IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 April 2026, accepted 4 June 2026, date of publication 9 June 2026, date of current version 15 June 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3701954

RESEARCH ARTICLE

# High Precision Localization of Mobile Robots in Indoor Environments With UWB and Odometry

MINA KHOSHRANGBAF[1], MURAT AYDEMIR[2], NUSIN AKRAM[2], GUL BOZTOK ALGIN[2], GEYLANI KARDAS[2], AND VAHID KHALILPOUR AKRAM[2]

[1]Department of International Computer, Graduate School of Natural and Applied Sciences, Ege University, 35100 İzmir, Türkiye
[2]International Computer Institute, Ege University, 35100 İzmir, Türkiye

Corresponding author: Vahid Khalilpour Akram (vahid.akram@ege.edu.tr)

This work was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under Grant 123E094.

**ABSTRACT** Determining the real-time and accurate location of mobile robots within indoor environments is crucial for a wide range of applications. Due to the limitations of GPS signals, alternative localization techniques are needed to determine the real-time location of mobile robots in indoor environments. This paper investigates the accuracy of Odometry and Ultra-Wideband (UWB)-based localization methods for detecting the location of mobile robots navigating complex trajectories. Odometry relies on wheel rotations or movement sensors to estimate a robot's position relative to its starting point, while UWB technology utilizes short-range radio waves to achieve precise distance measurements between UWB-equipped devices. We propose an adaptive localization method inspired by the weighting concept of Kalman filtering. Unlike a classical Kalman filter, the proposed method does not implement an explicit prediction-correction state cycle. Instead, it adaptively updates the contribution of UWB and Odometry according to their estimated error margins and the discrepancy between their reported coordinates. The method dynamically adjusts the contribution of each system based on their estimated error rates, relying more on Odometry when UWB experiences temporary interference, and shifting weight toward UWB as Odometry drift accumulates over time. To evaluate the accuracy of the proposed method, we conducted various experiments in a testbed environment with six UWB nodes and mobile robots equipped with Odometry sensors. Various trajectories, including circles, squares, triangles, and complex shapes, were tested. To evaluate the localization accuracy, combined coordinates were generated by integrating different proportions of Odometry and UWB outputs. The results indicate that the proposed method determines the location of the mobile robot with between 13.06% and 33.59% lower average error compared to UWB, Odometry, and their different combinations.

**INDEX TERMS** Autonomous mobile robots, indoor localization, odometry, ultra-wideband.

## I. INTRODUCTION

Mobile robots are increasingly used in various fields, particularly for indoor tasks that require continuous and precise navigation. These robots are capable of performing a range of activities, such as delivering heavy objects, disinfecting rooms, conducting surveillance tours, collecting garbage, and cleaning floors. However, several significant challenges must be addressed effectively before these robots can be deployed for real-world tasks. One of the key challenges of using mobile robots in indoor environments is accurately determining their real-time location. In the absence of GPS signals, indoor localization becomes a critical requirement for unlocking the full potential of autonomous mobile robots [1].

Generally, indoor localization refers to the process of determining the position of a robot within enclosed spaces, where traditional GPS systems often fail to provide accurate results due to signal attenuation [2]. This challenge has spurred the development of alternative localization techniques using various methods including Odometry, vision-based, and radio signal-based approaches. In the radio signal-based approaches, the location of the mobile robots is determined using different kinds of radio signals including Bluetooth Low Energy (BLE), WiFi, Ultra-Wideband (UWB), and

The associate editor coordinating the review of this manuscript and approving it for publication was Dusan Grujić.

Wireless Sensor Networks (WSNs). In the Odometry method, the location of the robot is determined based on the wheel rotation speed and movement sensors.

UWB technology operates by transmitting short-range radio waves over a wide spectrum, enabling highly precise distance measurements between UWB-equipped devices. UWB systems generally utilize Time-of-Flight (ToF) measurements to calculate distances between devices. By precisely measuring the time it takes for electromagnetic waves to travel between UWB-equipped devices, accurate distance estimations can be obtained. This approach enables UWB systems to achieve high accuracy in determining the range between devices, leading to precise localization. The wide bandwidth of UWB allows for more accurate time measurements, as the pulses can be accurately distinguished even in noisy environments. The very short length of signals allows the UWB receivers to mitigate multipath effects, where signals reflect off surfaces and arrive at the receiver at slightly different times.

However, UWB-based localization is susceptible to multipath interference and non-line-of-sight (NLOS) conditions caused by obstacles and walls, which can introduce temporary errors in the estimated position. Furthermore, UWB systems require a fixed infrastructure of anchor nodes, and their accuracy may degrade near the boundaries of the anchor coverage area. These limitations motivate combining UWB with Odometry, which provides continuous and infrastructure-free position updates, to achieve more robust localization.

On the other hand, Odometry relies on the measurement of wheel rotations or movement sensors to estimate a robot's position relative to its starting point. While Odometry offers a cost-effective and reliable solution for short-term navigation, its accuracy tends to degrade over time due to wheel slippage, uneven terrain, or cumulative errors. However, when combined with other localization techniques like UWB, Odometry can compensate for short-term inaccuracies and provide continuous real-time updates on the robot's position.

In this paper, we propose a new method to combine the received coordinates from UWB and Odometry and generate more accurate location estimation. We compare the performance of the proposed method with the accuracy of Odometry and UWB-based localization methods and their different combinations for detecting the location of mobile robots on complex trajectories. The main contributions of the paper are listed as follows:

1) This study proposes a new method that uses both UWB- and Odometry-based systems to find more accurate location of mobile robots.
2) In addition to the proposed method, we evaluated different combinations of UWB and Odometry coordinates and compared them with the proposed method.
3) We evaluated the proposed methods over complex trajectories on a testbed of mobile robots and UWB devices.
4) The experimental results show that the proposed adaptive method can generate more accurate coordinates with between 13.06% and 33.59% lower average error than UWB, Odometry, and their different combinations.

The remaining parts of this paper have been organized as follows; Section II provides a short survey about the existing approaches. Section III provides the details of the proposed algorithm. Section IV describes the details of implemented platform for evaluation. Section V includes the performance evaluation result and finally Section VI draws the conclusion.

## II. RELATED WORKS

Indoor localization of mobile robots is a fundamental task in most multi-robot systems because the appropriate working of high-level applications mostly depends on the accurate localization of robots. Without knowing the correct location of robots assigning the tasks and making correct decisions by the robots can be very complicated. Hence, a wide range of research is in progress to improve the accuracy of indoor localization methods for mobile robots.

For many years, the Odometry method was the main localization technique for mobile robots [3]. In this method, the location of the robot is calculated using the previous location and the received data from wheel encoders. The robot's position is detected by tracking the distance and direction of wheel rotations. However, the accuracy of the Odometry method is reduced over time due to wheel slippage and errors from wheel encoders. To improve the accuracy of Odometry, different methods have been proposed by the researchers. Using redundant complementary localization techniques such as wireless sensor networks [4], [5], Bluetooth low energy (BLE) [6], WiFi signals [7], and visible light [8] are some of these methods.

In recent years the UWB technology has been considered as one of the most promising technologies for indoor localization. The very short length of UWB signals allows the receiver to distinguish the direct and reflected signals and calculate the distance to the sender more precisely. Many approaches have been proposed in the literature to utilize the UWB signals for locating mobile robots in indoor environments. In [9] different localization algorithms have been evaluated using UWB signals for locating the mobile robot in line-of-sight and non-line-of-sight environments. However, the dimensions of the environment are less than 80 cm which is relatively small for most applications. The authors of [10] have proposed a cooperative relative localization method for autonomous aerial vehicles using UWB signals and Odometry data. In the proposed method, each node can estimate the relative positions of its neighbors using UWB-based distance detection and Odometry data. Similarly, the authors of [11] have proposed another method to combine the Odometry and UWB localization data to minimize the error rate of both methods. They have provided probabilistic and experimental analysis for their proposed method. The experimental results show that their proposed method may limit the maximum error to 20 cm with a probability of 90%.

To locate multiple robots at the same time, a UWB-based clock synchronization scheme has been proposed in [12] which allows the robots to determine their location by measuring the time difference of signal arrivals. This method only depends on the UWB signals and does not use the Odometry data. Using Kalman filters [13], [14], [15], minimizing the residual error of the UWB ranging measurements [16], feature extraction and landmark detection by visual-inertial Odometry [17], [18], using a probabilistic model and particle filter [19], [20], merging with 3D point-clouds using RGB-D sensors [21], minimizing the UWB ranging measurements taken at different positions [22], random particle generation and adaptive Monte Carlo localization [23] are some of the other proposed methods in the literature that combine the Odometry and UWB positioning data with different techniques.

Lin and Yeh [24] proposed a drift-free visual SLAM method for mobile robot localization by integrating UWB technology. However, their approach relies on visual features and camera-based sensing, which is sensitive to lighting conditions and textureless environments. Magnago et al. [11] combined Odometry and UWB localization data to minimize the error rate, providing stochastic guarantees with a maximum error of 20 cm at 90% probability. However, their method uses fixed fusion weights and does not adaptively adjust the contribution of each system based on real-time error estimation. Brunacci et al. [25] developed and analyzed a UWB relative localization system, demonstrating its effectiveness for short-range applications. However, their method does not incorporate Odometry data, and the localization accuracy degrades in the absence of a sufficient number of anchor nodes. In contrast to these works, the proposed method combines UWB and Odometry using an adaptive Kalman filter-based approach, which addresses both the cumulative drift of Odometry and the temporary position spikes of UWB without relying on visual sensors or fixed fusion weights. We compare the accuracy of the proposed method with the Odometry and UWB-based localization to find the location of mobile robots on complex trajectories. Also, we evaluate different merging schemas to combine the incoming position data from Odometry and UWB channels.

## III. PROPOSED METHOD

Most UWB-based localization systems usually offer up to 50 cm accuracy which is acceptable in many applications [2], [26]. By homogeneous distribution of UWB anchors in the environment, we may expect bounded error margin regardless of passed distance by the robots. On the other hand, the initial error ratio of Odometry system is very low but, it usually increases steadily while the robot moves around the environment. Integrating the estimated location from the UWB and Odometry systems reduces the error margin of the localization system.

Let $(x_t^o, y_t^o, \theta_t^o)$ be the reported location and direction from Odometry system at time $t$. If we receive $(x_t^u, y_t^u, \theta_t^u)$ estimated location from UWB at time $t$, we can merge the two coordinates to get more accurate estimation. As the simplest method we can take the average of the received positions:

$$(x_t, y_t, \theta_t) = (\frac{(x_t^o + x_t^u)}{2}, \frac{(y_t^o + y_t^u)}{2}, \frac{(\theta_t^o + \theta_t^u)}{2}) \quad (1)$$

However, since positioning systems have different error margins, the accuracy of the detected position can be increased by giving more weight to the system with a lower error.

To find more accurate coordinates, we propose an adaptive localization method inspired by the weighting concept of Kalman filtering. This method works based on the information received from the UWB sensor and Odometry. The Kalman Filter is one of the most widely used methods for data fusion. This method is used to combine absolute and relative positions in robot localization. The basic idea of the Kalman filter is to use a model of the system being measured and update the model as new measurements become available. It updates the current state using weighted combination of the previous measurements. In our case, the weighting coefficient can be calculated from the error rates as follows:

$$a = \frac{e_{uwb}}{e_{odo}^t + e_{uwb}}. \quad (2)$$

where $e_{odo}^t$ is the estimation error of Odometry at time $t$ and $e_{uwb}$ is the error margin of the UWB system. We propose a new adaptive method to adjust this weight based on the estimated error of both Odometry and UWB systems up to that point and the difference between the generated coordinates by two system. In the proposed methods, the new adaptive coordinate is calculated as follows:

$$\begin{aligned} x &= (a \times x_{odo}) + ((1 - a) \times x_{uwb}) \\ y &= (a \times y_{odo}) + ((1 - a) \times y_{uwb}) \end{aligned} \quad (3)$$

Let $e_{odo}^t > 0$ and $e_{uwb} = 0$, which means that we have some error in Odometry and no error in UWB systems. In this case, we get $a = 0$ from (2) and ignore the received coordinate from Odometry in (3). If we have $e_{odo}^t = 0$ and $e_{uwb} > 0$ then we get $a = 1$ and the resulting coordinate in Equation 3 is set to Odometry coordinate. The main challenge is estimating the updated error ratios $e_{odo}^t$ and $e_{uwb}$ as we do not have real coordinate of the robot to measure these error ratios. To solve this problem we propose an adaptive method that estimates the error ratios of both methods using previous coordinates.

The main idea is to estimate the average error based on the distance between previous and current coordinates and also the difference between received coordinates from UWB and Odometry channels. The steps of the proposed algorithm are presented in Algorithm 1. Since the error margin of Odometry system is low at the beginning of movement, we set the initial error margin of Odometry ($e_{odo}$) to 1 (line 1). The error margin of UWB system is not affected by the passed distance by the robot. The accuracy of UWB system initially depends on the current location of the robot and anchor nodes. Hence we set the initial error margin of UWB system ($e_{uwb}$) to an expected error $\omega$. The initial distance of both system and also the number of received coordinates is set to 0 (line 2). The estimated coordinate for robot is stored in $c$ which initially is set to the robot's starting position.

**Algorithm 1** Adaptive Localization

1: $e_{uwb} \leftarrow \omega, e_{odo} \leftarrow 1.$
2: $uwb \leftarrow 0, odo \leftarrow 0, n \leftarrow 0.$
3: $c \leftarrow$ initial coordinates of robot's location.

4: **upon receiving** new coordinates $c_{uwb}$ and $c_{odo}$
5: $\quad uwb \leftarrow uwb + dist(c_{uwb}, c)$
6: $\quad odo \leftarrow odo + dist(c_{odo}, c)$
7: $\quad n \leftarrow n + 1$
8: $\quad m_{uwb} = uwb/n$
9: $\quad m_{odo} = odo/n$
10: $\quad$ **if** $m_{uwb} > m_{odo}$ **then**
11: $\qquad e_{odo} \leftarrow e_{odo} + 1$
12: $\qquad e_{uwb} \leftarrow min(\omega, e_{uwb} + 1)$
13: $\quad$ **else**
14: $\qquad e_{odo} \leftarrow e_{odo} + \sigma$
15: $\qquad e_{uwb} \leftarrow max(5, e_{uwb} - \sigma)$
16: $\quad$ **if** $dist(c_{uwb}, c_{odo}) < 100\,mm$ **then**
17: $\qquad e_{odo} = e_{odo} - \sigma$
18: $\quad$ **if** $dist(c_{uwb}, c_{odo}) > 500\,mm$ **then**
19: $\qquad a = 1$
20: $\quad$ **else**
21: $\qquad a = e_{uwb}/(e_{odo} + e_{uwb})$
22: $\quad c = a \times c_{odo} + (1 - a) \times c_{uwb}$

When the robot moves, the algorithm adjusts the error margins of the Odometry and UWB systems based on the distance between the robot's previous and current locations. Upon receiving new coordinates from the UWB and Odometry systems (line 4), the distance between new coordinates and current position is calculated (lines 5, 6). The average distance between the coordinates reported by the UWB and Odometry systems is calculated and stored as $m_{uwb}$ and $m_{odo}$, respectively (lines 7-9). If the Odometry system has a lower average distance, the error margin of the Odometry system is slightly increased by 1. We also increase the error of UWB system by 1, if it is lower than the expected value $\omega$ (lines 11, 12).

If the average distance of the UWB system is lower than that of the Odometry system, the error margin of the Odometry system is increased by a larger value, denoted as $\sigma$, while the error margin of the UWB system is decreased by the same value (lines 14,15). In this way, the algorithm increases the weight of the UWB system for estimating the coordinates of the robot.

If the difference between the coordinates reported by the UWB and Odometry systems is less than 10 cm, we can assume that the robot's actual location is very close to the reported coordinates. Consequently, the error margin of the Odometry system is decreased by $\sigma$ (lines 16,17). On the other hand, if the difference between the reported coordinates exceeds 50 cm, we can conclude that the most recent coordinate reported by the UWB system is temporarily inaccurate. In this case, we set $a = 1$ to ignore the UWB system's coordinate (lines 18,19). Otherwise, we calculate the weighting coefficient based on Equation 2 (lines 20,21). Finally we estimate the new coordinate of robots based on Equation 3 (line 22). In the evaluations, we set $\sigma$ to 20 and $\omega$ to 80. The parameter $\omega = 80$ represents the initial expected error margin of the UWB system in millimeters, which was set based on the manufacturer's specified accuracy range of 10–30 cm for the POZYX system. The parameter $\sigma = 20$ controls the rate at which error margins are updated and was determined empirically to provide a balance between responsiveness and stability. The threshold of 100 mm was chosen as the boundary below which the two systems are considered to be in agreement, reflecting the typical accuracy range of the UWB system. The threshold of 500 mm was set as the upper bound for detecting UWB outliers, since a discrepancy larger than 50 cm between UWB and Odometry readings most likely indicates a temporary UWB measurement error. The minimum value of 5 in line 15 prevents the UWB error margin from reaching zero, ensuring that the Odometry system always retains a minimum contribution to the estimated position.

To evaluate the performance of the proposed algorithm, we implemented all the methods described in this section on a testbed of mobile robots, which is detailed in the next section.

## IV. IMPLEMENTED PLATFORM

We developed a mobile robot localization platform that provides high-level controlling and monitoring functions to manage the mobile robots. In this section, we explain the main components and general architecture of the implemented platform.

### A. ROBOT

As mobile robots, we used Kobuki robots which are widely used in research, education, and industrial applications [27]. Kobuki is compatible with the Robot Operating System (ROS), a widely used open-source robotics middleware, and provides ROS drivers and software libraries. The base of the robot is equipped with sensors, actuators, and interfaces necessary for autonomous navigation and interaction with the environment. The robot utilizes wheel encoders to measure the rotation of its wheels, enabling Odometry-based localization and motion control. By tracking the distance and direction traveled by each wheel, the robot can estimate its position and orientation relative to a starting point. An Internal Measurement Unit (IMU) consisting of accelerometers and gyroscopes provides information about the robot's acceleration and angular velocity. Kobuki is equipped with Cliff and Bumper sensors to perceive its surroundings and interact with the environment effectively. It also provides $5V$ and $12V$ power sources for other devices.

We used a Raspberry Pi 3B as a gateway between the robot and a web application. A Raspberry Pi is connected to the robot via a USB port to send control commands and receive real-time Odometry data from the robot. Additionally, an HC-SR04 Ultrasonic Distance Sensor is positioned at the front of the robot and connected to Raspberry Pi via GPIO pins to continuously measure the distance to the obstacles. Kobuki Robot has its battery, and it can provide power for Raspberry Pi via its $5V$ power outputs.

As the robot moves in the environment the accuracy of the calculated coordinates from Odometry decreases. Different factors such as friction and wheel encoder precision may increase the error margin of the Odometry over time. Kobuki robots also have factory-calibrated built-in 3-axis Digital Gyroscope with a measurement range of $\pm 250 deg/s$. We used Odometry readings for position updates at the x and y axes and utilized the gyroscope data to determine the Yaw axis angle.

### B. ULTRA-WIDE BAND

UWB technology has emerged as a promising wireless communication technique, characterized by its ability to transmit data over a wide spectrum of frequencies, with low power and high precision. UWB operates by transmitting very short pulses of electromagnetic energy across a broad range of frequencies, ranging from 3.1 GHz to 10.6 GHz. These pulses are extremely short in duration, often in the picosecond range, resulting in signals that occupy a large portion of the frequency spectrum.

UWB systems measure the time-of-flight of signals between transmitters and receivers to calculate the distance between them. In indoor environments, UWB-equipped devices (beacons or anchors) are deployed at known locations. The anchors emit UWB pulses, which are received by tags. Each tag measures the time-of-flight of the received signal, allowing them to calculate the distance to the anchor. By combining distance measurements from at least three anchors, the tag's position can be determined through a Multilateration process. UWB signals exhibit robustness against multipath interference, which commonly occurs in indoor environments due to signal reflections off walls, furniture, and other objects. Unlike narrowband signals, UWB pulses have very short durations, making it easier to distinguish between direct and reflected signals. UWB systems can achieve centimeter-level accuracy in determining the position of tags within indoor spaces.

The POZYX system used in this work supports two positioning protocols: Two-Way Ranging (TWR) and Time Difference of Arrival (TDOA). In TWR, the tag initiates communication with each anchor one by one, sending a packet back and forth. The round-trip time of this exchange is used to estimate the distance between the tag and the anchor.

(a) (b) (c)

**FIGURE 1. (a) A UWB tag device attached to mobile robot (b) a UWB anchor device and (c) a NodeMCU-ESP32 device.**

Once the tag has ranged with at least three, ideally four anchors, its 2D position is computed through multilateration. In multilateration, if the distances from the tag to anchors $A_1, A_2, \ldots, A_n$ are $d_1, d_2, \ldots, d_n$, the tag position $(x, y)$ is estimated by solving the following system of equations:

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2, \quad i = 1, 2, \ldots, n \tag{4}$$

where $(x_i, y_i)$ are the known coordinates of anchor $i$. Since measurements contain noise, in practice the system minimizes the residual error across all anchor measurements using a least-squares approach. Fig. 1a shows the used tag and Fig. 1b shows the used anchor devices in the experiments.

The detected coordinates by the tag devices are sent to the Raspberry Pi over the USB port. Using the provided middleware the received data from the driver is forwarded to an MQTT TOPIC over port 1883. We used Mosquitto broker on the Raspberry Pi to receive MQTT messages and forward them to a web server. The developed Python and C++ programs and also sample videos are available on the project GitHub page at https://github.com/vahidka/Kobuki

### C. BLUETOOTH AND WIFI

To compare the accuracy of the proposed method with BLE and WiFi based systems, we developed a BLE and a WiFi-based system using NodeMCU-ESP32 devices (Fig. 1c). The same testbed environment and trajectories used for UWB and Odometry experiments were also used for BLE and WiFi evaluations, ensuring a fair comparison under identical conditions. We used 9 NodeMCU devices to probe the broadcasted BLE or WiFi signals from the mobile target (Raspberry Pi). The NodeMCU-ESP32 device has both WiFi and BLE modules and an Xtensa 32-bit LX6 microprocessor. Fig. 1a shows a NodeMCU device that we used in BLE and WiFi based experiments.

We implemented an embedded program to run on the ESP nodes and a NodeJS based web application to collect the information and monitor the real time location of mobile robots. The embedded program in the ESP nodes sends the received signal strength indicator (RSSI) of the BLE and WiFi signals to a web server over HTTP protocol. The server collects the received data from all ESP nodes and determine the location of mobile robot using weighted averaging method. This method estimates the location of the target robot by taking the weighted average coordinates of the surrounding anchor nodes. The signals that are received by closer anchor nodes to the robots will have higher RSSI

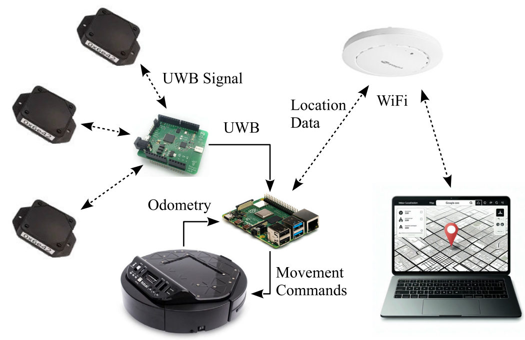



**FIGURE 2.** **General architecture of the implemented platform.**

values than far anchor nodes. So we may estimate the coordinates of a robot using the following formula:

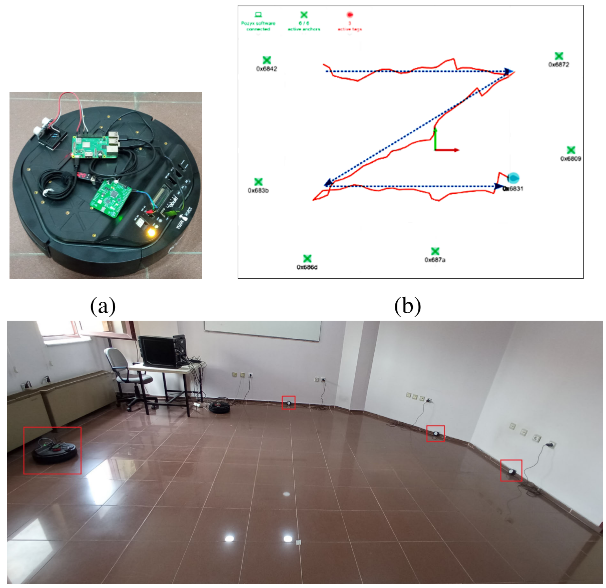
$$pos = \sum_{i \in A} (RSSI_i \times pos_i) / \sum_{i \in A} RSSI_i \tag{5}$$

In this formula, $A$ is the set of anchor nodes, $RSSI_i$ is the received RSSI from anchor $i$ and the $pos_i$ is the coordinates of anchor $i$. Many sophisticated variants of this formula have been proposed in different studies [28], [29], however, all of them depend on the RSSI values and the averages of the anchor's coordinates. The RSSI values are sensitive to the reflections and obstacles hence if we do not have line of sight between the anchors and the target mobile object, the estimated location can be far from the real location.

### D. GENERAL ARCHITECTURE

The collected coordinates from Odometry UWB, and BLE/WiFi channels are sent to a web server using HTTP protocol. We used a local WiFi network to make the connections between the robots and the web server. The Raspberry Pi on the robot connects to the WiFi access point and sends HTTP ping messages to the web server every 10 seconds. Besides the ping messages, location messages are sent to the web server when new coordinates are received from UWB or Odometry channels. We implemented a NodeJS-based web server to record the received coordinates from the robots. The web server provides an interface to show the real-time location of the robots, their speed, and other information. Fig. 2 shows the general architecture of the implemented system.

After receiving the coordinates from UWB, Odometry, and WiFi/BLE channels we recorded the coordinates in a local file and used these data to evaluate the accuracy of the methods. Fig. 3a shows the robot used in the experiments and Fig. 3b shows a screenshot of the POZYX UWB-based localization system that contains the actual and detected movement paths. Fig. 3c shows the testbed environment where the robot and UWB anchors are marked with red rectangles. In the next section, we present the results of experimental evaluations.



(c)

**FIGURE 3.** **(a) The robot used in the experiments, (b) the detected positions in a sample trajectory, (c) testbed environment.**

## V. EVALUATION RESULTS

In the experiments, we developed a testbed environment with six UWB anchor nodes positioned throughout the area, along with a robot equipped with an Odometry sensor, as depicted in Fig. 3. Experiments were conducted across ten distinct paths with different shapes and lengths. Circle, square, Z shape, snake, and triangle are some examples of the designed paths. Throughout these paths, coordinates generated by the Odometry, UWB, WiFi and BLE were recorded. Fig. 4 illustrates sample trajectories traversed by the robot, where the red points show the detected Odometry and the blue points show the UWB coordinates.

### A. EXPERIMENTAL SETUP

The testbed environment consists of a room measuring 5.60 m × 6.40 m (approximately 35.8 $m^2$). Six UWB anchor nodes were deployed along the walls of the room, with consecutive anchors placed approximately 3.5 m apart on average to provide a sufficient coverage for the entire area. Six anchor nodes were used as this number provides sufficient geometric diversity for accurate multilateration while maintaining a practical deployment density for the room dimensions. This configuration provides overlapping coverage across the room, allowing the tag to range with at least four anchors at any position, which is the recommended minimum for reliable 2D multilateration.

The actual path of the robot was defined as a set of waypoints. The robot was moved manually along the predefined path using the developed web application, which provides real-time visualization of the robot's position. By real-time tracking and manual control of the robot, we ensured that the robot followed the intended path as

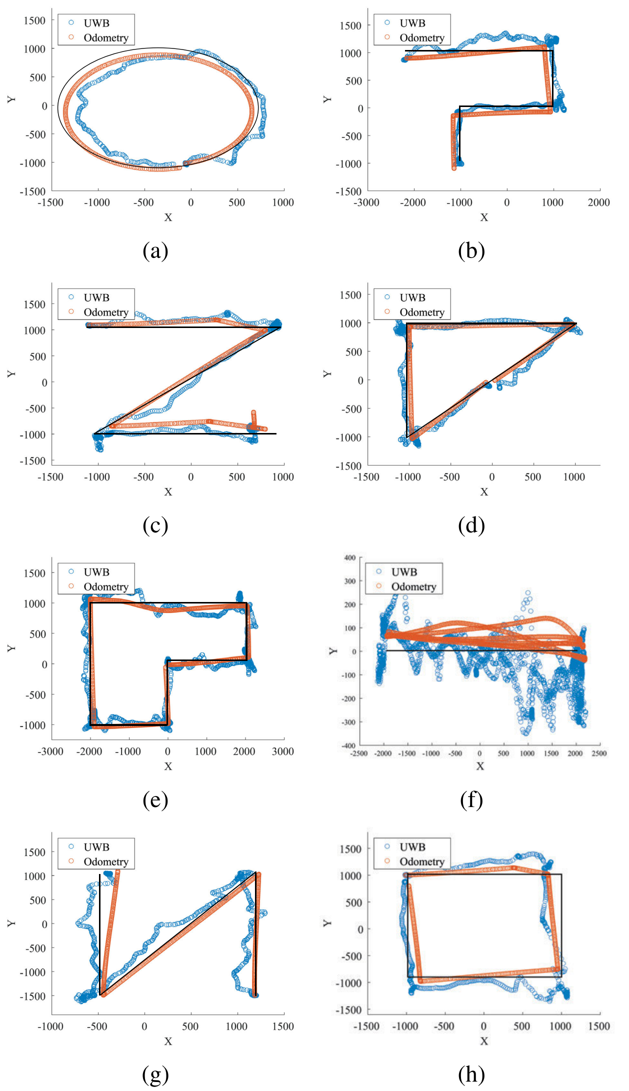



**FIGURE 4.** Sample trajectories of the robot where red points are the Odometry and blue points are UWB coordinates.

closely as possible. Although real-time visualization allowed the operator to follow the predefined path closely, the manual control procedure may introduce human-induced deviations from the intended trajectory. This deviation was not directly measured by an external motion capture or vision-based ground-truth system. Therefore, the reference path should be interpreted as a practical approximation of the traversed trajectory rather than an absolute ground truth. Based on the grid resolution and visual tracking interface used during the experiments, the expected manual deviation is estimated to be within a few centimeters, however, this value was not independently quantified.

### B. ERROR DISTRIBUTION ANALYSIS

We evaluated three combinational strategies in addition to the coordinates generated by the adaptive algorithm. These strategies involved calculating the weighted average of the

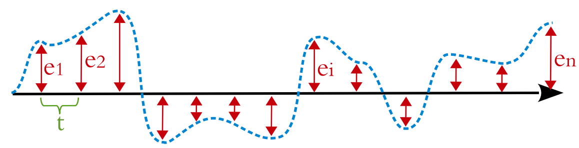



**FIGURE 5.** Estimating the error of detected coordinates.

detected coordinates, which were obtained by integrating different proportions of Odometry and UWB outputs as 20-80, 80-20, and 50-50 percent weightings. The computation for the 20-80 weighting model is as follows:

$$x = (0.2 \times x_{odo}) + (0.8 \times x_{uwb})$$
$$y = (0.2 \times y_{odo}) + (0.8 \times y_{uwb}) \tag{6}$$

In the above equations, $(x, y)$ represents the new coordinates, calculated using 20% of the Odometry output and 80% of the UWB output. The computation for the 50-50 weighting model, which assigns equal weights to both systems, is as follows:

$$x = (0.5 \times x_{odo}) + (0.5 \times x_{uwb})$$
$$y = (0.5 \times y_{odo}) + (0.5 \times y_{uwb}) \tag{7}$$

In (7), the estimated location is calculated by simply averaging the coordinates received from the UWB and Odometry systems. The computation for the 80-20 weighting model is as follows:

$$x = (0.8 \times x_{odo}) + (0.2 \times x_{uwb})$$
$$y = (0.8 \times y_{odo}) + (0.2 \times y_{uwb}) \tag{8}$$

To find the accuracy of the generated coordinates, we calculated the minimum distance of each detected coordinate from the closest point on the real path traversed by the robot and considered this value as the estimation error. Fig. 5 shows how the accuracy of the detected coordinates has been calculated. In this figure, the black line shows the actual traversed path by the robot and the dashed blue line shows the reported coordinates. The average error of each method is set to the average error of $e_1$ up to $e_n$ where $e_i$ is the minimum distance between detected coordinate and the actual path.

Fig. 6 displays the error of all methods across 8 sample trajectories. The error is depicted relative to the path length, where the path length and error are measured in centimeters and millimeters, respectively. The figure shows that, in all trajectories, the proposed adaptive algorithm generally has a lower error margin than other methods. This is expected, as the adaptive algorithm dynamically adjusts the weight of each system based on their estimated error rates, allowing it to rely more on the more accurate system at each step.

The box plot in Fig. 7 compares the localization errors of different methods. Fig. 7a and Fig. 7g shows that UWB may fail to find the correct location of robot with exceptionally high errors. However, the median error for UWB is lower than that of Odometry in Fig. 7b and Fig. 7c for Z shape and circle trajectories. Fig. 7d illustrates the estimation error

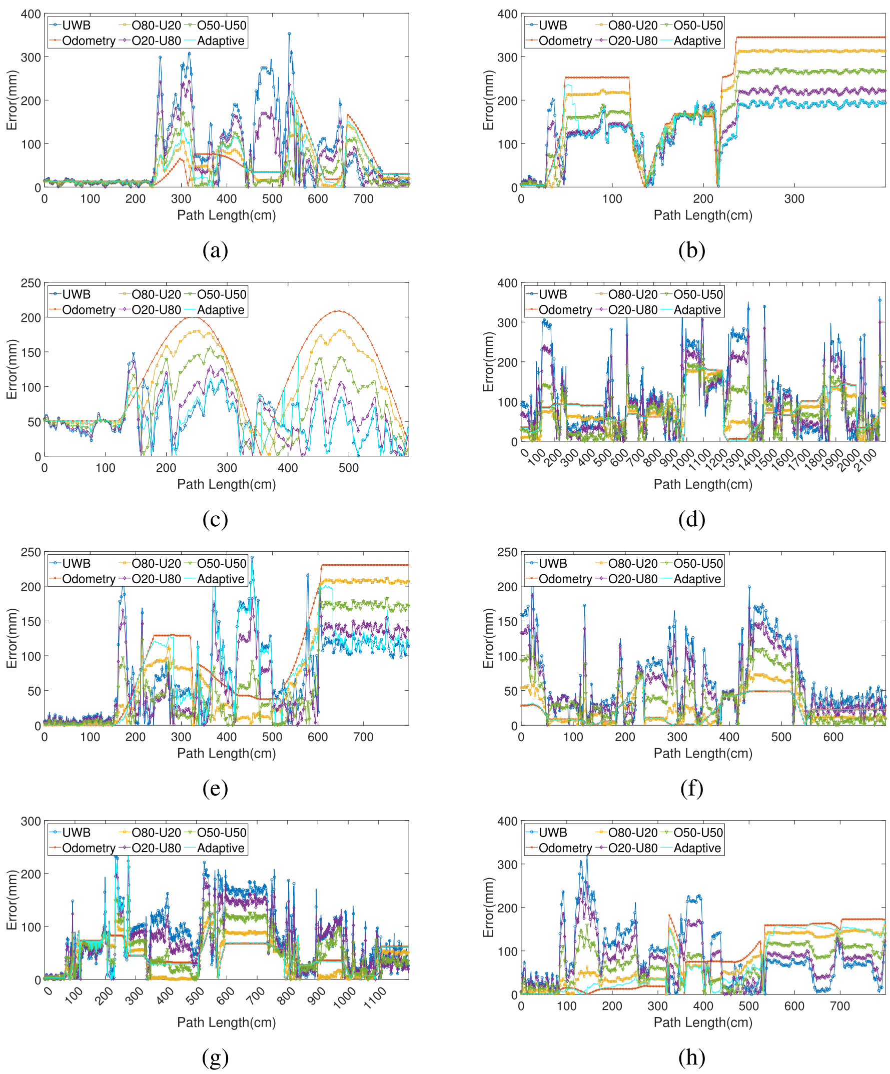



**FIGURE 6. The error trends of detected coordinates using different methods over 8 sample trajectories.**

for line trajectory where the interquartile range (IQR) of the Odometry localization method is smaller than UWB indicating more consistent performance. The results for the other trajectories are shown in Figs. 7e to 7h, respectively. Considering all trajectories, UWB shows a wider spread of errors, demonstrating greater variability. In terms of median error, the adaptive localization method consistently delivers better performances than all other methods in all trajectories. The wider spread of UWB errors can be attributed to temporary multipath interference and NLOS conditions, which cause occasional spikes in the estimated position. In contrast, Odometry errors grow more gradually and consistently due to cumulative wheel slippage and encoder drift.

### C. COMPARATIVE PERFORMANCE EVALUATION

To provide a comparative analysis of the methods across various paths, the Average Error Percentage (AEP) of all methods is computed for all trajectories as follows:

$$AEP = \frac{Ave_{Err} \times 100}{N}. \tag{9}$$

where:

$$Ave_{Err} = \frac{\sum_{i=1}^{N} e_i}{N}. \tag{10}$$

$N$ is the total number of samples or coordinates generated by the methods and $e_i$ denotes the error in the $i$ th coordinate. Fig. 8 presents the results of all trajectories arranged according to their respective path lengths. As depicted in the figure, for the first two trajectories with path lengths of 433 cm and 628 cm, the AEP of Odometry is higher than that of UWB. For the remaining paths, the disparities in performance between the methods are less pronounced. Across all trajectories, except for two routes, the adaptive method has a smaller AEP compared to both UWB and Odometry results. The higher AEP of Odometry in shorter trajectories is likely due

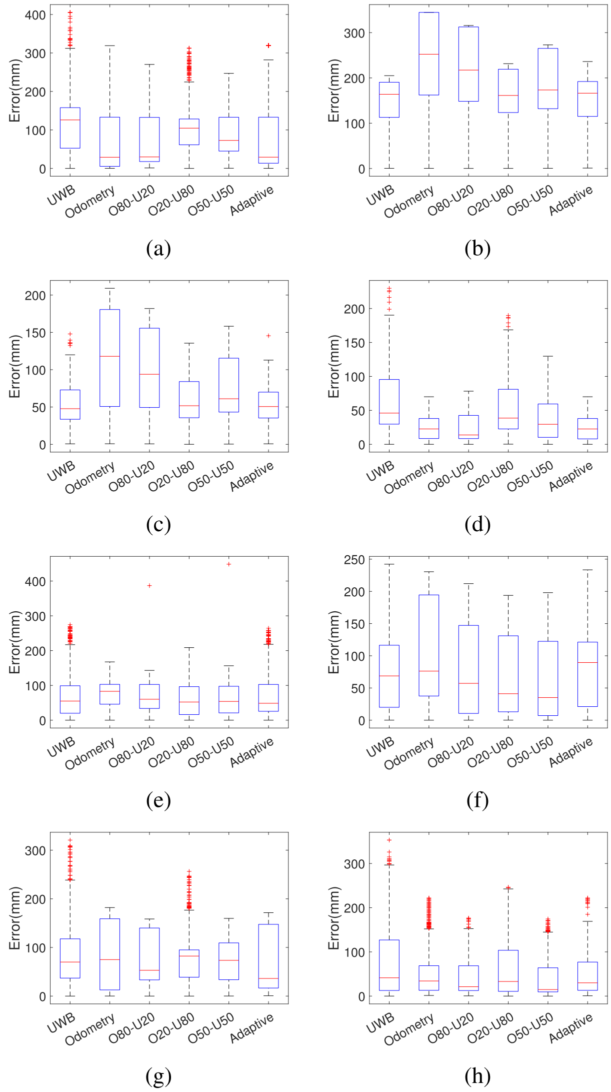



**FIGURE 7.** Error distribution of different methods for 8 sample trajectories.

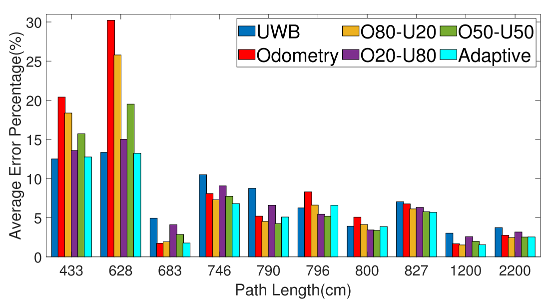



**FIGURE 8.** AEP of different methods against trajectory length.

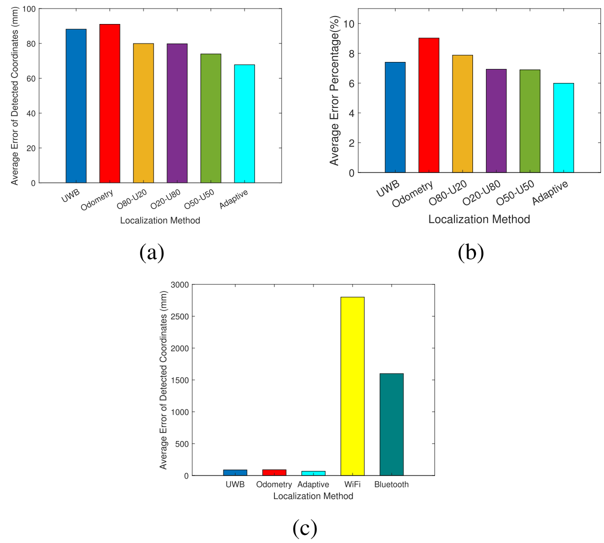



**FIGURE 9.** (a) Average error of the detected coordinates by the algorithms, (b) Overall AEP of methods, (c) Comparison of the average error of the detected coordinates by the proposed algorithm with BLE and WiFi based methods.

to the robot making sharper turns, which amplifies wheel slippage errors. For longer trajectories, the cumulative drift of Odometry becomes the dominant source of error, while UWB errors remain more bounded.

Subsequently, to assess the overall efficacy of the localization methods, the average error of the detected coordinates ($Ave_{Err}$) and the average of $AEP$ is computed across all traversed paths. As depicted in Fig. 9a, the average error of the detected coordinate using UWB is 88.19 mm which demonstrates better performance in comparison to Odometry with a 90.99 mm average error. In terms of average AFP, the coordinates generated by UWB exhibit a 17.96% lower error compared to Odometry (Fig. 9b).

The fusion of UWB with Odometry yields more precise localization compared to both Odometry and UWB methods. This enhancement becomes more pronounced with an increased contribution of UWB to the fusion process. In the combinational coordinates, the results obtained using 20% Odometry and 80% UWB outputs, denoted as O20-U80, exhibit superior performance compared to the other methods. Specifically, the O20-U80 combinational method enhances localization accuracy by 23.23% and 4.96%, in comparison to the Odometry and UWB methods, respectively. This trend is consistent with the observation that UWB generally provides more accurate absolute position estimates, and therefore a higher UWB weight in the fusion results in better overall accuracy.

On the other hand, the adaptive method with a 67.74 mm average error of the detected coordinates overcomes all the other methods. On average, the adaptive method presents 19.05%, 33.59%, and 13.06% lower AEP compared to UWB, Odometry, and the O50-U50 combinational methods, respectively. The superior performance of the adaptive method can be attributed to its ability to dynamically adjust the contribution of each system based on real-time error estimation, rather than using fixed weights as in the combinational methods.

Finally, Fig. 9c compares the average error of the detected coordinates using WiFi and BLE-based methods with UWB,

**TABLE 1.** Average Error Percentage (AEP) of all methods across different trajectories.

| Path (cm) | UWB (%) | Odo. (%) | O80-U20 (%) | O20-U80 (%) | O50-U50 (%) | Adap. (%) |
|---|---|---|---|---|---|---|
| 433 | 12.51 | 20.41 | 18.37 | 13.58 | 15.72 | 12.76 |
| 628 | 13.34 | 30.22 | 25.79 | 15.02 | 19.51 | 13.24 |
| 683 | 4.94 | 1.73 | 1.92 | 4.11 | 2.86 | 1.77 |
| 746 | 10.50 | 8.09 | 7.29 | 9.07 | 7.73 | 6.81 |
| 790 | 8.75 | 5.20 | 4.52 | 6.58 | 4.25 | 5.09 |
| 796 | 6.25 | 8.30 | 6.61 | 5.45 | 5.20 | 6.60 |
| 800 | 3.92 | 5.07 | 4.13 | 3.44 | 3.37 | 3.87 |
| 827 | 7.04 | 6.77 | 6.12 | 6.31 | 5.77 | 5.70 |
| 1200 | 3.03 | 1.67 | 1.52 | 2.58 | 1.98 | 1.54 |
| 2200 | 3.74 | 2.76 | 2.47 | 3.17 | 2.53 | 2.54 |
| **Mean** | **7.40** | **9.02** | **7.88** | **6.93** | **6.89** | **5.99** |

**TABLE 2.** Mean and standard deviation of AEP and average error of detected coordinates.

| Method | Mean AEP (%) | Std AEP (%) | Mean Error (mm) | Std Error (mm) |
|---|---|---|---|---|
| UWB | 7.40 | 3.73 | 88.20 | 27.73 |
| Odometry | 9.02 | 9.20 | 91.00 | 57.13 |
| O80-U20 | 7.87 | 7.93 | 79.94 | 50.65 |
| O20-U80 | 6.93 | 4.35 | 79.77 | 30.71 |
| O50-U50 | 6.89 | 5.97 | 73.95 | 40.09 |
| Adaptive | 5.99 | 4.14 | 67.75 | 32.63 |

Odometry, and the proposed adaptive method. Generally, the accuracy of BLE and WiFi-based systems is lower than that of UWB-based systems. The average errors of the WiFi and BLE-based methods in all trajectories are 2851 mm and 1608 mm, respectively, which are higher than the other methods. Specifically, the average error of the WiFi-based system is 42 times higher, and the average error of the BLE-based system is 23.7 times higher than the proposed adaptive algorithm.

Table 1 presents the AEP of all methods across the ten evaluated trajectories. As shown in the table, the adaptive method achieves the lowest mean AEP of 5.99%, outperforming UWB (7.40%), Odometry (9.02%), O80-U20 (7.88%), O20-U80 (6.93%), and O50-U50 (6.89%).

Table 2 summarizes the mean and standard deviation of the AEP and average error of the detected coordinates for all evaluated methods. The adaptive method achieves the lowest mean AEP (5.99%) and the lowest mean average error (67.75 mm), outperforming all other methods. Furthermore, the standard deviation of the adaptive method (4.14% for AEP and 32.63 mm for average error) is lower than that of the Odometry-dominated methods, indicating more consistent performance across different trajectories. The Odometry method exhibits the highest standard deviation (9.20% for AEP and 57.13 mm for average error), reflecting its sensitivity to trajectory length and cumulative drift. In contrast, UWB shows a lower standard deviation than Odometry, as its error is bounded by the anchor coverage rather than accumulated over distance.

## VI. CONCLUSION

Mobile robots equipped with advanced sensors are finding increasing applications in various fields. Accurate indoor localization remains one of the most challenging tasks, as it directly impacts the efficiency and usability of autonomous mobile robots. In this work, we proposed a new adaptive algorithm that combines the reported coordinates from Odometry and UWB-based systems to estimate the location of mobile robots more accurately. We evaluated the accuracy of the proposed method by comparing it with Odometry and UWB-based localization methods while tracking mobile robots along various complex trajectories.

The experiments demonstrated that the UWB method is generally more reliable than the Odometry method. On average, the coordinates generated by UWB exhibit 19.21% higher accuracy compared to the Odometry method. Integrating UWB with Odometry provides a more accurate approach that addresses the limitations of individual localization techniques. A weighted averaging of UWB and Odometry methods improved the localization accuracy of the Odometry and UWB methods by 23.23% and 4.96%, respectively. In contrast, the proposed adaptive method achieved 19.05%, 33.59%, and 13.06% lower AEP compared to the UWB, Odometry, and 50-50 combinational methods, respectively. Among the combinational methods, O20-U80 achieved the best performance, while O50-U50 and O80-U20 improved over Odometry by 13.06% and 8.93%, respectively. These results confirm that a higher UWB contribution in the fusion leads to better localization accuracy. Lastly, the experiments confirmed that WiFi and BLE-based methods exhibit higher error margins than both the proposed adaptive method and other UWB-based methods. Specifically, the average error of the WiFi-based system is 42 times higher and the BLE-based system is 23.7 times higher than the proposed adaptive algorithm.

As the future work, we investigate how this adaptive localization method can be empowered with the multi-agent based coordination, planning and interaction towards the formalization of an autonomous cyber-physical system (CPS) [30], [31]. Mobile robots used in this study will be equipped with various autonomous software components to provide real-time proactive planning for more accurate coordination with other peers. These required proactive plans can be designed according to the well-known Belief-Desire-Intention (BDI) agent architecture [32] which we also adopted for mobile agent and robotic system implementations [33]. To facilitate the design and implementation emerging from the complicated nature of autonomy, distributedness and openness of such CPS as well as deploying BDI entities on Kobuki robots, our plan is to benefit from the system modeling methods and tools [34], [35] that we previously used for the creation of various autonomous agent systems with varying capabilities and complexities on the industrial scale. Furthermore, we plan to evaluate the effect of different anchor configurations, including varying the number and placement of anchors, on the localization accuracy of the proposed adaptive method. Additionally, a formal analysis of the latency, computational load, and deployment scalability of the proposed method can be conducted in the future work.

## REFERENCES


[1] C. Wu, Z. Gong, B. Tao, K. Tan, Z. Gu, and Z.-P. Yin, "RF-SLAM: UHF-RFID based simultaneous tags mapping and robot localization algorithm for smart warehouse position service," *IEEE Trans. Ind. Informat.*, vol. 19, no. 12, pp. 11765–11775, Dec. 2023.

[2] M. Khoshrangbaf, V. K. Akram, M. Challenger, and O. Dagdeviren, "An experimental evaluation of indoor localization in autonomous mobile robots," *Sensors*, vol. 25, no. 7, p. 2209, Mar. 2025.

[3] N. Ganganath and H. Leung, "Mobile robot localization using odometry and Kinect sensor," in *Proc. IEEE Int. Conf. Emerg. Signal Process. Appl.*, Jan. 2012, pp. 91–94.

[4] R. Reis, J. Mendes, F. N. Dos Santos, R. Morais, N. Ferraz, L. Santos, and A. Sousa, "Redundant robot localization system based in wireless sensor network," in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competitions (ICARSC)*, Apr. 2018, pp. 154–159.

[5] A. Loganathan, N. Ahmad, and P. Goh, "Self-adaptive filtering approach for improved indoor localization of a mobile node with zigbee-based RSSI and odometry," *Sensors*, vol. 19, no. 21, p. 4748, Nov. 2019.

[6] K. Weinmann and S. Simske, "Design of Bluetooth 5.1 angle of arrival homing controller for autonomous mobile robot," *Robotics*, vol. 12, no. 4, p. 115, Aug. 2023.

[7] M. L. Rodrigues, L. F. M. Vieira, and M. F. M. Campos, "Mobile robot localization in indoor environments using multiple wireless technologies," in *Proc. Brazilian Robot. Symp. Latin Amer. Robot. Symp.*, Oct. 2012, pp. 79–84.

[8] Z. Yan, W. Guan, S. Wen, L. Huang, and H. Song, "Multirobot cooperative localization based on visible light positioning and odometer," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–8, 2021.

[9] O. Oumar, T. Sattar, and M. O. Tokhi, "Indoor localization of mobile robots with wireless sensor network based on ultra wideband using experimental measurements of time difference of arrival," in *Proc. CLAWAR*, 2018, pp. 1–13.

[10] K. Guo, X. Li, and L. Xie, "Ultra-wideband and odometry-based cooperative relative localization with application to multi-UAV formation control," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2590–2603, Jun. 2020.

[11] V. Magnago, P. Corbalán, G. P. Picco, L. Palopoli, and D. Fontanelli, "Robot localization via odometry-assisted ultra-wideband ranging with stochastic guarantees," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1607–1613.

[12] M. Hamer and R. D'Andrea, "Self-calibrating ultra-wideband network supporting multi-robot localization," *IEEE Access*, vol. 6, pp. 22292–22304, 2018.

[13] J. Lima and P. Costa, "Ultra-wideband time of flight based localization system and odometry fusion for a scanning 3 DoF magnetic field autonomous robot," in *Proc. Iberian Robot. Conf.*, vol. 1, 2018, pp. 879–890.

[14] T.-M. Nguyen, A. H. Zaini, C. Wang, K. Guo, and L. Xie, "Robust target-relative localization with ultra-wideband ranging and communication," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2312–2319.

[15] S. Zhang, X. Tan, and Q. Wu, "Self-positioning for mobile robot indoor navigation based on wheel odometry, inertia measurement unit and ultra wideband," in *Proc. 5th Int. Conf. Vis., Image Signal Process. (ICVISP)*, Dec. 2021, pp. 105–110.

[16] Z. Cao, R. Liu, C. Yuen, A. Athukorala, B. K. K. Ng, M. Mathanraj, and U.-X. Tan, "Relative localization of mobile robots with multiple ultra-WideBand ranging measurements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 5857–5863.

[17] C. Wang, H. Zhang, T.-M. Nguyen, and L. Xie, "Ultra-wideband aided fast localization and mapping system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1602–1609.

[18] T. H. Nguyen, T.-M. Nguyen, and L. Xie, "Tightly-coupled single-anchor ultra-wideband-aided monocular visual odometry system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 665–671.

[19] J. González, J. L. Blanco, C. Galindo, A. Ortiz-De-Galisteo, J. A. Fernández-Madrigal, F. A. Moreno, and J. L. Martínez, "Mobile robot localization based on ultra-wide-band ranging: A particle filter approach," *Robot. Auto. Syst.*, vol. 57, no. 5, pp. 496–507, May 2009.

[20] Y. Liu and Y. Song, "A robust method of fusing ultra-wideband range measurements with odometry for wheeled robot state estimation in indoor environment," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 1269–1274.

[21] F. J. Perez-Grau, F. Caballero, L. Merino, and A. Viguria, "Multi-modal mapping and localization of unmanned aerial robots based on ultra-wideband and RGB-D sensing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 3495–3502.

[22] R. Liu, Z. Deng, Z. Cao, M. Shalihan, B. P. L. Lau, K. Chen, K. Bhowmik, C. Yuen, and U.-X. Tan, "Distributed ranging SLAM for multiple robots with ultra-WideBand and odometry measurements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 13684–13691.

[23] S. Romaniuk, A. Wolniakowski, A. Pawłowski, and C. Kownacki, "Adaptation of ultra wide band positioning system for adaptive Monte Carlo localization," in *Proc. 26th Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2022, pp. 238–243.

[24] H.-Y. Lin and M.-C. Yeh, "Drift-free visual SLAM for mobile robot localization by integrating UWB technology," *IEEE Access*, vol. 10, pp. 93636–93645, 2022.

[25] V. Brunacci, A. De Angelis, G. Costante, and P. Carbone, "Development and analysis of a UWB relative localization system," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[26] S.-H. Bach, P.-B. Khoi, and S.-Y. Yi, "Global UWB system: A high-accuracy mobile robot localization system with tightly coupled integration," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16618–16626, May 2024.

[27] M. Aydemir, M. Khoshrangbaf, V. K. Akram, and M. Challenger, "A mapping and monitoring platform for mobile robots," in *Proc. Int. Congr. Hum.-Comput. Interact., Optim. Robotic Appl. (HORA)*, May 2024, pp. 1–5.

[28] A. Moradbeikie, R. Azevedo, C. Jesus, and S. I. Lopes, "RSSI-based localization in industrial environments: A Wi-Fi/BLE hybrid approach," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2024, pp. 1–6.

[29] T. He, Y.-P. Zhang, M.-Z. Yan, W.-P. Luo, and S.-B. Ma, "A hierarchical DV-hop localization algorithm based on RSSI with self-localization weighted offset correction," *Ad Hoc Netw.*, vol. 173, Jun. 2025, Art. no. 103832.

[30] M. Yemini, A. Nedić, A. J. Goldsmith, and S. Gil, "Resilient distributed optimization for multiagent cyberphysical systems," *IEEE Trans. Autom. Control*, vol. 70, no. 6, pp. 3952–3967, Jun. 2025.

[31] M. Challenger, B. T. Tezel, V. Amaral, M. Goulao, and G. Kardas, "Agent-based cyber-physical system development with SEA_ML++," in *Multi-Paradigm Modelling Approaches for Cyber-Physical Systems*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 195–219.

[32] L. B. Becker, I. De Oliveira Silvestre, J. F. Hübner, and M. Fisher, "An expedited BDI agent architecture: Improving the responsiveness of agent-based autonomous systems for handling critical situations," *Robot. Auto. Syst.*, vol. 186, Apr. 2025, Art. no. 104917.

[33] G. Kardas, B. T. Tezel, and M. Challenger, "Domain-specific modelling language for belief–desire–intention software agents," *IET Softw.*, vol. 12, no. 4, pp. 356–364, Aug. 2018.

[34] O. F. Alaca, B. T. Tezel, M. Challenger, M. Goulão, V. Amaral, and G. Kardas, "AgentDSM-eval: A framework for the evaluation of domain-specific modeling languages for multi-agent systems," *Comput. Standards Interfaces*, vol. 76, Jun. 2021, Art. no. 103513.

[35] B. T. Tezel and G. Kardas, "Debugging in the domain-specific modeling languages for multi-agent systems," *J. Comput. Lang.*, vol. 83, Jun. 2025, Art. no. 101325.


**MINA KHOSHRANGBAF** received the B.Sc. degree from the Department of Mathematics, Payame Nour University, and the M.Sc. degree in computer sciences from Islamic Azad University. She is currently pursuing the Ph.D. degree with the International Computer Institute, Ege University, İzmir, Türkiye. Her research interests include autonomous mobile robots and swarm intelligence.

**MURAT AYDEMIR** received the bachelor's degree in electrical engineering from Koç University, Istanbul, Türkiye, and the master's degree in information technologies from Ege University, İzmir, Türkiye. He is an Embedded Software Engineer at Analog Devices Inc. His research interests include autonomous mobile robots and software systems.

**NUSIN AKRAM** received the B.S. degree in computer engineering from Azad University, in 2011, and the M.S. degree in IT from Ege University, in 2021, where she is currently the Ph.D. degree in IT with the International Computer Institute. Her research interests include drone networks, the Internet of Things, and distributed algorithms.

**GUL BOZTOK ALGIN** received the B.Sc. degree in computer engineering from Dokuz Eylul University, İzmir, Türkiye, in 2004, and the M.Sc. and Ph.D. degrees in computer science from the International Computer Institute, Ege University, İzmir, in 2007 and 2017, respectively. She has been with the International Computer Institute, Ege University, since 2005. She is currently working on communication networks and the Internet of Things.

**GEYLANI KARDAS** received the B.Sc. degree in computer engineering and the M.Sc. and Ph.D. degrees in information technologies from Ege University, Türkiye, in 2001, 2003, and 2008, respectively. He is a Full Professor with the International Computer Institute, Ege University. His research interests include agent-oriented software engineering, model-driven engineering, domain-specific (modeling) languages, and low-code software development.

**VAHID KHALILPOUR AKRAM** received the B.Sc. and M.Sc. degrees in computer engineering from Azad University, and the Ph.D. degree from the International Computer Institute, Ege University, İzmir, Türkiye. He is currently an Associate Professor with the International Computer Institute, Ege University, İzmir. His research interests include wireless networks, distributed algorithms, and mobile robots.