

# Enhancing Indoor Position Estimation Accuracy: Integration of Accelerometer, Raw Distance Data, and Extended Kalman Filter in Comparison to Vicon Motion Capture Data <sup>†</sup>

Tolga Bodrumlu <sup>1,\*</sup>  and Fikret Çalışkan <sup>2</sup> <sup>1</sup> Mechatronics Engineering Department, Istanbul Technical University, Istanbul 34025, Turkey<sup>2</sup> Control and Automation Engineering Department, Istanbul Technical University, Istanbul 34025, Turkey; caliskanf@itu.edu.tr

\* Correspondence: bodrumlu@itu.edu.tr

<sup>†</sup> Presented at the 10th International Electronic Conference on Sensors and Applications (ECSA-10), 15–30 November 2023; Available online: <https://ecsa-10.sciforum.net/>.

**Abstract:** Indoor positioning systems are a significant area of research and development, helping people navigate within buildings where GPS signals are unavailable. These systems have diverse applications, including aiding navigation in places like shopping malls, airports, and hospitals and improving emergency evacuation processes. The purpose of this study is to evaluate various technologies and algorithms used in indoor positioning. This study focuses on using raw distance data and Kalman filters to enhance indoor position accuracy. It employs a trilateration algorithm based on Recursive Least Squares (RLS) for initial position estimation and combines the results with accelerometer data. The designed algorithm using real sensor data collected in an ROS(Robot Operating System) environment was tested, and the results obtained were compared with data obtained from the Vicon Indoor Positioning System. In this comparison, the Root Mean Square Error metric was used. As a result of the comparison, it was observed that the error obtained from the designed algorithm is less than that of the Vicon system.

**Keywords:** indoor positioning; extended Kalman filter; ROS; sensor fusion



**Citation:** Bodrumlu, T.; Çalışkan, F. Enhancing Indoor Position Estimation Accuracy: Integration of Accelerometer, Raw Distance Data, and Extended Kalman Filter in Comparison to Vicon Motion Capture Data. *Eng. Proc.* **2023**, *58*, 40. <https://doi.org/10.3390/ecsa-10-16089>

Academic Editor: Stefano Mariani

Published: 15 November 2023



**Copyright:** © 2023 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

In recent years, Unmanned Aerial Vehicles (UAVs) have found extensive applications in various fields, including military, industry, agriculture, and tasks like aerial photography and reconnaissance [1–3]. However, it is worth noting that these applications are primarily designed for the outdoors and rely on robust GPS signals for accurate positioning. In cases where GPS signals are unavailable or weak, the precision of UAV positioning is significantly impacted. Presently, there is a growing need for indoor UAV technology, particularly for inspection purposes, and this area is closely tied to control optimization and precise path tracking.

Ultra-Wideband (UWB) technology has garnered significant attention due to its high precision in indoor positioning. UWB systems leverage short-duration pulses of radio waves spread across a wide spectrum, enabling accurate distance measurements using time-of-flight and trilateration techniques. Aiello and Shalom [4] delved into the principles of UWB and its significance in achieving precise and real-time positioning. Heimovirta, Salanterä, and Röning [5], discuss the practical implementation of a real-time indoor localization system based on UWB technology. This work focuses on the utilization of UWB technology for indoor positioning, providing insights into its application in real-world scenarios.

Vicon positioning systems are based on motion capture technology, utilizing cameras and markers to accurately track the movement and position of objects or individuals in

a controlled environment. Rhea [6] assessed the accuracy and precision of Vicon motion capture systems for tracking movements in three dimensions (3D). This work focused on evaluating the reliability of Vicon systems in capturing 3D motion data. Gentil, de la Rouviere, and Elton [7] discuss Vicon as a computer vision solution for real-time 3D motion capture. Their work highlights the capabilities of Vicon technology in capturing and analyzing 3D motion data in real-time, particularly in the context of rehabilitation and neuro-engineering. Benn and Martin [8] provide insights into the utilization of Vicon systems for capturing and analyzing motion data in the context of sports and healthcare.

In this paper, an accurate position estimation was made by combining the IMU and the raw distance data with the help of the Extended Kalman Filter (EKF). Initially, position estimation was obtained using the RLS method employed via a trilateration algorithm. This solution was used as a starting point for RLS. Afterward, this position estimation was fused with the acceleration data. As a result, the estimated position obtained with the designed EKF was compared with the Vicon indoor positioning system, and the results are presented both graphically and in tabular form.

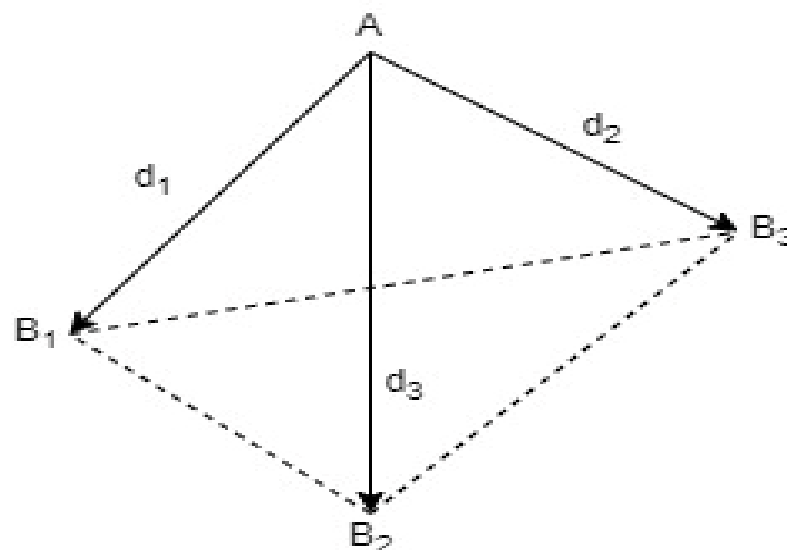
## 2. Position Estimation Algorithm

The position estimation algorithm is based on a geometric approach and EKF algorithms. After these algorithms were designed in a simulation environment, they were integrated into an ROS and tested with real sensor data, and the results were observed.

### 2.1. Geometric Approach

Herein, a geometric approach is put forward on the basis of research. As shown in Figure 1 below, three reference points are given,  $B_1(x_1, y_1, z_1)$ ,  $B_2(x_2, y_2, z_2)$ , and  $B_3(x_3, y_3, z_3)$ , and interval measurements,  $d_1$ ,  $d_2$ , and  $d_3$ , up to point A are given. The determination of the coordinates of point A is carried out by solving a system of quadratic equations.

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



**Figure 1.** Reference points and interval measurements.

The system of equations given here can be expressed as follows:

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

In addition, this expression can be shown in matrix form, as presented below:

$$\begin{bmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ 1 & -2x_3 & -2y_3 & -2z_3 \end{bmatrix} \begin{bmatrix} x^2 + y^2 + z^2 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ d_3^2 - x_3^2 - y_3^2 - z_3^2 \end{bmatrix}\quad (3)$$

This matrix form can generally be expressed as follows:

$$\begin{aligned}A_0 \cdot x &= b_0 & x &\in E \\ E &= \left\{ (x_0, x_1, x_2, x_3)^T \in \mathbb{R}^4 \mid x_0 = x_1^2 + x_2^2 + x_3^2 \right\}\end{aligned}\quad (4)$$

When examining the solution set, there are generally two different approaches. These are divided into three-reference-point-based and more-than-three-reference-point-based solutions. The details of these solution sets are elaborated in [9,10]. In brief, for the first scenario, the solution space varies depending on whether three reference points are in the same line. The second scenario involves having more than three reference points, which necessitates the use of the recursive least squares method for the solution set.

Distance data are used together with the recursive least square algorithm to aid in calculating the position. In the next section, the details of a more accurate position estimation with the help of EKF will be explained.

## 2.2. Sensor Fusion Algorithm

The system's state vector was set as

$$X = \begin{bmatrix} p(w) \\ v(w) \\ a(w) \end{bmatrix}\quad (5)$$

The vector  $p(w)$  denotes an object's position in the world coordinate system, specifying its coordinates on the x, y, and z axes as  $[px, py, pz]$ . The object's velocity along these axes in the world coordinate system is represented as  $v(w) = [vx, vy, vz]$ , while its accelerometer data in the world coordinate system are denoted as  $a(w) = [ax, ay, az]$ .

To elaborate further, the time interval between measurements is defined as  $\Delta t$ , and  $(\Delta t)w(k)$  denotes the process noise related to acceleration. Specifically,  $\frac{\Delta t^2}{2}w(k)$  represents the process noise affecting velocity, and  $\frac{\Delta t^3}{6}w(k)$  indicates the process noise that influences position.

Equations of motion in the  $(k + 1)$  time interval can be expressed as follows:

$$\begin{aligned}p(k + 1) &= p(k) + v(k)(\Delta t) + a(k)\frac{\Delta t^2}{2} + \frac{\Delta t^3}{6}w(k) \\ v(k + 1) &= v(k) + a(k)(\Delta t) + \frac{\Delta t^2}{2}w(k) \\ a(k + 1) &= a(k) + (\Delta t)w(k)\end{aligned}\quad (6)$$

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

$$x(k + 1) = Ax(k) + Gw(k)\quad (7)$$

The matrices  $A$  and  $G$  represent the transition matrix and noise process matrix, respectively. The process noise vector and  $Q$  covariance can be defined as follows:

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

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \Delta t^2/2 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 1 & 0 & \Delta t^2/2 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \Delta t^2/2 \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} \frac{\Delta t^3}{6} & 0 & 0 \\ 0 & \frac{\Delta t^3}{6} & 0 \\ 0 & 0 & \frac{\Delta t^3}{6} \\ \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \quad (8)$$

The observation vector  $z(k)$  contains the distance values used in the geometric approach and an additional noise vector:

$$z(k) = \begin{bmatrix} d_1(k) + n_1(k) \\ d_2(k) + n_2(k) \\ d_3(k) + n_3(k) \\ d_4(k) + n_4(k) \end{bmatrix} = H(k)x(k) + n(k) \quad (9)$$

The observation matrix is represented by  $H(k)$ , and the vector of noise, with a mean of zero and a covariance matrix, is represented by  $n(k)$ .

$$R = \text{diag} \left( \begin{bmatrix} \sigma_{r1}^2 & \sigma_{r2}^2 & \sigma_{r3}^2 & \sigma_{r4}^2 \end{bmatrix} \right) \quad (10)$$

Here,  $r_1, r_2, r_3$ , and  $r_4$  are the distances between each of the values of the Marvelmind sensor. The  $H(k)$  observation matrix is a Jacobian matrix and is calculated as follows:

$$H(k) = \begin{bmatrix} \frac{\delta d_1(k)}{\delta p_x(k)} & \frac{\delta d_1(k)}{\delta p_y(k)} & \frac{\delta d_1(k)}{\delta p_z(k)} \\ \frac{\delta d_2(k)}{\delta p_x(k)} & \frac{\delta d_2(k)}{\delta p_y(k)} & \frac{\delta d_2(k)}{\delta p_z(k)} \\ \frac{\delta d_3(k)}{\delta p_x(k)} & \frac{\delta d_3(k)}{\delta p_y(k)} & \frac{\delta d_3(k)}{\delta p_z(k)} \\ \frac{\delta d_4(k)}{\delta p_x(k)} & \frac{\delta d_4(k)}{\delta p_y(k)} & \frac{\delta d_4(k)}{\delta p_z(k)} \end{bmatrix} \quad (11)$$

The equations described in the structure of the applied EKF are utilized. The initialization stage of this filter 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.

$$\begin{aligned} \bar{x}_k &= f(\hat{x}_k^{-1}, u_k) \\ \bar{P}_k &= A_k^{-1} \hat{P}_k^{-1} A_k^{-1T} + Q_k^{-1} \\ K_k &= \bar{P}_k H_k^T [H_k \bar{P}_k H_k^T + R_k]^{-1} \\ \hat{x}_k &= \bar{x}_k + K_k(z_k - h(\bar{x}_k)) \\ \hat{P}_k &= (I - K_k H_k) \bar{P}_k \end{aligned} \quad (12)$$

In the following section, the integration of this geometric approach with EKF algorithms into the ROS platform is described, and its implementation with a real dataset is discussed. The obtained results are presented in detail.

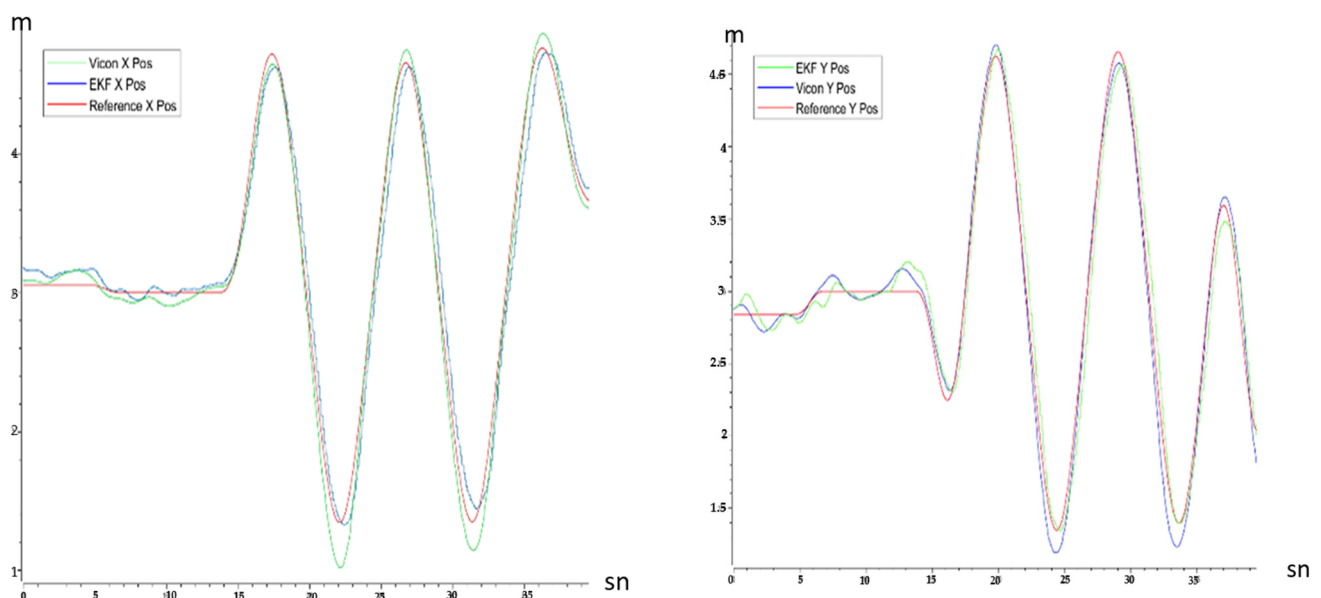
### 3. Implementation and Simulation System Results

After the geometric approach and EKF structure were designed, they were transferred to the simulation environment using C++. The code structure was integrated into the PX4 code, making it compatible with the ROS environment. The acceleration data from UWB sensors are received at approximately 40 Hz, while the raw distance data arrive at around 80 Hz. To address this discrepancy, time synchronization was achieved between the acceleration and distance data. The acceleration data were upsampled to 80 Hz using interpolation. Additionally, since Vicon data are received at a rate of 20 Hz, it was necessary to perform time synchronization with the position data obtained from the EKF results. Therefore, the Vicon data were upsampled to 80 Hz using interpolation to ensure proper time synchronization.

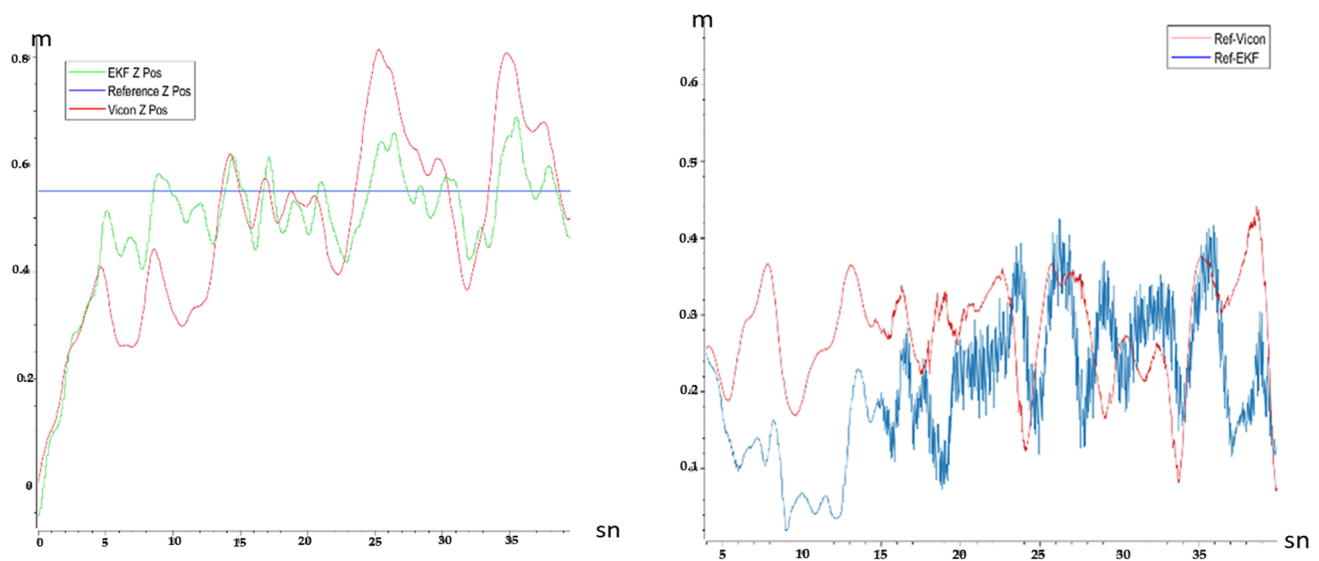
When looking at Figures 2 and 3, it is easily noticeable that the X, Y, and Z positions obtained from the EKF are closer to the reference results than those obtained from the Vicon system. Additionally, Figure 3 depicts the differences between the EKF and Vicon in relation to their respective reference values. Upon examining the position error values in Table 1, it is evident that the average difference between the EKF and the reference is 0.205 m, while for the Vicon, this difference is 0.255 m. Moreover, when looking at the minimum and maximum errors, it is evident that the values obtained from the EKF are lower than those obtained from the Vicon system. In summary, through a numerical comparison of the results, it is apparent that the designed EKF algorithm provides better results compared to the Vicon positioning system.

**Table 1.** Position error values.

|           | Minimum Error (m) | Mean Error (m) | Maximum Error (m) |
|-----------|-------------------|----------------|-------------------|
| Ref-EKF   | 0.019             | 0.205          | 0.424             |
| Ref-Vicon | 0.07              | 0.255          | 0.441             |



**Figure 2.** X and Y positions (Vicon-EKF-Reference).



**Figure 3.** Z position and Vicon and EKF differences from reference.

#### 4. Conclusions

In this study, position estimation was carried out by fusing raw distance data with the IMU using sensor fusion algorithms, specifically via the EKF. The fundamental structure of the designed algorithm incorporates a geometric approach. Firstly, a position calculation was derived from the geometric approach, and then the accuracy of this position was enhanced using accelerometer data and the EKF algorithm. The designed algorithm was transformed for incorporation into the C++ environment and integrated into an ROS. Real sensor data is used during the testing of the algorithm. Sensor data were collected using the ROS platform, and the algorithm was executed within the ROS to observe the obtained results. According to the results obtained, the designed EKF structure yielded more successful outcomes compared to the Vicon position system. These results are supported both graphically and by numerical tables.

**Author Contributions:** Conceptualization, T.B. and F.Ç.; 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.Ç.; visualization, T.B.; supervision, F.Ç. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially financially 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 article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

#### References

1. Valavanis, K.P.; Vachtsevanos, G.J. *UAV Applications: Introduction*; Springer: Amsterdam, The Netherlands, 2015.
2. Samad, T.; Bay, J.S.; Godbole, D. Network-centric systems for military operations in urban Terrain: The role of UAVs. *Proc. IEEE* **2007**, *95*, 92–107.
3. Li, Z.; Liu, Y.; Walker, R.; Hayward, R.; Zhang, J. Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved Hough transform. *Mach. Vis. Appl.* **2010**, *21*, 677–686.
4. Aiello, R.; Bar-Shalom, Y. Ultra-Wideband Technology for Precise Real-Time Location Systems. *Proc. IEEE* **2000**, *88*, 62–85.
5. Heimovirta, E.; Salantera, S.; Röning, J. Real-Time Indoor Localization Using Ultra-Wideband Technology. In Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, London, UK, 2–5 December 2014; pp. 265–274.

6. Rhea, C.K. Accuracy and Precision of Vicon Motion Capture Systems for Tracking in 3D. *Gait Posture* **2004**, *20*, 332–335.
7. Gentil, C.; de la Rouviere, S.; Elton, C. Vicon: A Computer Vision Solution for Reliable Real-Time 3D Motion Capture. *J. NeuroEng. Rehabil.* **2009**, *6*, 11.
8. Benn, A.; Martin, A. Vicon Revue: Motion Capture Technology for Rehabilitation and Sports Medicine. *Biomech. Med. Swim.* **2004**, *IX*, 122–126.
9. 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.
10. Bodrumlu, T.; Caliskan, F. Indoor Position Estimation Using Ultrasonic Beacon Sensors and Extended Kalman Filter. *Eng. Proc.* **2022**, *27*, 16. [[CrossRef](#)]

**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.