can be written as:

$$\bar{z}[k+1] = \phi[k]^{-T} \left[I - K_{UDU[k]} \bar{G}[k]^T \right] \hat{z}[k] \quad (54)$$

The equation systems specified in this section have been utilized to enhance the performance of the EKF structure in real-time applications. The outcomes of this structure will be comprehensively analyzed in Section 4 of the article. The next section will explain how data are collected from the Marvelmind sensor.

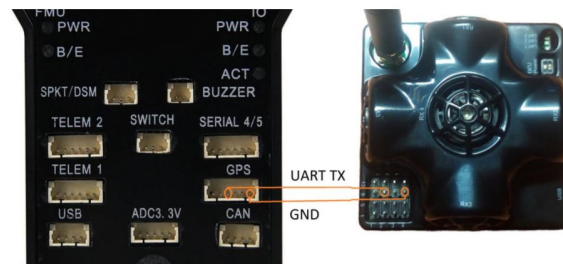
3. Data Acquisition

This section will provide detail on how to retrieve data from the Marvelmind sensor. The mobile Marvelmind sensor is connected to a laptop computer via the serial port. To ensure that all of the necessary components are working correctly, four Marvelmind sensors attached to the wall were activated, and their functionality was confirmed through the Marvelmind's own interface [23]. After verifying that all of the sensors were operating properly, a software tool on the ROS platform was run on the laptop to gather the data shown in Table 1.

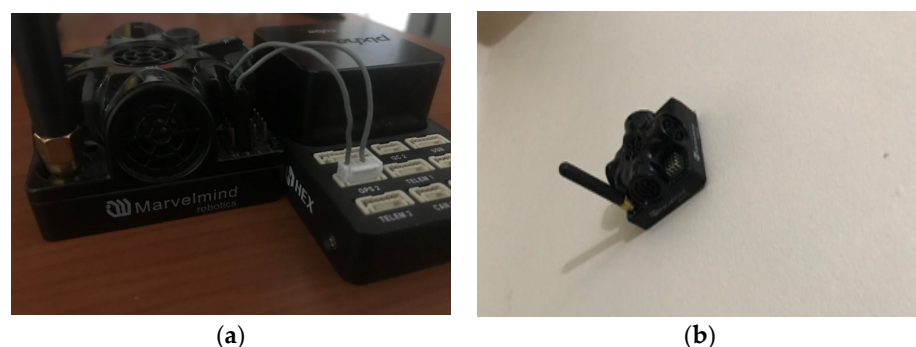
Table 1. Marvelmind sensor data structure [23].

Message	Message Variable	Data Type	Description of Data
beacon_distance	dist_m	float64	Raw distance data of beacon, in meters
	add_hedge	uint8	Address no of mobile beacon
	add_beacon	uint8	Address no of stationary beacon
hedge_imu_raw	time_var	int64	Timestamp of IMU data
	accel_x	int16	Accelerometer of x data
	accel_y	int16	Accelerometer of y data
	accel_z	int16	Accelerometer of z data

After the data were collected in the ROS environment, the packet parsing functions that interpret the data were added to Pixhawk in order to be able to collect them in the same way from Pixhawk as well. In addition to the written codes, a physical connection must also be made between Pixhawk and the Marvelmind sensor. This physical connection schema can be seen in Figure 4 [24].

**Figure 4.** Pixhawk physical wiring diagram with the Marvelmind sensor [24].

Following the schematic connection depicted in Figure 5a,b, a physical connection between Pixhawk and the Marvelmind sensor was established. Data collection was then conducted on Pixhawk, while the other Marvelmind sensors remained fixed to the wall.

**Figure 5.** (a). Pixhawk physical connection with the Marvelmind sensor (b). Marvelmind sensor fixed on the wall.

The explanation of the algorithm tests and their outcomes will be provided in detail in the next section.

4. Testing Algorithm and Results

The UDU-factorized EKF structure and the geometric approach based on the RLS method were implemented in the C++ environment using MATLAB code generation support. The integration was carried out on the Pixhawk platform. The trajectories obtained from the Marvelmind sensor are presented in Table 2. The robotic platform used

in this study is a quadcopter. Due to reliability concerns associated with indoor flight, the quadcopter was constrained to a fixed table, and data were collected using Marvelmind sensors. In the initial six trajectories, no motion was applied to the robot along the z-axis; however, intentional movement in the z-axis was introduced during the data collection for the last two trajectories to assess the algorithm's performance.

Table 2. Trajectories of the tests.

Trajectory ID	Beacon Number	Trajectories
1	5 (4 stationary + 1 mobile)	
2	5 (4 stationary + 1 mobile)	
3	5 (4 stationary + 1 mobile)	
4	4 (3 stationary + 1 mobile)	
5	4 (3 stationary + 1 mobile)	
6	4 (3 stationary + 1 mobile)	
7	5 (4 stationary + 1 mobile)	With a different z value 
8	4 (3 stationary + 1 mobile)	With a different z value 

The position estimations, calculated solely through the trilateration algorithm, and the positions derived by integrating the trilateration with the EKF and UDU factorization algorithms, incorporating IMU data, are presented in Figures 6–13. Additionally, the locations of the fixed sensors used in these figures are indicated as black dots. Comprehensive information on the minimum, maximum, and average error values for the calculated positions is provided in Table 3, generated using the RMS value.

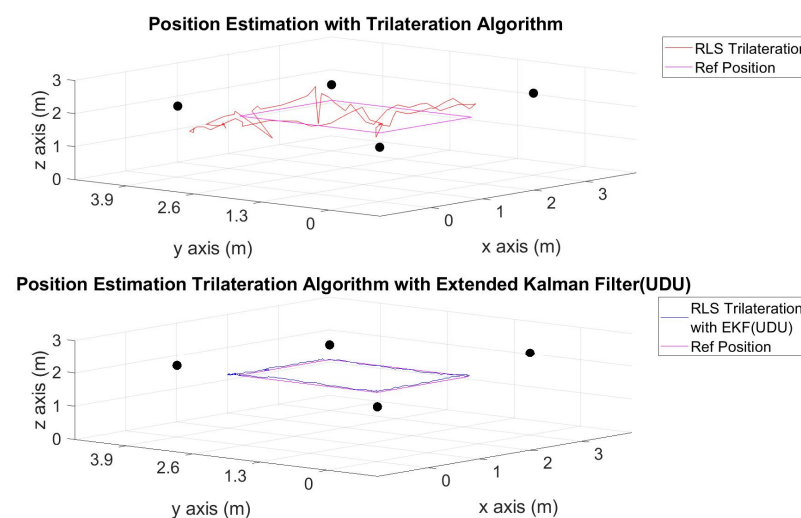


Figure 6. Position-estimation-only trilateration algorithm and trilateration algorithm with the EKF in Trajectory 1.

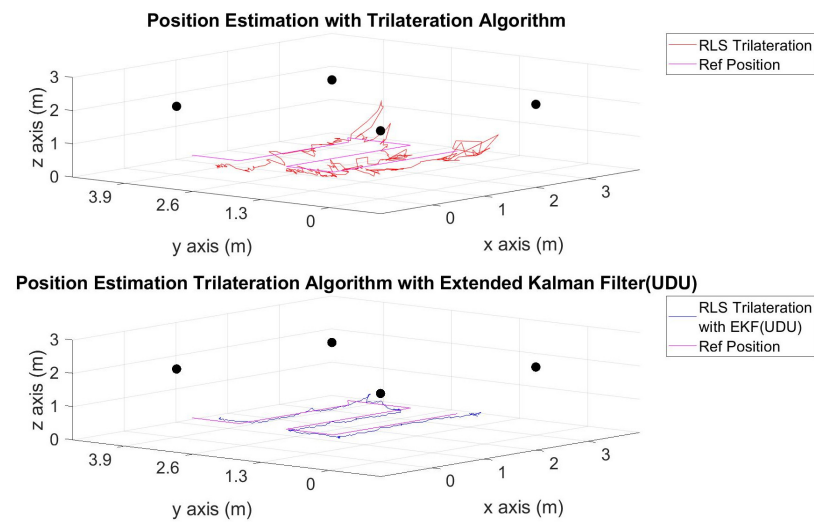


Figure 7. Position-estimation-only trilateration algorithm and trilateration algorithm with the EKF in Trajectory 2.

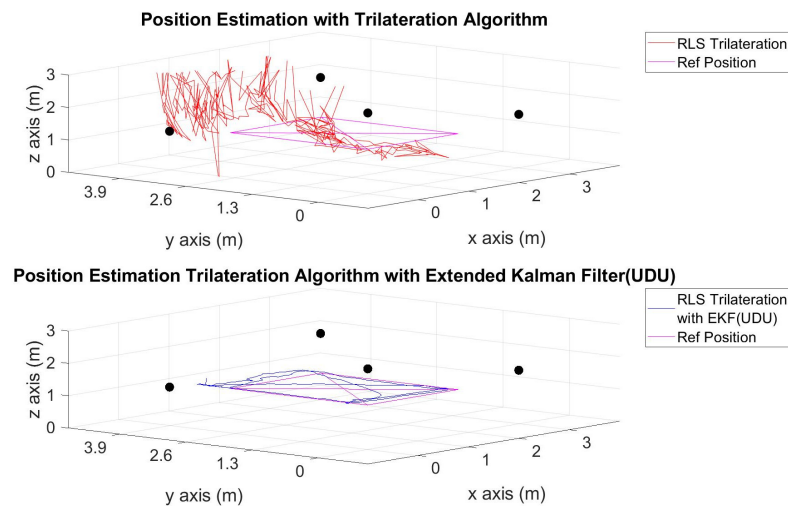


Figure 8. Position-estimation-only trilateration algorithm and trilateration algorithm with the EKF in Trajectory 3.

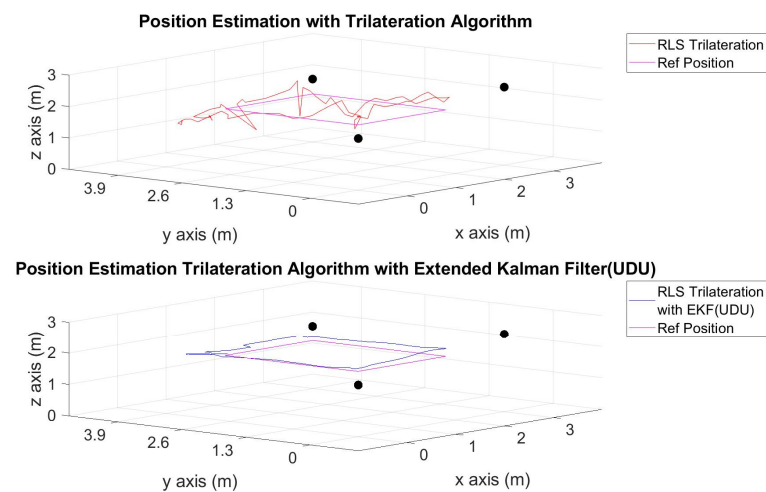


Figure 9. Position-estimation-only trilateration algorithm and trilateration algorithm with the EKF in Trajectory 4.

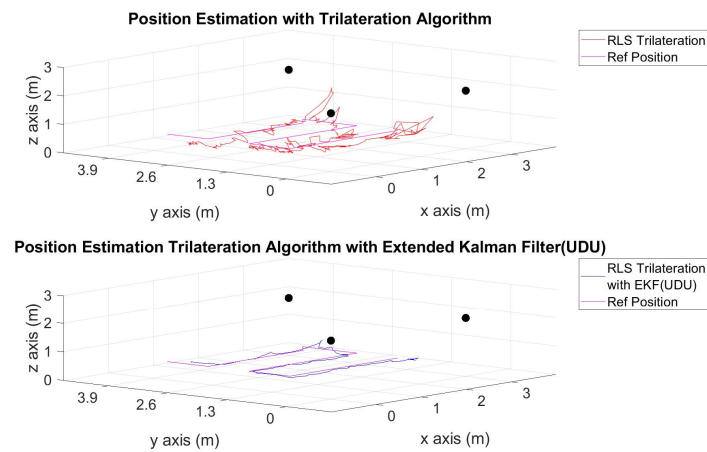


Figure 10. Position-estimation-only trilateration algorithm and trilateration algorithm with the EKF in Trajectory 5.

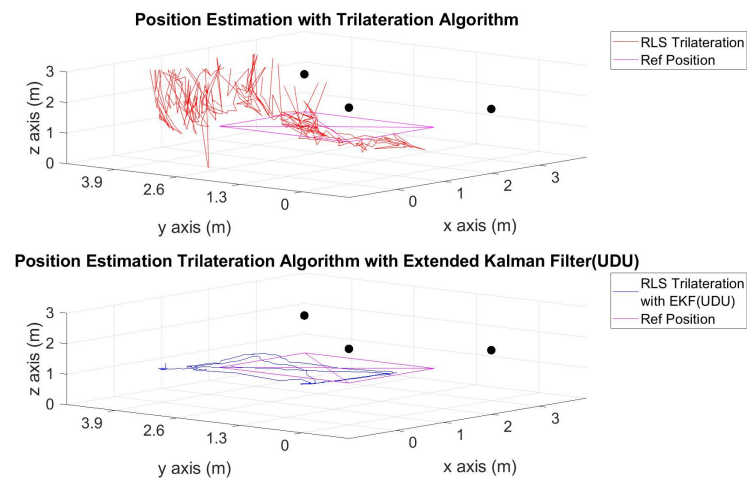


Figure 11. Position-estimation-only trilateration algorithm and trilateration algorithm with the EKF in Trajectory 6.

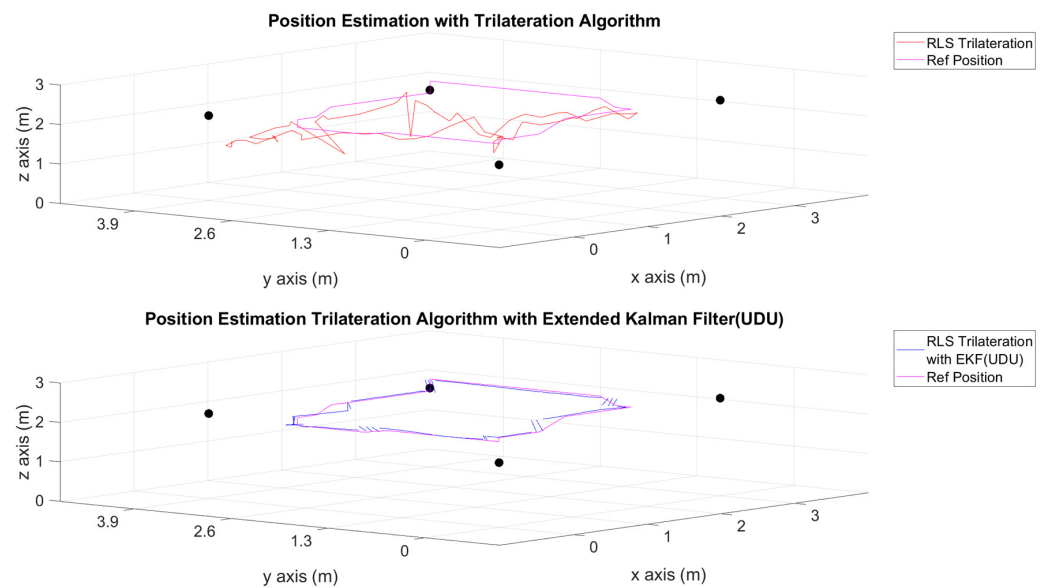


Figure 12. Position-estimation-only trilateration algorithm and trilateration algorithm with the EKF in Trajectory 7.

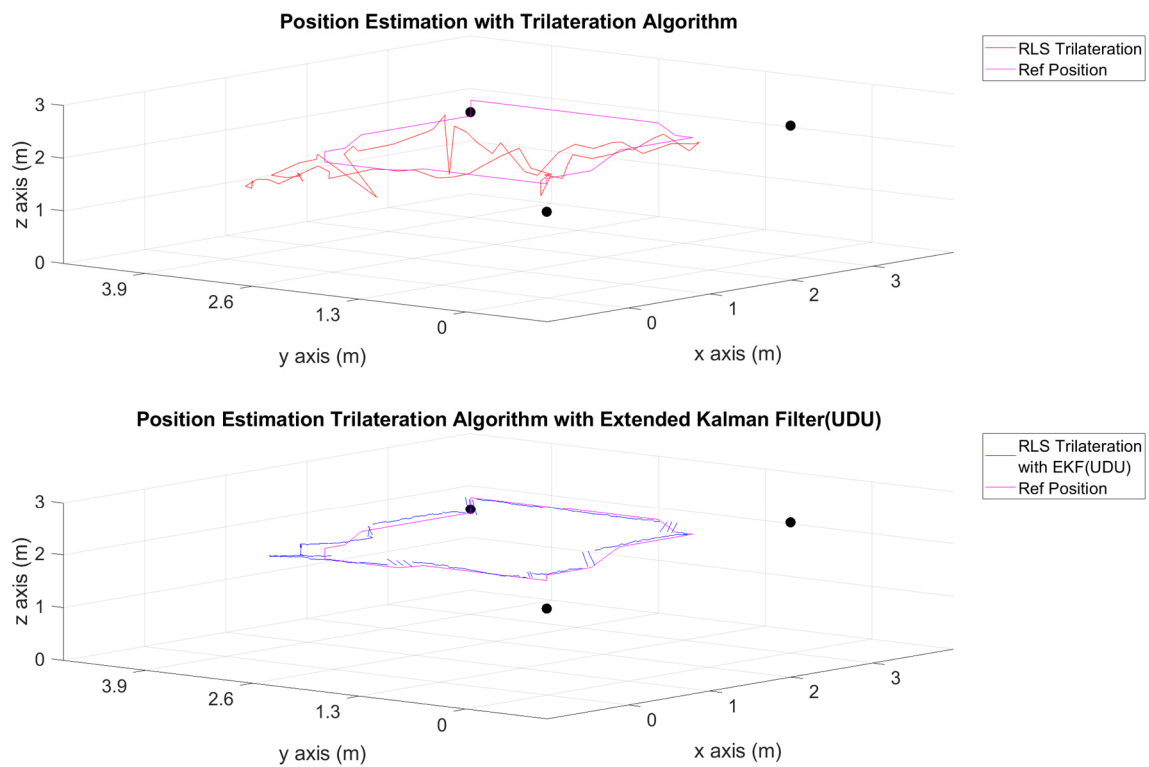


Figure 13. Position-estimation-only trilateration algorithm and trilateration algorithm with the EKF in Trajectory 8.

Table 3. Error tables.

Trajectory ID	Algorithm	Min Error (m)	Mean Error (m)	Max Error (m)
1	Only geometric approach	0.0734	0.3356	0.9940
1	Geometric approach with the EKF (UDU)	0.0272	0.0617	0.1439
2	Only geometric approach	0.0776	0.4533	1.7680
2	Geometric approach with the EKF (UDU)	0.0177	0.0827	0.1686
3	Only geometric approach	0.3536	1.2583	5.0031
3	Geometric approach with the EKF (UDU)	0.0191	0.1034	0.2148
4	Only geometric approach	0.1142	0.3628	1.1472
4	Geometric approach with the EKF (UDU)	0.0430	0.2070	0.4387
5	Only geometric approach	0.1348	0.5312	2.1322
5	Geometric approach with the EKF (UDU)	0.0752	0.1660	0.3361
6	Only geometric approach	0.4612	1.436	5.4721
6	Geometric approach with the EKF (UDU)	0.1141	0.2378	0.4796
7	Only geometric approach	0.0682	0.3781	1.2343
7	Geometric approach with the EKF (UDU)	0.0305	0.1312	0.1980
8	Only geometric approach	0.0836	0.4116	1.3841
8	Geometric approach with the EKF (UDU)	0.0504	0.2162	0.3226

It is evident from the figures that there are differences between the position estimations obtained using UDU factorization applied to the EKF structure and those obtained solely through the geometric approach. The position estimation derived from UDU factorization is closer to the reference position compared to the geometric approach. This is expected

because the geometric approach calculates position estimations solely based on distance data from the Marvelmind sensor. On the other hand, in the approach utilizing the EKF, a sensor fusion algorithm is employed, incorporating both distance and accelerometer data to enhance the accuracy of position estimations.

In Figures 6–8, four mobile and one fixed Marvelmind sensors were utilized. In Figures 9–11, three mobile and one fixed Marvelmind sensors were used. In Figures 12 and 13, both four mobile and one fixed in addition to three mobile and one fixed Marvelmind sensors were sequentially employed. In the first six scenarios, there was no movement along the z-axis. Conversely, in the last two scenarios, motion along the z-axis was introduced. At this point, the obtained RMS values exhibit differences depending on the trajectories, the number of sensors used, and the algorithmic structure.

When examining Table 3, it is apparent that the results obtained solely through the geometric approach for all trajectories fall short of the desired accuracy, with a notably high error amount; however, it is observed that position estimations reached the desired levels when UDU-EKF was employed in conjunction with the geometric approach. The primary reason for this is, as detailed in the article, that the sensor fusion algorithm is capable of providing a more accurate estimation compared to the geometric-based algorithm. Incorporating accelerometer data into the position estimation process contributes to achieving a more accurate outcome.

For each trajectory, the geometric approach, relying solely on trilateration, exhibits notable variability in accuracy across trajectories, with maximum errors reaching up to 5.4721 m in Trajectory 6. In contrast, the integration of UDU factorization into the EKF consistently enhances accuracy, as demonstrated by reduced minimum, mean, and maximum errors. Trajectory 6 stands out, with a remarkable improvement, showcasing a substantial drop in the maximum error from 5.4721 to 0.4796 m. This underscores the pivotal role of UDU-EKF in refining position estimates, particularly in scenarios involving intentional z-axis movement. The overall pattern highlights the algorithm's adaptability and robustness in dynamic indoor environments, reaffirming its efficacy for precise positioning in real-world applications.

In addition to that, the code structure was running multiple times for the comparison of UDU factorization EKF with normal EKF in terms of processing speeds, and the processing times for both structures were observed. The comparison of processing speeds between the EKF with UDU factorization and the conventional EKF, as detailed in Table 4, offers valuable insights into their computational efficiency. The experimental setup involved multiple runs to ensure a thorough assessment, and the recorded processing times for UDU-EKF and the EKF, presented in seconds for each trajectory, quantify the time required for algorithm execution. The observed results confirm the consistent outperformance of the EKF structure with UDU factorization, with a minimum increase in the processing time of 18%. The percentage difference analysis underscores the substantial improvement in processing speed achieved by incorporating UDU factorization. Trajectory-specific performance evaluations reveal notable percentage differences of 21%, 25%, and 18% for Trajectories 1, 2, and 3, respectively, emphasizing UDU-EKF's faster execution across diverse scenarios. The implications for real-world applications are substantial, especially in time-sensitive scenarios where quick and accurate positioning is paramount. The observed reduction in computational cost further positions UDU factorization as an efficient choice, particularly for resource-constrained systems. These findings suggest that implementing UDU factorization in the EKF structure can be considered a strategic optimization for improving overall algorithm efficiency. In summary, the analysis highlights UDU-EKF's superiority in processing speed, showcasing its potential to significantly enhance the efficiency of positioning algorithms in critical real-world applications.

Table 4. Process time difference of the EKF vs. UDU-EKF.

Trajectory	UDU-EKF (s)	EKF (s)	Percentage of Difference
1	0.0094	0.0119	21%
2	0.0076	0.0101	25%
3	0.0099	0.0121	18%

As a result, the integration of UDU factorization into the EKF structure has been observed to improve the accuracy of position estimation, as evidenced by both the graphs and Table 3. UDU-EKF achieves reliable results even in scenarios involving motion along the three axes. Additionally, upon reviewing Table 4, it is evident that the UDU-EKF structure operates faster than the traditional EKF. Considering both algorithms, it is observed that UDU-EKF plays a significant role in position estimation with lower computational cost.

5. Conclusions

This paper presents a comprehensive study on the integration of UDU factorization into the structure of the EKF for indoor position estimation algorithms. The proposed algorithm demonstrates its effectiveness in real-time systems, showcasing improved accuracy in position estimation compared to traditional EKF approaches. The decomposition of the EKF structure with UDU factorization aims to reduce computational costs, making it particularly suitable for real-time applications.

The study focuses on a geometric method based on reference points, introducing an approach that utilizes distance measurements and quadratic equations to determine the position coordinates of a target point.

Experimental results, conducted using Marvelmind sensors and a quadcopter, provide a thorough analysis of the algorithm's performance. Trajectory-based comparisons emphasize the superiority of UDU-EKF over the geometric approach, highlighting its ability to achieve more accurate position predictions. The integration of UDU factorization into the EKF not only enhances accuracy but also demonstrates adaptability to dynamic indoor environments.

Furthermore, the paper investigates the computational efficiency of the proposed algorithm, comparing processing speeds between UDU-EKF and traditional EKF structures. The results consistently show that UDU-EKF outperforms traditional EKF in terms of processing speed. This finding is crucial for real-world applications, especially those requiring quick and accurate positioning.

In summary, the integration of UDU factorization into the EKF structure emerges as a strategic optimization for achieving higher accuracy with lower computational costs. The algorithm's adaptability to all three axes and its superior processing speed position it as a promising solution for real-time positioning systems, emphasizing its potential impact on various applications, including robotics and indoor navigation.

Author Contributions: Conceptualization, T.B. and F.C.; methodology, T.B.; software, T.B.; validation, T.B.; data acquisition, T.B.; formal analysis, T.B.; investigation, T.B.; writing—original draft preparation, T.B.; writing—review and editing, T.B. and F.C.; visualization, T.B.; supervision, F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially partly supported by Istanbul Technical University, grant number 42754.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the work and article.

Conflicts of Interest: The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

References

1. Nirjon, S.; Liu, J.; DeJean, G.; Priyantha, B.; Jin, Y.; Hart, T. COIN-GPS: Indoor localization from direct GPS receiving. In Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services—MobiSys 2014, Bretton Woods, NH, USA, 16–19 June 2014; pp. 301–314.
2. Vasisht, D.; Kumar, S.; Katabi, D. Decimeter-Level Localization with a Single WiFi Access Point. In Proceedings of the USENIX Symposium on Networked Systems Design and Implementation, Santa Clara, CA, USA, 16–18 March 2016; pp. 165–178.
3. Zafari, F.; Papapanagiotou, I.; Christidis, K. Microlocation for internet-of-things-equipped smart buildings. *IEEE Internet Things J.* **2016**, *3*, 96–112. [\[CrossRef\]](#)
4. Grisetti, G.; Stachniss, C.; Burgard, W. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. Robot.* **2007**, *23*, 34–46. [\[CrossRef\]](#)
5. Kohlbrecher, S.; von Stryk, O.; Meyer, J.; Klingauf, U. A flexible and scalable SLAM system with full 3D motion estimation. In Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 1–5 November 2011; pp. 155–160.
6. Ren, R.; Fu, H.; Wu, M. Large-scale outdoor SLAM based on 2D lidar. *Electronics* **2019**, *8*, 613. [\[CrossRef\]](#)
7. Chintalapudi, K.; Padmanabha Iyer, A.; Padmanabhan, V.N. Indoor localization without the pain. In Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, Chicago, IL, USA, 20–24 September 2010; pp. 173–184.
8. Gomez, C.; Oller, J.; Paradells, J. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors* **2012**, *12*, 11734–11753. [\[CrossRef\]](#)
9. Aiello, G.R.; Rogerson, G.D. Ultra-wideband wireless systems. *IEEE Microw. Mag.* **2003**, *4*, 36–47. [\[CrossRef\]](#)
10. Chehri, A.; Fortier, P.; Tardif, P.M. UWB-based sensor networks for localization in mining environments. *Ad Hoc Netw.* **2009**, *7*, 987–1000. [\[CrossRef\]](#)
11. Norrdine, A. An Algebraic Solution to the Multilateration Problem. In Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation, Sydney, Australia, 13–15 November 2012; pp. 1–4.
12. Bodrumlu, T.; Caliskan, F. Indoor Position Estimation Using Ultrasonic Beacon Sensors and Extended Kalman Filter. *Eng. Proc.* **2022**, *27*, 16.
13. Bodrumlu, T.; Çalışkan, F. Enhancing Indoor Position Estimation Accuracy: Integration of IMU, Raw Distance Data, and Extended Kalman Filter with Comparison to Vicon Indoor Positioning System Data. In Proceedings of the 10th International Electronic Conference on Sensors and Applications, Online, 15–30 November 2023; MDPI: Basel, Switzerland, 2023. [\[CrossRef\]](#)
14. Fung, M.L.; Chen, M.Z.Q.; Chen, Y.H. Sensor fusion: A review of methods and applications. In Proceedings of the 2017 29th Chinese Control and Decision Conference (CCDC), Chongqing, China, 28–30 May 2017; pp. 3853–3860.
15. Krishnaveni, B.V.; Reddy, K.S.; Reddy, P.R. Indoor Tracking by Adding IMU and UWB Using Unscented Kalman Filter. *Wirel. Pers. Commun.* **2022**, *123*, 3575–3596. [\[CrossRef\]](#)
16. You, W.; Li, F.; Liao, L.; Huang, M. Data Fusion of UWB and IMU Based on Unscented Kalman Filter for Indoor Localization of Quadrotor UAV. *IEEE Access* **2020**, *8*, 64971–64981. [\[CrossRef\]](#)
17. Benini, A.; Mancini, A.; Marinelli, A.; Longhi, S. A Biased Extended Kalman Filter for Indoor Localization of a Mobile Agent using Low-Cost IMU and UWB Wireless Sensor Network. In Proceedings of the 10th IFAC Symposium on Robot Control International Federation of Automatic Control, Dubrovnik, Croatia, 5–7 September 2012.
18. Kwon, S.-G.; Kwon, O.-J.; Kwon, K.-R.; Lee, S.-H. UWB and MEMS IMU Integrated Positioning Algorithm for a Work-Tool Tracking System. *Appl. Sci.* **2021**, *11*, 8826. [\[CrossRef\]](#)
19. Oh, K.H.; Ahn, H.S. Extended Kalman Filter with Multi-frequency Reference Data for Quadrotor Navigation. In Proceedings of the 2015 15th International Conference on Control, Automation and Systems (ICCAS 2015), Busan, Republic of Korea, 13–16 October 2015.
20. Asl, H.G.; Pourtakdoust, S.H. UD Covariance Factorization for Unscented Kalman Filter using Sequential Measurements Update. *Int. J. Aerosp. Mech. Eng.* **2007**, *1*, 564–572.
21. D'souza, C.; Zanetti, R. Information Formulation of the UDU Kalman Filter. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 493–498. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Carlson, N. Fast triangular factorization of the square root filter. *AIAA J.* **1973**, *11*, 1259–1265. [\[CrossRef\]](#)
23. Marvelmind Indoor Navigation System Operating Manual. Available online: https://marvelmind.com/wp-content/uploads/2017/08/marvelmind_navigation_system_manual.pdf (accessed on 20 July 2017).
24. PixHawk and Marvelmind Integration Manual. Available online: https://marvelmind.com/pics/PixHawk_Marvelmind_Integration_Manual.pdf (accessed on 1 December 2017).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.