VNUHCM Journal of Science and Technology Development 2026, 29(2):4005-4017

Open Access Full Text Article

Research Article

Research and Design of an Enhanced Localization and Mapping Algorithm for an Indoor Mobile Robot based on the Ultra-Wideband Technique

Trung Kien Nguyen, Hai Yen Tran, Hoang Quan Vo, Duc Thien Tran

Use your smartphone to scan this QR code and download this article

ABSTRACT

This study proposes an enhanced localization and mapping algorithm for indoor mobile robots based on Ultra-Wideband (UWB) technology. By integrating UWB with odometry and LiDAR sensors through an Extended Kalman Filter (EKF), the developed approach addresses the limitations of conventional Simultaneous Localization and Mapping (SLAM) techniques in warehouse environments. Whereas conventional SLAM methods rely on relative positioning, the proposed algorithm leverages UWB-based absolute measurements to reduce uncertainty and drift in repetitive or low-feature workspaces. First, the linear velocity and angular velocity of the differential robot are determined based on forward kinematics and odometry sensor feedback signals. Additionally, the Time Difference of Arrival (TDoA) technique is used to calculate the distance from the robot to the UWB anchors to determine the absolute position of the robot. Then, the Adaptive Monte Carlo Localization (AMCL) method is used to obtain the global positioning of the robot based on LiDAR sensor data combined with the map server. Finally, the EKF filter combines the above information to accurately determined the pose of the robot. To evaluate the effectiveness of the algorithm, a numerical simulation is performed to compare the odometry sensor-based positioning algorithm and the proposed odometry, UWB, and EKF sensor-based positioning algorithm in three cases, with the number of UWB anchors increasing from 3 to 5 in a Python environment. Furthermore, a visual simulation is developed within the ROS-Gazebo framework using a differential-drive mobile robot equipped with multiple plug-in sensors to evaluate the improvement of the proposed algorithm. Simulation results from both simulation platforms confirm that the fusion of UWB, odometry, and LiDAR data using EKF substantially improves localization precision and system stability.

Key words: Localization and Mapping, Ultra-Wideband, mobile robot, Time Difference of Arrival, Adaptive Monte Carlo Localization, Extended Kalman Filter

Ho Chi Minh City University of Technology and Education, Vietnam

Correspondence

Trung Kien Nguyen,
Ho Chi Minh City University of Technology and Education, Vietnam
Email: trungkien29052001@gmail.com

History

- Received: 11-07-2025
- Revised: 27-09-2025
- Accepted: 29-12-2025
- Published Online: 09-05-2026

DOI :

https://doi.org/10.32508/stdj.v29i2.4547

Check for updates

Copyright

© VNUHCM Journal. This is an open- access article distributed under the terms of the Creative Commons Attribution 4.0 International license.

INTRODUCTION

The development of service robots has emerged as a research trend in many countries. Many advanced types of service robot exist, such as home care robots, housework robots, hospital robots, and airport guide robots[1]. A common feature among these robots is that their operating environment is primarily indoors. Broadly, they are automated devices capable of moving independently and performing a range of different tasks. These types of robots are designed to operate in closed spaces, such as homes, offices, factories, shopping malls, hospitals, among other places. They can be used for a wide variety of applications, such as routing, mapping, environmental inspection and monitoring, goods transportation, customer service, and many other tasks. Many studies with different algorithms and methods have aims to design robots in indoor environments. These studies are very diverse, using different types of sensors and serving many different purposes[2]. Fully autonomous mobile robots are required to navigate when performing tasks, wherein the challenge of data collection and robot positioning in the operating environment is of crucial importance in the navigation structure for mobile robots[2,3]. Previous research has employed many methods to collect data and, together with algorithms to measure and calculate the position of the robot, studies (e.g., [4]) have proposed the use of cameras with image processing methods to determine the current position of the robot while avoiding obstacles, thereby planning the path to the destination for a mobile robot. In addition to the above method, another study[5] proposed the use of laser sensors through particle filters and Monte Carlo algorithms to update maps to determine the position of the robot. Additionally the Dead Reckoning method has been proposed to calculate the current position of a robot based on its previous position and the current velocity and rotation

Cite this article : Nguyen T K, Tran H Y, Vo H Q, Tran D T. **Research and Design of an Enhanced Localization and Mapping Algorithm for an Indoor Mobile Robot based on the Ultra-Wideband Technique**. *VNUHCM J. Sci. Technol. Dev.* 2026; 29(2):4005-4017.

direction of the robot[6–8], which can be obtained using an inertial measurement unit (IMU) sensor and encoder sensors. However, using image processing algorithms requires high-performance computational configurations, which can be difficult if there is a limit on computational resources. The Dead Reckoning algorithm also leaves accumulated errors during operation due to wheel slippage, and odometry data can only be calculated if one sets the initial value of the robot manually, i.e., we can use LiDAR sensors with a scan matching algorithm to determine the initial position of the robot[9]; however, in cases where the environment does not have significant features, such as in a large warehouse, this algorithm will not work. To overcome this challenge, we need an absolute positioning system for the robot. Global Positioning System (GPS) is a powerful positioning system, but it is only effective outdoors. In closed indoor environments, the satellite signal is suboptimal, leading to inaccurate positioning. For indoor robots, positioning also faces numerous challenges due to the heterogeneous nature and presence of obstacles that cause variations in signal strength and noise levels, rendering high-precision localization a difficult task[10]. To overcome these challenges, various technologies such as RFID, WiFi, Bluetooth, and Zigbee have been widely used. However, Ultra-Wideband (UWB) technology has emerged as a powerful positioning system with the ability to provide greatly improved timing accuracy relative to other radio systems[11]. UWB offers several notable advantages, including its high channel capacity due to its exceptionally wide bandwidth, which allows for low transmission power[12]. Additionally, its very short pulse duration (typically on the order of nano- or picoseconds) helps mitigate multipath fading. Its resilience to multipath effects and its excellent temporal resolution also make UWB an ideal choice for high-precision localization, facilitating accurate distance measurements through techniques including the time of arrival (ToA), time difference of arrival (TDoA), and two-way time of arrival (TW-ToA) methods[13].

Building on developments in UWB positioning technology, several studies have focused on improving the accuracy of indoor localization systems. In 2017, Marquez and colleagues conducted a study on the fusion of data from UWB and IMU sensors using the extended Kalman method to improve the accuracy of indoor positioning systems[14]. The results of the study showed that the integration of indoor navigation systems (INS) and UWB provides a wider range and better response than systems using single sensors. In 2022, Hua and colleagues studied the use of the Extended Kalman Filter (EKF) to fuse data between 2D LiDAR sensors, cameras, IMUs, and encoders, thereby improving the accuracy of positioning and map building[15]. The effectiveness of the method was verified through simulation results in the ROS environment. The combination of sensors to improve positioning accuracy was furthered by the Phueakthong group in 2023[9]. This experimental research was verified to improve accuracy and reduce drift data in the positioning of mobile robots when combining IMU, odometry, and UWB sensors with an EKF. In 2024, He and colleagues proposed a method for locating the position of indoor mobile robots using UWB and odometry and IMU sensors combined with an EKF[16]. The research was experimentally proven to have high accuracy in locating the position of indoor mobile robots. In 2024, the Ranjan group proposed asymptotic methods through the use of filtering techniques to improve accuracy and reduce noise for indoor positioning systems using UWB. Their approaches include a combination of a low-pass filter with a median filter, a low-pass filter with a Kalman filter, and a low-pass filter with an EKF. The results obtained show that the combination of a low-pass filter with an EKF is the most effective among the three approaches[17]. The above studies have proposed many positioning methods for indoor robots based on UWB technology. However, these methods still have certain limitations, and each sensor type that collects robot position information will have a certain sampling frequency. If the robot operates beyond this sampling threshold, the positioning information will gradually become distorted, unstable, and of reduced accuracy. Consequently, combining different types of sensors is necessary to provide absolute positioning. This paper proposes an approach integrating UWB with odometry and LiDAR sensors through an EKF to obtain stable and accurate position information for indoor mobile robots.

The remainder of this study is structured as follows: Section 2 provides the system description, including two parts: the kinematics of the differential robot and the proposed indoor positioning system. Section 3, discusses the proposed algorithms, including four parts: the EKF algorithm, Adaptive Monte Carlo (AMCL) algorithm, Time Difference of Arrival (TdoA) technique, and the proposed algorithm. Section 4 (Simulations and Results) evaluates the performance of the proposed method using many cases are presented. Finally, Section 5 presents the conclusion of this study.

METHODS AND SYSTEM DESCRIPTION

Kinematics of the differential robot

A differential robot belongs to a typical class of nonholonomic wheeled mobile robots, characterized by two independently actuated wheels aligned on a common axis and a passive rolling wheel for balance and support. The motion of the robot is controlled by varying the velocities of the wheels: identical wheel speeds result in straight-line motion, while different speeds cause rotational motion around a vertical axis. The kinematics modeling of a differential robot is shown in Figure 1.

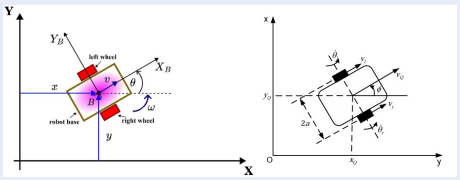


Figure 1: Kinematics modeling of a differential robot

The differential steering mobile robot system described in Figure 1 consists of a robot base and two driving wheels, where $X-Y$ is the fixed coordinate system, X_B-Y_B is the axis system mounted on the robot, (x,y) is the coordinate of point B and the center of gravity of the robot, θ is the angle formed by X_B and the X-axis, v_l, v_r are the linear velocities of the left and right wheels, and ω_l, ω_r are the angular velocities of the left and right wheels.

By discretizing the continuous-time kinematics of the mobile robot, the following discrete model is obtained:

$$\begin{cases} x_{k+1} = x_k + v_k \triangle t \cos(\theta_k) \\ y_{k+1} = y_k + v_k \triangle t \sin(\theta_k) \\ \theta_{k+1} = \theta_k + w_k \triangle t \end{cases} \quad (1)$$

where k is the discrete time considered, $x_k - y_k$ are the coordinates of the robot at time k, v_k is the linear velocity from k to $k+1$, assuming the linear velocity v_k is constant, θ_k is the angle formed by X_B and the X-axis at time k, and $\triangle t$ is the time interval from k to $k+1$.

During the measurement of v_k, θ_k, process noise appears (18). When this process noise is added to the system, the discrete model (1) is rewritten as follows:

$$\begin{cases} x_{k+1} = x_k + \widehat{v}_k \triangle t \cos(\theta_k) + n_{v,k} \cos(\theta_k) \\ y_{k+1} = y_k + \widehat{v}_k \triangle t \sin(\theta_k) + n_{v,k} \sin(\theta_k) \\ \theta_{k+1} = \theta_k + \widehat{w}_k \triangle t + n_{w,k} \end{cases} \quad (2)$$

where $n_{v,k}$ are measurement noises.

Overview of the indoor positioning system

The differential mobile robot model was set with a contour size of 350 × 300 × 270 mm (length × width × height), the distance between two wheels is 235 mm, and the radius of the vehicle is 70 mm. Additionally, a fixed UWB anchor is integrated on the robot together with a LiDAR sensor and odometry sensors, as shown in Figure 2. The fixed UWB anchor on the robot performs high-bandwidth transmission/reception to calculate the distance from the robot to the fixed UWB anchors in the indoor working environment. The LiDAR sensor combines with a known environment to collect information about the surrounding environment and obstacles and the odometry sensors collect information relating to the angle and angular velocity of the robot.

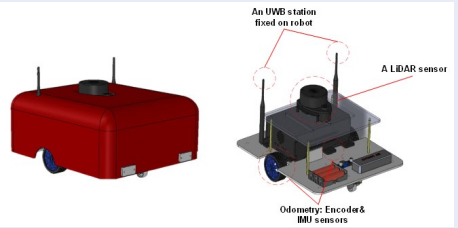


Figure 2: Differential robot integrated an UWB station, with LiDAR and odometry sensors.

Figure 3 shows an indoor working environment for the differential mobile robot model used in this paper with overall dimensions 3.3m x 2.4m and four fixed obstacles with sizes of 0.75m x 0.3m.

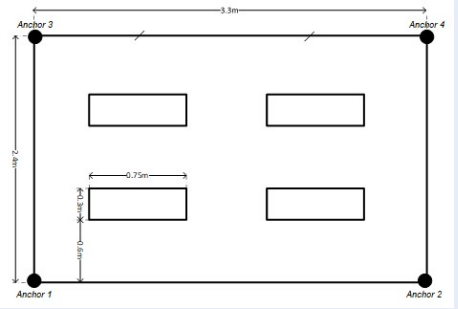


Figure 3: The proposed indoor positioning system.

The positions of the UWB stations are fixed in the known environment at (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3) and the position of the differential robot at the center of the environment is (x, y), as shown in Figure 4.

The distance from the robot to the anchors is d_0, d_1, d_2, d_3.

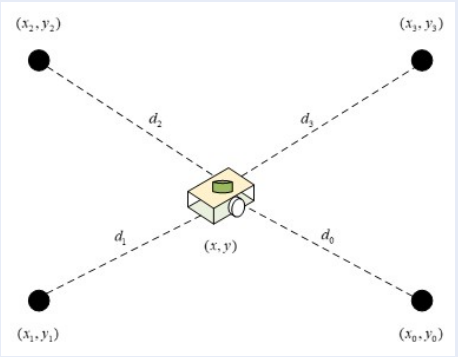


Figure 4: The initial reference positions of UWB anchors and differential robots.

The proposed indoor positioning system used to demonstrate the effectiveness of the main proposed algorithm of the paper is presented in Section 3 below.

PROPOSED ALGORITHMS

Extended Kalman Filter (EKF) algorithm

Since real-world systems are almost always nonlinear, the state distribution no longer follows a Gaussian distribution, which causes the Kalman algorithm to fail to converge properly. To address this issue, we apply the EKF (19), a nonlinear state estimation method. The implementation of the EKF involves prediction and update phases:

In the **prediction phase** at time k, the estimated state and estimated covariance are calculated as follows:

$$S_{k|k-1} = f\left(S_{k-1|k-1}, u_{k-1}, 0\right) \quad (3)$$

$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^T + Q_{k-1} \quad (4)$$

where s_k, u_k and n_k are the state, control input, and process disturbance, respectively, of the system at time k, $f(.)$ is a known function, S_{k-1-1} and P_{k-1-1} is the estimated state in the update phase at the time $k-1$, A_{k-1} is a Jacobian matrix providing the partial derivative of $f(S_{k-1-1}, u_{k-1}, 0)$ with respect to k through linearization as follows:

$$A_{k-1} = \frac{\delta f}{\delta S}(S_{k-1}, u_{k-1}, 0) \quad (5)$$

and Q_{k-1} is the covariance matrix of the process noise n_k as follows:

$$Q_{k-1} = W_{k-1}Q_{n_k}W_{k-1}^T \quad (6)$$

where Q_{n_k} is the covariance matrix of the process noise n_k, W_{k-1} is a Jacobian matrix providing the partial derivative of $f(S_{k-1-1}, u_{k-1}, 0)$ with respect to n_k through linearization as follows:

$$W_{k-1} = \frac{\delta f}{\delta n_k}(S_{k-1-1}, u_{k-1}, 0) \quad (7)$$

In the **update phase**, the Kalman constant K_k is defined as follows:

$$K_k = P_{k-1}H_k^T S^{-1} \quad (8)$$

where the measurement error covariance is confirmed as follows:

$$S_k = H_{k-1}P_{k-1}H_{k-1}^T + R_k \quad (9)$$

where R_k is the measurement covariance matrix and H_{k-1} is the Jacobian matrix of the partial derivatives between $h(.)$ and s_k:

$$H_{k-1} = \frac{\delta h(s_{k-1}, 0)}{\delta s_{k-1}} \quad (10)$$

The estimated state s_k and the estimated covariance P_k are determined as follows:

$$s_k = s_{k+1} + K_k\tilde{y}_k \quad (11)$$

$$P_k = (I - K_kH_k)P_{k-1} \quad (12)$$

where $\tilde{y}_k$ is the measurable error determined by the formula:

$$\tilde{y}_k = z_k - h(s_{k-1}, 0) \quad (13)$$

After completing formula [12] in the update phase, the next prediction process takes place.

Adaptive Monte Carlo (AMCL) algorithm

AMCL is a method for estimating position in a predefined map, and is essentially an expression of the Monte Carlo Localization (MCL) algorithm. MCL, also known as Particle Filter (PF), is used to determine the position of a robot on a predefined map. The filter receives map information, estimates the position and direction of the robot from the algorithm as it moves, and updates the surrounding environment. The AMCL operation process operates as follows:

Step 1: Initialization. The particles are initialized to random values in the possible position space of the robot. The set of particles is represented as follows:

$$S = \left\{ \left\langle s^{[i]}, w^{[i]} \right\rangle = 1, \ldots, M \right\} \quad (14)$$

where, $s^{[i]}$ denotes a hypothesized state of the system at the i-th particle, $w^{[i]}$ is the corresponding weight of each particle, and M is the total number of particles used in the simulation or state estimation process. Samples representing the posterior are as follows:

$$p(x) = \sum_{i=1}^{M} w_i\delta_{s^{[i]}}(x) \quad (15)$$

where $\delta_{s^{[i]}}(x)$ is a delta function and w_i is the weight of the i-th particle.

Step 2: Prediction. Particles are moved based on data from the sensors; linear velocity and rotational velocity data are fed into the kinematic and dynamic motion of the robot used to predict movement, which can be calculated as follows:

$$\hat{x}_t^m = f\left(x_{t-1}^m, u_t\right) \quad (16)$$

where $\hat{x}_t^m$ is the predicted position of the m-th particle at time t and $f\left(x_{t-1}^m, u_t\right)$ is the model function that predicts the directional velocity of the robot.

Step 3: Update weights. The particles are updated based on sensor data such as LiDAR or IMU. Specifically, each particle is evaluated by calculating the probability of the observed sensor data compared to the prediction of that particle. Particles with high probabilities corresponding to actual sensor data will have a higher weight, whereas particles with low probability will be discarded or have their weight reduced. The formula for calculating weights in AMCL based on the sensor data collected by the LiDAR sensor is as follows:

$$\widehat{w}_t^m = \frac{f\left(x_t^{[m]}\right)}{g\left(x_t^{[m]}\right)} \quad (17)$$

where $\widehat{w}_t^m$ is the updated weight of the ith particle, $f\left(x_t^{[m]}\right)$ is a cost function, and $g\left(x_t^{[m]}\right)$ is a proposed function.

Step 4: Resampling: After updating, some or all of the particles are resampled based on their weights. Particles with higher weights are selected with a higher probability for reconstruction, whereas particles with lower weights are discarded. This process helps to cluster the particles in the highest probability locations, thereby creating a new particle set. The cumulative weight of each particle is calculated by summing up the weights from the first particle to the m-th particle as follows:

$$W^m = \sum_{j=1}^{m} w^m \quad (18)$$

where W^m is the cumulative weight of particle m. This process creates an array of cumulative weights $\left\{W^{(m)}\right\}_{m=1}^{M}$; $W^M = W_{..}$

Step 5: Repeat. The prediction, update, and sampling process is repeated cyclically, with each iteration improving the estimation of robot position based on new sensor data. The AMCL loop stop condition is met when one of the following conditions is satisfied: max iterations, min weight change, minimum weight threshold, or accuracy of estimated location.

Time Difference of Arrival (TDoA) technique

TDoA is a positioning technique widely used in wireless positioning systems, especially systems using radio waves such as UWB or indoor positioning systems. Based on difference in signal arrival time between anchors, the system will calculate the relative distance between the transmitter and each pair of receiving anchors in the proposed indoor positioning system as shown in Figure 5.

The proposed system includes a robot with coordinates (x, y, z) and four UWB anchors with coordinates (x_0, y_0, z_0), (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3). The distance from each UWB anchor to the robot is

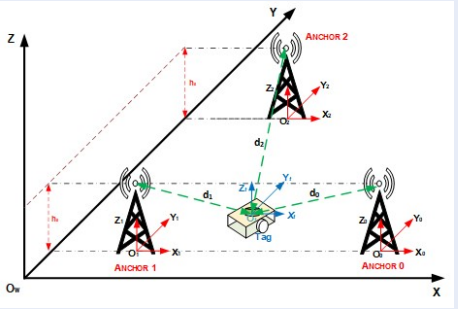


Figure 5: Time Difference of Arrival (TDoA) technique for a system including robot and four UWB stations.

calculated using the ToA (20) technique, respectively, as d_0, d_1, d_2, d_3. The distance from the robot to the remaining anchors in the system, assuming that $z_{i,(i=0,1,2,3)}$ is equal, is determined as follows:

$$AX = B \quad (19)$$

$$where\ A = \begin{bmatrix} x_0 - x_1 & y_0 - y_1 \\ x_0 - x_2 & y_0 - y_2 \\ x_0 - x_3 & y_0 - y_2 \end{bmatrix},\ X = \begin{bmatrix} x & y \end{bmatrix}^T,$$

$$B = \begin{bmatrix} \frac{1}{2}\left(d_1^2 - d_0^2 + x_0^2 - x_1^2 + y_0^2 - y_1^2\right) \\ \frac{1}{2}\left(d_2^2 - d_0^2 + x_0^2 - x_2^2 + y_0^2 - y_2^2\right) \\ \frac{1}{2}\left(d_3^2 - d_0^2 + x_0^2 - x_3^2 + y_0^2 - y_3^2\right) \end{bmatrix}$$

The position of the robot is then calculated using the following formula:

$$X = \left(A^T \quad A\right)^{-1} A^T B \quad (20)$$

Proposed positioning algorithm based on odometry, LiDAR, and UWB through EKF

The structure of the proposed localization algorithm based on odometry, LiDAR, and UWB systems through EKF is shown in Figure 6. This algorithm is built on the forward kinematics of the robot, the TDoA technique, the AMCL method, and the EKF. Forward robot kinematics calculates the linear velocity and angular velocity based on the feedback signal from odometry sensors. The TDoA technique calculates the distance from the robot to the UWB transceiver anchors, and therefore determines the absolute position of the robot. The AMCL method calculates the pose of the robot based on the signal from the LiDAR sensor, a known environment, and the pose of the robot in its previous state.

The EKF is used to fuse information from the odometry and LiDAR sensors with the UWB system as shown in Figure 7. The proposed method consists of two main processes, namely the predicting process and the updating process:

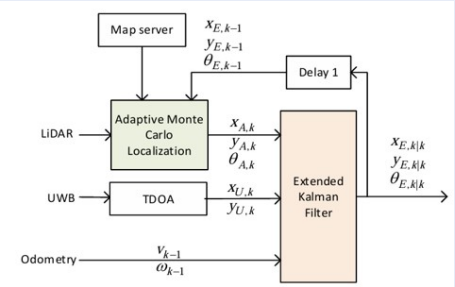


Figure 6: Schematic showing the positioning algorithm based on the IMU sensor, encoder, Ultra Wideband system and extended Kalman filter.

Step 1: During the predicting process, the pose of the robot at time k is estimated based on the pose of the robot, linear velocity, and angular velocity at time $k-1$. In this paper, the discrete model of the differential robot used in the design of the proposed positioning algorithm is defined as follows:

$$f(s_k,\ u_k,\ n_k)=\begin{bmatrix} s_{1k}+u_{1k}\triangle t\cos(s_{3k})+n_{1k}\cos(s_{3k}) \\ s_{2k}+u_{1k}\triangle t\cos(s_{3k})+n_{1k}\sin(s_{3k}) \\ s_{3k}+u_{2k}\triangle t+n_{2k} \end{bmatrix} (21)$$

where $s_k=\begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^T$, $u_k=\begin{bmatrix} v_k & \omega_k \end{bmatrix}^T$, $n_k=\begin{bmatrix} n_{v,k} & n_{w,k} \end{bmatrix}^T$ are the system state, system control input, and process disturbance component, respectively.

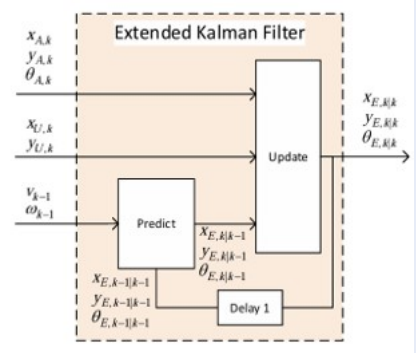


Figure 7: Schematic showing the extended Kalman filterused to combine signals from the IMU, encoder, and UWB sensors.

Using signals derived from UWB transceiver anchors, we can calculate the position of the robot. The position vector obtained is determined as follows:

$$z_k=h(s_k,\ n_{uwb,k})\in R^{2\times 1}\ (22)$$

where $h(s_k,\ n_{uwb,k})=\begin{bmatrix} s_{1k}+n_{uwb1,k} \\ s_{2k}+n_{uwb2,k} \end{bmatrix}\in R^{2\times 1}$ $with$ $n_{uwb,k(i=1,2)}$ has the covariance matrix:

$$R_k=diag\left(\begin{bmatrix} r_{uwb1,k}^2,\ r_{uwb2,k}^2 \end{bmatrix}\right)\ (23)$$

where $r_{uwbi,k(i=1,2)}$ is the standard deviation of the position of the robot; this terms is estimated based on the distances from the robot to the UWB anchors.

In the case of a signal from the LiDAR sensor, we obtain the following measurement vector:

$$z_k=h(s_k,\ n_{AMCL,k})\in R^{3\times 1}\ (24)$$

where $h(s_k,\ n_{AMCL,k})=\begin{bmatrix} s_{1k}+n_{AMCL1,k} \\ s_{2k}+n_{AMCL2,k} \\ s_{3k}+n_{AMCL3,k} \end{bmatrix}$, $n_{AMCLi,k(i=1,2,3)}$ has the following covariance matrix:

$$R_k=diag\left(\begin{bmatrix} r_{AMCL1,k}^2,\ r_{AMCL2,k}^2,\ r_{AMCL3,k}^2 \end{bmatrix}\right)\ (25)$$

with $r_{AMCLi,k(i=1,2,3)}$ representing the standard deviation of AMCL method.

Initialization $s_{0|0}=\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, $P_{0|0}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ are the initial state and covariance matrix of the robot, respectively, assuming that the initial position of the robot coincides with the origin and no process disturbance exists in the initial state.

Predicting the process at the time k is performed according to the following formulae:

$$s_k=f\left(s_{k-1|k-1},u_{k-1},0\right)=\begin{bmatrix} s_{1,k-1|k-1}+u_{1,k-1}\triangle t\cos\left(s_{3,k-1|k-1}\right) \\ s_{2,k-1|k-1}+u_{1,k-1}\triangle t\sin\left(s_{3,k-1|k-1}\right) \\ s_{3,k-1|k-1}+u_{2,k-1}\triangle t \end{bmatrix} (26)$$

$$P_{k-1}=A_{k-1}P_{k-1-1}A_{k-1}^T+Q_{k-1}\ (27)$$

where A is a Jacobian matrix and determines the partial derivative of f with respect to s through linearization as follows:

$$A_{k-1}=\frac{\delta f}{\delta s}(s_{k-1},\ u_{k-1},\ 0)=\begin{bmatrix} 1 & 0 & -u_{1k-1}\triangle t\sin\left(s_{3,k-1}\right) \\ 0 & 1 & u_{1k-1}\triangle t\cos\left(s_{3,k-1}\right) \\ 0 & 0 & 1 \end{bmatrix} (28)$$

and Q_{k-1} is the covariance matrix of the noise process n_k, which is defined as follows:

$$Q_{k-1}=\begin{bmatrix} \sin^2\left(s_{3,k-1-1}\right)q_{v,k}^2 & & \\ -\sin\left(s_{3,k-1-1}\right)\cos\left(s_{3,k-1-1}\right)q_{v,k}^2 & & \\ 0 & & \\ -\sin\left(s_{3,k-1-1}\right)\cos\left(s_{3,k-1-1}\right)q_{v,k}^2 & 0 & \\ \cos^2\left(s_{3,k-1-1}\right)q_{v,k}^2 & 0 & \\ 0 & q_{v,k}^2 & \end{bmatrix} (29)$$

Step 2: During the updating process, upon receiving a signal from UWB transceiver anchors, the measurement error is defined as follows:

$\tilde{y}_k = z_k - h(s_{k-1}, 0)$ (30)

where $h\left(s_{k|k-1}, 0\right) = \left[s_{1,k} + s_{2,k}\right]^T \in R^{2\times 1}$

The measurement error covariance S_k is defined as follows:

$S_k = H_{k-1} P_{k-1} H_{k-1}^T + R_k$ (31)

where H_{k-1} is a Jacobian matrix providing the partial derivative between z_k and s_k:

$$H_{k-1} = \frac{\delta h_k}{\delta s_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (32)$$

When a signal is received from the LiDAR sensor, the measurement error $\tilde{y}_k$ is determined as follows:

$\tilde{y}_k = z_k - h(s_{k-1}, 0)$ (33)

where $h(s_{k-1}, 0) = \left[s_{1,k}, s_{2,k}, s_{3,k}\right]^T$, the measurement error covariance S_k is defined similarly to (31), and H_{k-1} is a Jacobian matrix denoting the partial derivative between z_k and s_k:

$$H_{k-1} = \frac{\delta h_k}{\delta s_k} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (34)$$

The Kalman coefficient K_k, estimated state s_k, and estimated covariance P_k are determined using formulae (8), (11), and (12), respectively. After executing formula (12) in the update step, the next prediction process takes place.

RESULTS AND DISCUSSION

A numerical simulation in the Python environment

To evaluate the positioning efficiency of the proposed method, a system consisting of a differential robot and UWB stations was implemented using Visual Studio software with the Python programming language. The mobile robot model is equipped with two motors, encoder sensors to determine the linear velocity and rotational velocity of the robot, and a sensor to measure the distance from the robot to the UWB anchors, with a maximum measurement range of 20 m. The positions of the UWB anchors are known. The simulation time is 50 s with a sampling time of 0.1 s. The variance of the process noise and the measurement noise set up during the simulation process are given variances of $Q_{sim} = diag\left(\left[0.4, 3.49 \times 10^{-2}\right]\right)$ *and* $R_{sim} = diag\left([2, 0.349]\right)$.

Three simulation cases are performed to evaluate the effectiveness of the two proposed algorithms:

Case 1: The system includes a mobile robot, and the positions of three UWB anchors are L_1, L_2, L_3 *and* $Q = diag\left([0.25, 0.25, 0.274, 0, 0, 0, 0, 0, 0, 0]\right)$, $R = diag\left([0.25, 0.25]\right)$.

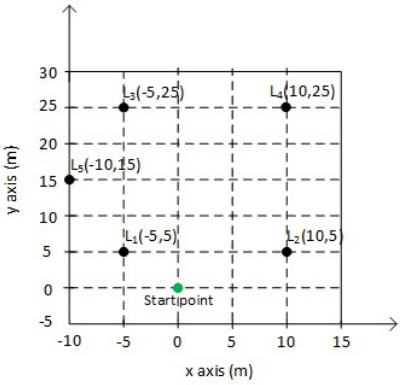


Figure 8: Coordinates of the UWB stations

Case 2: The system includes a mobile robot, and the positions of four UWB anchors are L_1, L_2, L_3, L_4 *and* $Q = diag\left([0.25, 0.25, 0.274, 0, 0, 0, 0, 0, 0, 0]\right)$, $R = diag\left([0.25, 0.25]\right)$.

Case 3: The system includes a mobile robot, and the positions of five UWB anchors are L_1, L_2, L_3, L_4, L_5 with the variance of the process noise and the measurement noise as follows: $Q = diag\left([0.25, 0.25, 0.274, 0, 0, 0, 0, 0, 0, 0]\right)$, $R = diag\left([0.25, 0.25]\right)$.

Two algorithms are presented to simulate and evaluate the above three cases:

Algorithm 1: Odometry sensor-based positioning algorithm.

Algorithm 2: Odometry sensor-based positioning algorithm, UWB system, and EKF.

Figure 9 shows the motion trajectories of the robot in a field with three, four and five UWB anchors. The black line shows the actual trajectory of the robot, the blue line shows the estimated trajectory of the robot when applying algorithm 1, and the red line shows the estimated trajectory of the robot when applying algorithm 2. During the positioning process, to evaluate the effectiveness of both algorithms examined, the standard deviation of the error (Root Mean Square Error, RMSE) between the estimated position and the actual position is determined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}} \quad (35)$$

where $y(i)$, $\hat{y}(i)$ are the measured value and the actual value at the i-th data, respectively, and N is the length of the data. The RMSE is presented in two tables below with respect to the x-axis and y-axis.

Table 1 and Table 2 show the standard deviation of the robot trajectory when applying algorithm 1 and

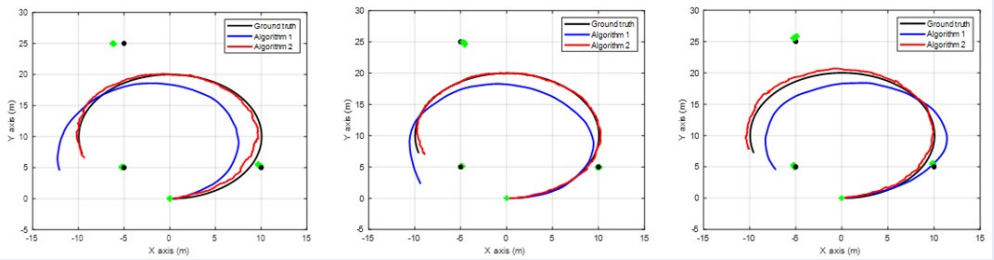


Figure 9: Numerical simulation results for algorithm 1 and algorithm 2 in: a) Case 1: three UWB anchors, b) Case 2: four UWB anchors, and c) Case 3: five UWB anchors.

Table 1: Standard deviation of the error of the robot trajectory along the x-axis in a numerical simulation.

RMSE of robot position along the x-axis			
Case	3 UWB anchors	4 UWB anchors	5 UWB anchors
Algorithm 1	2.2114	3.2689	1.3435
Algorithm 2	1.0970	0.8810	0.3480

Table 2: Standard deviation of the error of the robottrajectory along the y-axis in a numerical simulation

RMSE of robot position along the y-axis			
Case	3 UWB anchors	4 UWB anchors	5 UWB anchors
Algorithm 1	2.2114	3.2689	1.3435
Algorithm 2	1.0970	0.8810	0.3480

algorithm 2, respectively, under environmental conditions of linear velocity of $1m/s$, angular velocity of $0.1rad/s$, sampling time of $0.1s$, UWB scanning frequency of $10Hz$, and Odom of $10Hz$, with a changing number of UWB anchors. The results obtained show that the trajectory of the robot is closer to the actual trajectory and exhibits less slippage when applying algorithm 2 relative to algorithm 1. In addition, when four UWB stations are used, the robot position along the x-axis and y-axis estimated by algorithm 2 is more accurate. When applying algorithm 1, the accuracy of the robot position is not influenced by the change in the number of UWB stations because it considers only data from the odometry. Therefore, applying algorithm 2 is more accurate and effective in determining the position of the robot.

A visual simulation in the ROS-Gazebo environment

An indoor position system is implemented as shown in Figure 10, the dimensions of which are presented in Figure 4. In addition, the structure of the robot proposed is as shown in Figure 2. To ensure that the simulation closely approximates reality, environmental noise, external force impact, and wheel slip phenomena are added with the parameters shown in Table 3.

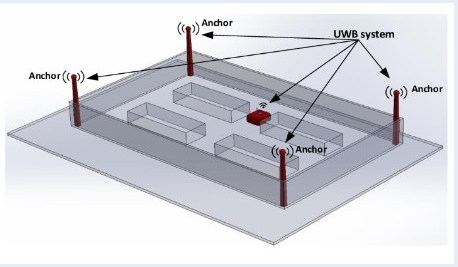


Figure 10: i=Indoor position system for a visual simulation in the ROS-Gazebo environment

The implementation process of the visual simulation in the ROS-Gazebo environments includes the following steps:

Step 1: Designing the CAD differential mobile robot model.

Step 2: Setting up the coordinate system and axes for the CAD differential mobile robot model.

Step 3: Converting the assembly file to a URDF file.

Step 4: Adding sensor plugins and the control plugin.

Step 5: Building environments in Gazebo.

Step 6: Using Hector Slam to build the map.

Step 7: Running simulations to evaluate the effectiveness of the proposed method.

The nodes used in the ROS environment are interconnected as shown in Figure 11.

Table 3: Wheel specifications in URDF

Specifications	Symbols in URDF	Value
Direction of friction	*Fdir*1	$\begin{bmatrix}1 & 0 & 0\end{bmatrix}$
Main friction	*mu*1	0.9
Secondary friction	*mu*2	0.9
Contact stiffness	*kp*	5000000 *N/m*
Contact damping coefficient	*kd*	10*Ns/m*

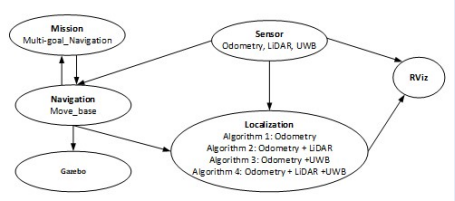


Figure 11: Summary of the nodes used in ROS for the visual simulation

The survey trajectory to evaluate the effectiveness of the proposed method passes through the following coordinate in sequence: $P_1\,(0,0) \rightarrow P_2\,(2.1,0) \rightarrow P_3\,(4.8,0) \rightarrow P_5\,(2.1,1.7) \rightarrow P_8\,(2.1,3.4) \rightarrow P_9\,(4.8,3.4)$, as presented in Figure 12. The visual simulation consists of four algorithms, including: odometry sensors-based, odometry + LiDAR sensors-based, odometry sensors + UWB-based, and odometry sensors + LiDAR + UWB-based positioning. The algorithms are evaluated in two cases as follows:

Case 1: The robot sensors operate normally.

The trajectory error along the x-axis and y-axis of each algorithm relative to the true trajectory in case 1 is shown in Figure 14.

In case 1, when the sensors in all four algorithms are operating normally, the positions of the robot relative to the corresponding positions in the actual trajectory are shown in Figure 13 and Figure 14; this indicates that the results are quite accurate. As shown in Table 4, which presents the trajectory error in the x-axis and y-axis, respectively, combining the algorithms with the UWB absolute positioning system gives smaller error values. Applying algorithm 4 results in the best positioning.

Case 2: The LiDAR signal suddenly stops at t = 11 sec, and the pose of the robot is not fed back. At t = 33 sec, the sensor works normally, and the pose of the robot information is returned.

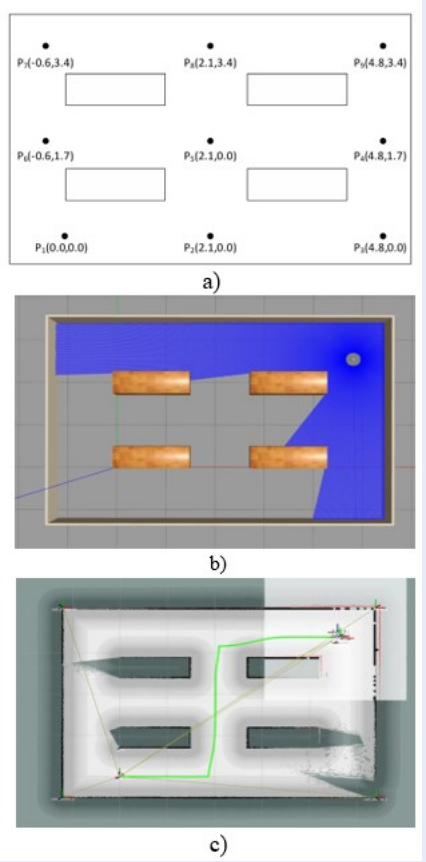


Figure 12: Work map and coordinates of points through which the robot passes in: a) the raw drawing; b) a known environment; c) a map is built using the LiDAR sensor

The trajectory error along the x-axis and y-axis of each algorithm relative to the true trajectory in case 2 is shown in Figure 16.

The LiDAR signal suddenly stopped at 11 s, and the pose of the robot is not fed back. Then, at 33 s, the sensor worked normally, and the pose of the robot information was returned as shown in Figure 15 and Figure 16. At this time, the pose of the robot estimated by

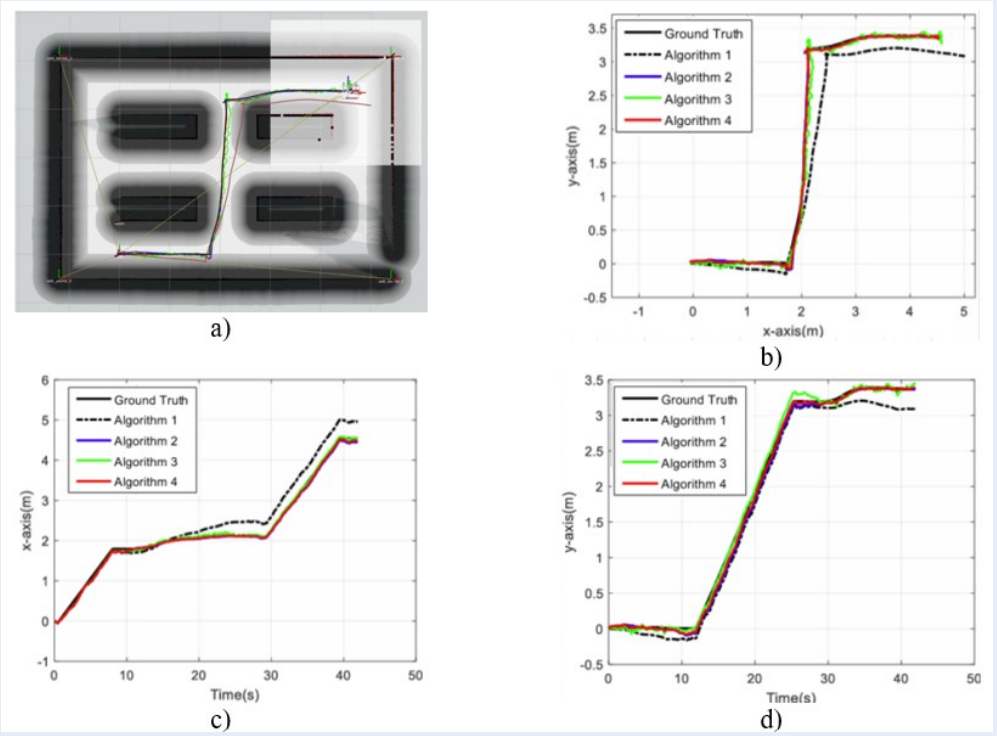


Figure 13: Motion trajectory of the robot in case 1: a) Rviz environment; b) Oxy coordinate system; c) Ox coordinate system; d) Oy coordinate system.

Table 4: Standard deviation of error between the actual trajectory and the trajectory estimated by algorithms in case 1.

RMSE	Along the x-axis	Along the y-axis
Algorithm 1: Odometry-based	0.2754	0.1621
Algorithm 2: Odometry + LiDAR-based	0.0497	0.0715
Algorithm 3: Odometry + UWB-based	0.0457	0.0489
Algorithm 4: Odometry + LiDAR + UWB-based	0.0422	0.0510

algorithm 2 is no longer accurate because algorithm 2 operated based only on the signal from odometry during the laser cut-off time. For algorithms 1 and 3, the sudden cutoff of the laser signal does not affect the pose of the robot estimation result. For algorithm 4, when the laser signal is cut off, because the pose of the robot estimation result during this time is based on the signals from odometry and UWB, the accuracy is still maintained.

Table 5 shows that although there an interruption occurs in the operation of the LiDAR sensor, the trajectory error in the x-axis and y-axis, respectively, only increases significantly for algorithm 2, which is based on odometry + LiDAR sensors. Since algorithms 3 and 4 include combinations of the UWB system to determine the absolute position, they continue to yield very good positional determination and are similar to case 1, in which the sensors have no interruption. As such, it can be seen that the proposed algorithm is effective, stable, and flexible.

CONCLUSIONS

This paper presents an enhanced localization and mapping algorithm for use with indoor mobile robots based on UWB technology. The approach integrates UWB with odometry and LiDAR sensors through an EKF with the objective of addressing limitations of conventional SLAM techniques in warehouse environments. To demonstrate the effectiveness of the proposed algorithm, a numerical simulation in the

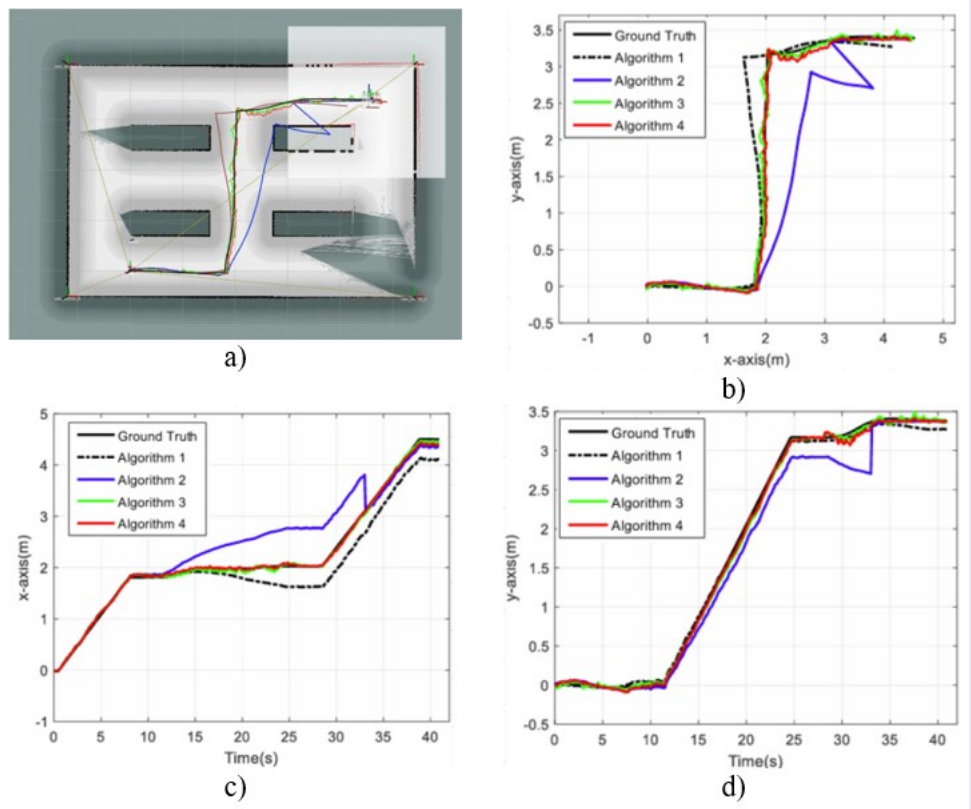


Figure 15: Motion trajectory of the robot in case 2: a) Rviz environment; b) Oxy coordinate system; c) Ox coordinate system; d) Oy coordinate system.

Table 5: Standard deviation of error between actual trajectory and trajectory estimated by algorithms in case 2

RMSE	Along the x-axis	Along the y-axis
Algorithm 1: Odometry-based	0.2705	0.0553
Algorithm 2: Odometry + LiDAR-based	0.4231	0.2071
Algorithm 3: Odometry + UWB-based	0.0361	0.0443
Algorithm 4: Odometry + LiDAR + UWB-based	0.0385	0.0425

Python environment and a visual simulation in the ROS-Gazebo environment are performed. The numerical simulation shows that the proposed algorithm reduces drift and improves the trajectory accuracy of the robot relative to using only the odometry sensor. As the number of UWB stations increases, the positional accuracy along both axes improves, while the odometry-based algorithm remains unaffected. This confirms that the proposed algorithm enhances the robot's position determination. Secondly, the visual simulation shows that the proposed algorithm provides better positioning accuracy, especially when combined with the UWB system. In Case 1, all algorithms perform well, with Algorithm 4 yielding the best results. In Case 2, when the LiDAR signal is cut off, algorithms 1 and 3 maintain accurate robot pose estimates, whereas the estimation of algorithm 2 becomes inaccurate. Algorithm 4 maintains accuracy during LiDAR cutoff by relying on both odometry and UWB signals. Finally, the proposed algorithm demonstrates strong potential for real-world applications, particularly in environments where accurate and reliable localization is critical. Its ability to integrate UWB, odometry, and LiDAR sensors ensures robustness against typical challenges such as signal interference, sensor failures, and dynamic changes in the environment. Although real-time processing capabilities were not the primary focus of this study,

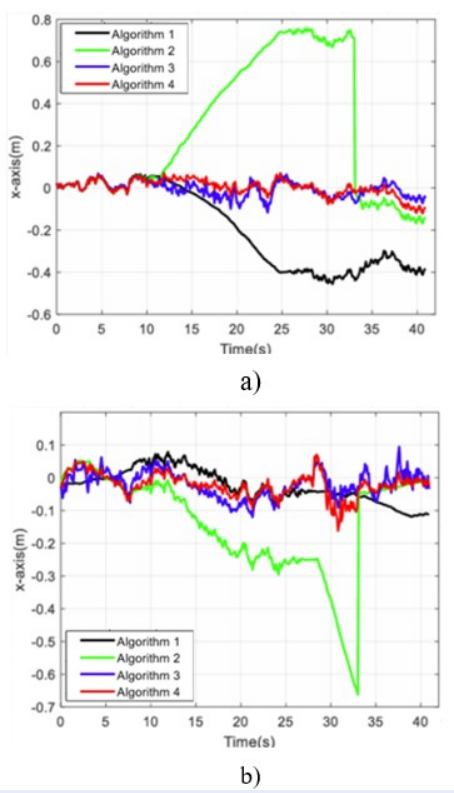


Figure 16: Errors in case 2: a) x-axis and b) y-axis of algorithms with respect to the ground truth.

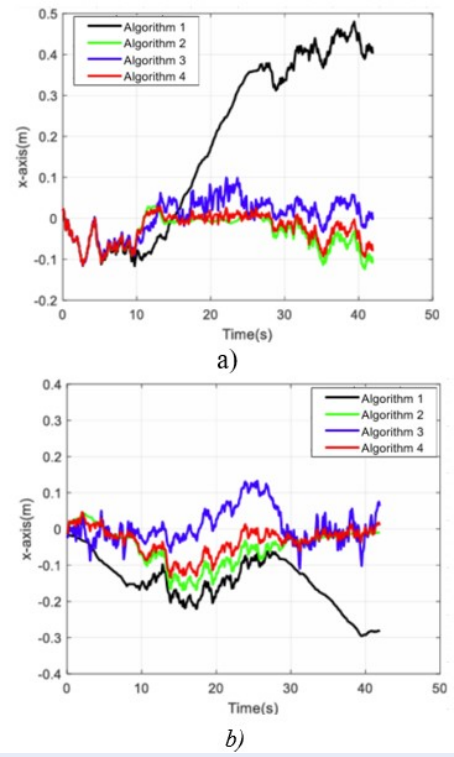


Figure 14: Errors in case 1: a) x-axis and b) y-axis of algorithms with respect to ground truth

future research should explore ways to optimize computational efficiency, ensuring that the algorithm can meet the time constraints required for real-time applications. This combination of high accuracy, stability, and potential for further optimization positions the proposed approach as a valuable solution for future autonomous mobile systems.

ABBREVIATIONS

UWB: Ultra-Wideband
LiDAR: Light Detection and Ranging
EKF: Extended Kalman Filter
SLAM: Simultaneous Localization and Mapping
TDoA: Time Difference of Arrival
AMCL: Adaptive Monte Carlo Localization
ROS: Robot Operating System
IMU: Inertial Measurement Unit
GPS: Global Positioning System
ToA: Time of Arrival
TW-ToA: Two Way Time of Arrival
INS: Indoor Navigation System

COMPETING INTERESTS

The authors declare no conflict of interest.

AUTHORS' CONTRIBUTIONS

All authors contributed to the conception, methodology, experimentation, analysis, and manuscript preparation. All authors reviewed and approved the final manuscript.

ACKNOWLEDGEMENTS

This research was implemented at the Robotics and Intelligent Control Laboratory (RIC Lab), Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Vietnam.

REFERENCES

1. P M, G R, G L, D S, GD P, et al. Smart nursing robot for COVID-19 patients. In: International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE). IEEE.; 2021.
2. Kamlofsky J, Naidoo N, Bright G, Bergamini M, Zelasco J, Ansaldo F, et al. Semi-Autonomous Robot Control System with an Improved 3D Vision Scheme for Search and Rescue Missions. In: and others, editor. A Joint Research Collaboration between South Africa and Argentina; 2018.
3. Siegwart R, Nourbakhsh IR, Scaramuzza D. Introduction to autonomous mobile robots. MIT press; 2011.

4. ST K, H K, M M, S M, A F, R M, et al. Smart robot navigation using rgb-d camera. In: International Conference on Applied Smart Systems (ICASS). IEEE.; 2018.
5. F L, B D, et al. Mobile robot localization based on particle filter. In: and others, editor. Proceeding of the 11th World Congress on Intelligent Control and Automation. IEEE.; 2014.
6. DB J, NY K, et al. Dead Reckoning of a Mobile Robot in 2-Dimensional Special Euclidean Group. In: and others, editor. 22nd International Conference on Control, Automation and Systems (ICCAS); 2022.
7. 2013 International Symposium on Next-Generation Electronics. IEEE.; 2013.
8. RA D, MA H, et al. A navigation scheme for autonomous mobile service robots working in GPS denied commercial indoor spaces. In: and others, editor. International Conference on Communication, Circuits, and Systems (IC3S); 2023.
9. Leitch SG, Ahmed QZ, Abbas WB, Hafeez M, Lazaridis PI, Sureephong P, et al. On indoor localization using wifi, ble, uwb, and imu technologies. Sensors (Basel). 2023;23(20):8598. Available from: https://doi.org/10.3390/s23208598.
10. Long Z, Xiang Y, Lei X, Li Y, Hu Z, Dai X. Integrated indoor positioning system of greenhouse robot based on uwb/imu/odom/lidar. Sensors (Basel). 2022;22(13):4819. Available from: https://doi.org/10.3390/s22134819.
11. Accurate UWB and IMU based indoor localization for autonomous robots. In: A M, B T, SK M, S A, K T, et al., editors. IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE); 2017.
12. and He D HZ. Multi-Sensor Fusion Localization and Mapping of Indoor Mobile Robot. In: and others, editor. 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE); 2022.
13. Study on the indoor mobile robot localization based on multi-sensor fusion. Journal of Physics: Conference Series. IOP Publishing; 2024.
14. Ranjan R, Shin D, Jung Y, Kim S, Yun JH, Kim CH, et al. Comparative analysis of integrated filtering methods using UWB localization in indoor environment. Sensors (Basel). 2024;24(4):1052. Available from: https://doi.org/10.3390/s24041052.
15. M F, R H, M A, K AM, H M, et al. Robot localization using extended kalman filter with infrared sensor. In: and others, editor. IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA); 2014.
16. Welch G, Bishop G. An introduction to the Kalman filter; 1995.
17. Weng Y, Xiao W, Xie L. Total least squares method for robust source localization in sensor networks using TDOA measurements. Int J Distrib Sens Netw. 2011;7(1). Available from: https://doi.org/10.1155/2011/172902.