electronics

MDPI

Article

Bluetooth Low Energy-Based Docking Solution for Mobile Robots

Kyuman Lee

School of Space Engineering Sciences and Department of Robot and Smart System Engineering, Kyungpook National University, Daegu 41566, Republic of Korea; klee400@knu.ac.kr

Abstract

Existing docking methods for mobile robots rely on a LiDAR sensor or image processing using a camera. Although both demonstrate excellent performance in terms of sensing distance and spatial resolution, they are sensitive to environmental effects, such as illumination and occlusion, and are expensive. Some environments or conditions require low-power, low-cost novel docking solutions that are less sensitive to the environment. In this study, we propose a guidance and navigation solution for a mobile robot to dock into a docking station using the values of the angle of arrival and received signal strength indicator between the mobile robot and the docking station, measured via wireless communication based on Bluetooth low energy (BLE). This proposed algorithm is a LiDAR- and camera-free docking solution. The proposed algorithm is used to run an actual mobile robot and BLE transceiver hardware, and the obtained result is significantly close to the ground truth for docking.

Keywords: Bluetooth low energy; wireless communication; docking station; mobile robot; navigation

1. Introduction

With the advancement of robotic technologies, the application of mobile robots in industrial automation, hospital operations, agriculture, home automation, space exploration, military, and search and rescue has increased [1]. To perform these complex tasks in uncooperative environments, robots are required to operate for an extended period. To achieve this, automatic recharging is one of the most viable methods, and it is commonly performed at docking stations. Moreover, docking stations are important for the payload transfer or overhauling of robots during or after missions. Studies in this field are commonly referred to as auto docking in robotics [1–3]. For a successful docking procedure, a mobile robot should be capable of accurately localizing a docking station and of autonomously navigating to the docking station under various conditions. In addition, it should be capable of performing this task efficiently, as otherwise its operation will be compromised via the consumption of the remaining battery power, which may adversely affect the longevity of the power sources of the robot.

Outdoor robotic navigation systems work well under an adequate GPS signal range. However, for docking in an indoor environment, GPS signals are not adequately strong or precise [4]. Modern indoor robots are well equipped with simultaneous localization and mapping (SLAM) technologies to enable their navigation to a known docking station. Additionally, vision-based tracking methods, such as a visual marker [5,6], which exhibit high accuracy for short-range precise docking, have been employed in robots. However, in

Check for updates

Academic Editor: Giuseppe Prencipe

Received: 25 December 2025
Revised: 19 January 2026
Accepted: 21 January 2026
Published: 22 January 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

addition to the high computational cost of these visual methods, achieving a good tracking performance under ill-illuminated conditions is challenging. These vision-based passive-fashion approaches cannot localize docking stations that are located in obscured, unknown areas or moving in real time. Alternatively, indoor localization based on radio frequency (RF), Wi-Fi, ultrasonic, or infrared (IR) has emerged as a potential solution to address this problem. However, certain factors have limited the further application of these methods. For example, IR-based localization exhibits limited line of sight and a short detection range; ultrasonic-based systems [7] generally require expensive hardware; and the power consumption of Wi-Fi-based systems is higher than that of RF-based localization systems. Additionally, the deployment cost of ultrasonic- and Wi-Fi-based systems is higher than that of RF-based systems. Although RF identification (RFID)-based localization methods [8] have been employed, they suffer from a lack of robustness due to signal reflections and power absorption by obstacles and bodies.

Bluetooth, an RF technology, has demonstrated great promise in medium- to short-range localization-based applications over the years. The Bluetooth beacons provided by Bluetooth low energy (BLE) technology [9] are cheap and scalable. Moreover, Bluetooth transceivers are present in almost all modern computer devices, even in personal phones. Most Bluetooth-related location services use the received signal strength indicator (RSSI), which is a measure of the link quality, and it is used for estimating the distance between a transmitter and a receiver. With the aid of multiple of these link-based techniques, such as trilateration, a robot can relatively localize itself with reasonable accuracy [10]. However, there are inherent limitations to the accuracy of this approach, such as its inability to position multiple transmitters or receivers under certain conditions in some environments.

New capabilities that support high-accuracy direction finding have been introduced in the Bluetooth core specification version 5.x or higher [11]. The BLE 5.x operates in the 2.4 to 2.41 GHz band and divides this band into 40 channels, with a width of 2 MHz each, to generate a special direction-finding signal. By utilizing an antenna array at the receiver side, it can estimate the angle of arrival (AoA) by calculating the phase difference and the amplitude (e.g., RSSI) of the received signal at each antenna. This method is very easy and cheap to deploy, uses a simple algorithm, and exhibits low energy consumption. This study proposes the deployment of this concept, based on the use of the RSSI and AoA data acquired by an antenna array consisting of three 2.4 GHz dipole antennas [12], to achieve the accurate localization of a docking station and navigation of a mobile robot towards the docking station in an indoor environment.

The rest of this article is organized as follows. Related works are reviewed in Section 2, after which the proposed methodology is presented in Section 3. Section 4 presents a comprehensive summary of the results of the experiments conducted, and the conclusion and future work are discussed in Section 5.

2. Related Work

Seco et al. [13] conducted a survey on the mathematical methods for indoor localization and positioning systems, with a focus on RF-based applications. They categorized these methods into four types: geometry-based methods, minimization of the cost function, fingerprinting, and Bayesian techniques. Additionally, they provided further details on the applicability, requirements, and immunity of each method. Shang et al. [14] proposed two RSSI-based localization mechanisms for wireless sensor networks. To demonstrate their application, they gathered a vast amount of experimental data and determined the distance for specific RSSI values using Gaussian fitting and a novel interpolation method. Furthermore, a method was devised to deduce whether an unknown node was internal or external. The proposed method outperformed the existing approaches at the time in

terms of location accuracy. To improve the applicability of original RSSI-based lateration methods, Yang et al. [15] proposed regression-based and correlation-based approaches for RSSI-based lateration methods. They confirmed the improved performance of the proposed method over standard methods through simulation and real-world experiments using Wi-Fi-based applications. Huang et al. [8] proposed an RFID-based indoor localization system with a Kalman filter, Heron-bilateration LE, and novel post-processing techniques for drift removal. The system was cost-effective and scalable. Older versions of Bluetooth did not provide AoA (angle of arrival) measurement capabilities, and while position estimation in self-localization based solely on RSSI (received signal strength indicator) without AoA is possible, estimating orientation is not. For docking, an algorithm that aligns the orientation is also necessary.

Heydon et al. [9] introduced BLE technology, and Siekkinen et al. [16] conducted a study on BLE and confirmed the extremely small energy consumption of BLE and its very attractive ratio of energy per bit transmitted. Bruno et al. [17] proposed and analyzed an indoor tracking system based on Bluetooth technology. The proposed method was integrated into the infrastructure-based network of Bluetooth access points of the building, and it was confirmed to be a cost-effective solution. Danis et al. proposed a novel localization method based on BLE [18] using an observation model specifically tailored for RSSI fingerprints (a combination based on the optimal transport model of the Wasserstein distance). The model was used to obtain accurate estimation from noisy Bluetooth data. Park et al. [19] proposed a hybrid method that utilized BLE and low frequency (LF) for indoor localization in limited-battery-capacity mobile devices. Additionally, they proposed a situational awareness system to increase robustness. Although this method achieved good results, its accuracy was in the meter range. Previous researches have mostly focused on self-localization using BLE. There has been little research on target tracking or navigation using BLE, and this study proposes a novel method for robot docking using BLE.

The earliest study on RF-based AoA localization attempts was conducted by Kim et al. [20]. They estimated the AoA using the ratio of the received strength between two antennas and conducted simulation and real-world experiments under various conditions. Although reasonably good results were obtained, the results were adversely affected by signal distortion. Yin et al. [21] proposed a method for estimating AoA using the channel state estimation data from Wi-Fi chips. Additionally, they created a virtual antenna array without any hardware adjustment using affine propagation clustering. This method achieved meter-level localization accuracy in real-world indoor experiments. Most of the aforementioned studies focused on self-localization and not on target (i.e., docking station) tracking. Furthermore, they assumed that the location of a docking station was fixed or known in advance, and the results of previous studies were not sufficiently accurate for docking.

3. Methodology

To propose a BLE-based docking solution in this study, we designed the following two stages: (1) the mission stage and (2) the docking stage. During the mission stage, a mobile robot visits desired locations to perform the given missions, which depend on the field (e.g., logistics) of application of the mobile robots. For automatic recharging, payload transfer, or overhauling of the robot during or after missions, the robot requires a stage transition from the mission stage to the docking stage. That is, when homing at the mission stage, a stage transition occurs when the strength of the RSSI signal is less than or equal to the RSSI transition threshold. Next, during the docking stage, the mobile robot executes the proposed docking algorithm to guide it to a docking station based on the measured AoA and RSSI values.

The overall architecture of the proposed docking solution is shown in Figure 1. A pre-map is only constructed for the mission zone by one of the open-source 2D SLAM algorithms [22], and the docking zone is an unknown environment without any pre-map. During the mission stage, a user sets mission locations as target points in the pre-map, after which they are sent to the navigation stack. This navigation stack includes obstacle avoidance and path planning functionality. The basic implementations of the path planner are both the Trajectory Rollout and Dynamic Window Approach (DWA) [23] algorithms for local robot navigation on a plane. Given a plan to follow and a costmap, the controller produces velocity commands to send to a mobile base.

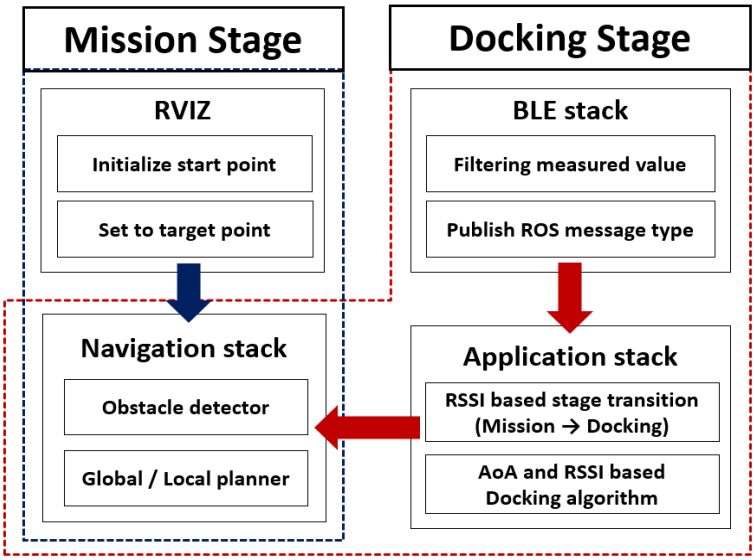


Figure 1. Overall architecture of the Bluetooth low energy (BLE)-based docking solution.

The BLE stack built at the docking station filters the AoA and RSSI measurements and publishes the filtered values. The application stack on the mobile robot examines the filtered RSSI signals to determine the occurrence of a stage transition. Once the mission stage transitions to the docking stage, the proposed docking algorithm is implemented based on the filtered AoA and RSSI values. Here, the navigation stack is shared in both stages. Each stage sends only the next target points to the navigation stack to navigate to certain mission locations or the docking station.

3.1. Mission Stage

In this study, the mission stage where the mobile robot performs given missions (e.g., visiting specific locations) was defined using the open-source SLAM package [22]. This SLAM package creates a pre-map in the mission zone using a LiDAR sensor mounted on the mobile robot. This study also proposes how existing SLAM packages work in our docking solution.

An example of the pre-map is shown in Figure 2. The white floor was assumed to have no obstacles, whereas the black dots (or black lines) represent obstacles, such as walls. Gray regions, including the docking zone, represent unmapped areas obscured by obstacles. In Figure 2, the (1) starting point is the home location of the mobile robots. During the mission stage, the mobile robot moves to the (2) mission points to perform certain missions, and when the missions end, or auto docking is required, it returns to (1). Additionally, while homing, the mobile robot passes the unmapped docking zone centered at the (3) docking station, and the mission stage transitions to the docking stage.

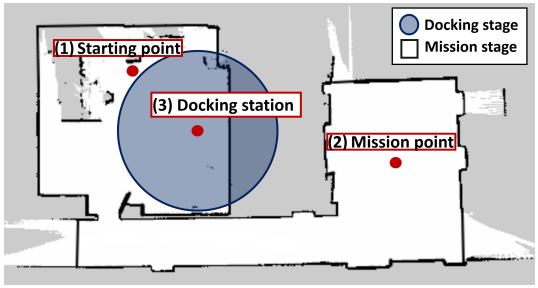


Figure 2. Example of the pre-map.

3.2. Docking Stage

The overall docking process is illustrated in Figure 3. When homing to the starting point during the mission stage, the mission stage transitions to the docking stage if the filtered RSSI value is less than the preset RSSI transition threshold. Otherwise, the mobile robot keeps moving to the starting point, as the starting point is a known location in the pre-map. During the docking stage, if the filtered RSSI value is less than the preset RSSI arrival threshold while docking, the docking movement is assumed to be completed. Here, an RSSI value is used for two criteria regardless of what stage the robot is in.

$$\hat{RSSI}_j = \frac{1}{N}\sum_{i=0}^{N-1}\widetilde{RSSI}_{j-i} \tag{1}$$

$$\hat{AoA}_j = \frac{1}{N}\sum_{i=0}^{N-1}\widetilde{AoA}_{j-i} \tag{2}$$

The measured $\widetilde{RSSI}$ and $\widetilde{AoA}$ raw values (blue lines) are relatively noisy (Figure 4), which might be caused by the limitations of the BLE transceiver hardware or vibrations caused by the movement of the mobile robot. The docking movement of the mobile robot changes significantly when $\widetilde{RSSI}$ and $\widetilde{AoA}$ are noisy. In this study, these two measurements are filtered by taking the average value of each N stored measurements in Equations (1) and (2). In other words, the j-th filtered $\hat{RSSI}$ and $\hat{AoA}$ values are the average value of each stacking N measurement from the $j-N-1$-th to the j-th on the stack. To keep the most recent N data in the stack, the oldest measurement is popped when a new measurement arrives. This moving average seems like a very primitive method, but Figure 4 illustrates the acceptable filtering results of the RSSI and AoA values (orange lines).

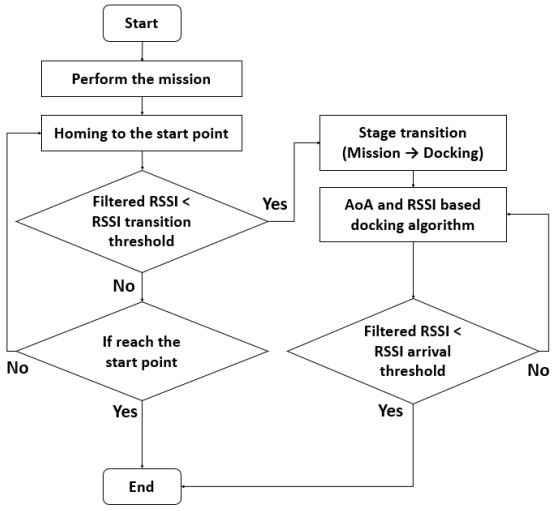
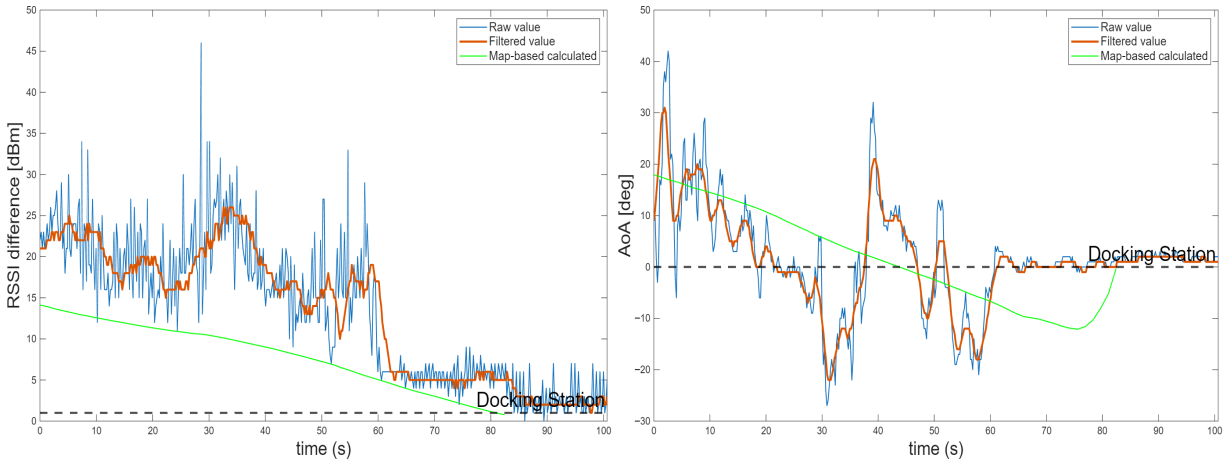


Figure 3. Flowchart of the BLE-based docking solution.



(**a**) RSSI (**b**) AoA

Figure 4. Measured, filtered, and calculated values of signal strength indicator (RSSI) and angle of arrival (AoA).

A mobile robot stops in front of the docking station by comparing the current $\hat{RSSI}$ value and the preset RSSI arrival threshold, $RSSI_{arrival}$. As shown in Figure 5, this is defined as the RSSI value at a certain boundary (i.e., the circle with a blue dotted line) centered at the docking station, which is the threshold for assuming that the docking operation is complete. Here, the constant $RSSI_{arrival}$ is pre-measurable (e.g., an RSSI value at a point 1 m away) in real-world applications. Furthermore, we define the RSSI difference, $RSSI_{diff}$, as the difference between the $\hat{RSSI}$ obtained at a certain location and $RSSI_{arrival}$. As $RSSI_{diff}$ is close to zero, it indicates entry into the boundary of the docking station and successful docking. The $x-$ and $y-$axes in the world coordinate system are also defined in Figure 5.

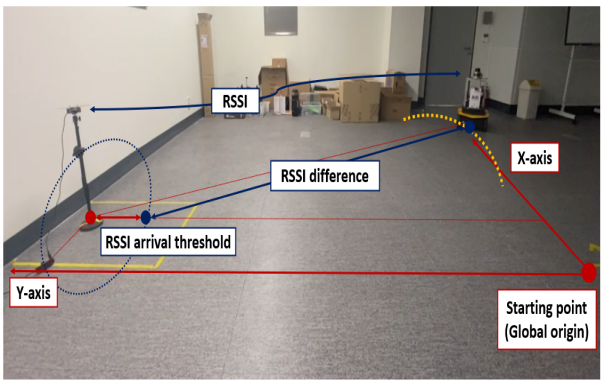


Figure 5. Definition of the preset RSSI arrival threshold.

This section presents our docking solution based on the filtered AoA and RSSI values at the docking stage. At timestamp k, the position and orientation of the mobile robot in the local coordinates $(x_k, y_k)^\top$ and θ_k, respectively. This odometry information is a localization estimate generated by processing data from the wheel encoders and integrated inertial measurement unit (IMU) using an extended Kalman filter (EKF). As mentioned in Section 3, the docking zone is typically an unknown environment without any pre-existing map, so we cannot know the global pose values in the world coordinate system. Instead, we can only obtain local pose values in the local coordinate system through the localization estimate. For precisely docking using the filtered $\hat{AoA}_k$ and $\hat{RSSI}_k$, the next goal position

$(x^g_{k+1}, y^g_{k+1})^\top$ and next goal orientation θ^g_{k+1} in the local coordinates can be calculated as expressed in Equations (3) and (4).

$$\theta^g_{k+1} = \theta_k + \Delta\theta_k, \tag{3}$$

$$\begin{pmatrix} x^g_{k+1} \\ y^g_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} + R_{wb,k} \begin{pmatrix} \Delta r_k \\ 0 \end{pmatrix}, \tag{4}$$

where

$$\begin{aligned} RSSI_{diff,k} &= RSSI_{arrival} - \hat{RSSI}_k \\ \Delta\theta_k &= G_a * RSSI_{diff,k} * (\hat{AoA}_k - \hat{AoA}_{k-1}) \\ \Delta r_k &= G_d * RSSI_{diff,k} \\ R_{wb,k} &= \begin{bmatrix} \cos(\theta_k + \Delta\theta_k) & -\sin(\theta_k + \Delta\theta_k) \\ \sin(\theta_k + \Delta\theta_k) & \cos(\theta_k + \Delta\theta_k) \end{bmatrix} \end{aligned}$$

The angle command, $\Delta\theta_k$, at which the mobile robot should rotate to dock to the station is calculated by multiplying the difference between the AoA values at timestamps k and $k-1$ by the tunable gain G_a and $RSSI_{diff,k}$. In fact, AoA is not solely determined by the relative orientation difference between the docking station and the robot's heading. It is a complex value determined by both the relative orientation difference and the bearing angle based on the relative position. Therefore, when calculating $\Delta\theta_k$, we do not use only the $\hat{AoA}_k$ value, but instead compute it using the historical change in the AoA measurements (i.e., $\hat{AoA}_k - \hat{AoA}_{k-1}$). In Equation (3), the next goal orientation θ^g_{k+1} that the mobile robot should rotate to is determined. Thereafter, the distance command, Δr_k, required for the mobile robot to reach the dock station is calculated by multiplying the tunable gain G_d and $RSSI_{diff,k}$. $R_{wb,k}$ is a rotation matrix that converts body-fixed coordinates to local coordinates. By multiplying Δr_k by $R_{wb,k}$, the next goal position $(x^g_{k+1}, y^g_{k+1})^\top$ that the mobile robot should go to is determined in Equation (4). That is, the desired orientation and position at the next timestamp, which are expressed in Equations (3) and (4), are sent to the navigation stack (Figure 1) for docking. This process is repeated until the $RSSI_{diff,k}$ is less than α.

The reason the mobile robot rotates first and then moves forward is shown in Figure 6. Case 1 of Figure 6 is a case of rotating after moving forward, whereas Case 2 is a case of moving forward after rotating. After the two movements, Case 2 is looking toward the docking station, whereas Case 1 is looking elsewhere. This implies that there is a problem with Case 1, in which the mobile robot does not face the docking station after two movements. If the transmitter on the mobile robot is looking elsewhere as in Case 1, the RSSI signal becomes weak, and the measured AoA value becomes inaccurate. To solve this problem, Case 2 (moving forward after rotating) is selected in our docking solution. In other words, $R_{wb,k}$ in Equation (4) is computed by θ^g_{k+1} (angle after rotation).

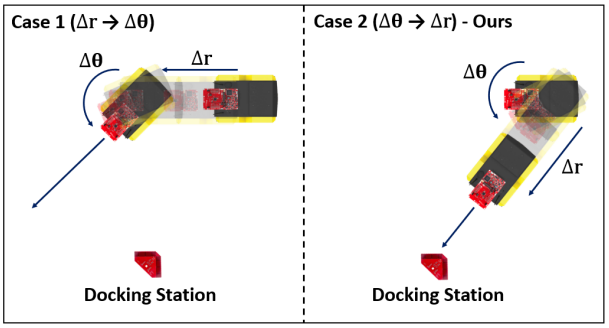


Figure 6. Difference in the outcome of the sequence of the two actions.

The pseudocode of docking using Equations (3) and (4) is shown in Algorithm 1. After initializing gains, thresholds, and parameters, the proposed docking algorithm is repeatedly performed until the RSSI difference value, $RSSI_{diff,k}$, exceeds α. A re-alignment process will be necessary if $RSSI_{diff,k}$ is less than β and $\hat{AoA}_k$ is greater than γ degrees. β is related to the distance between the docking station and the mobile robot, and γ is related to the angle. Sometimes, even when the mobile robot is almost close to the docking station, there is still a large angle difference. That is, β is small and γ is large. Moving forward in this state may result in the deviation of the mobile robot from the docking path. Therefore, a re-alignment process that only rotates the robot in place is needed. The re-alignment process refers to aligning the heading direction of the robot toward the direction of the docking station by rotating it at fixed δ degrees without a change in its position. Once the re-alignment is complement, the mobile robot calculates the next goal poses θ^g_{k+1} and $(x^g_{k+1}, y^g_{k+1})^\top$ using Equations (3) and (4) and moves to the corresponding goal. By iterating the above processes, if $RSSI_{diff,k}$ is less than α, the mobile robot arrives at the docking station, so Algorithm 1 is terminated. The proposed docking method seems like a conventional P-gain control, but it is practical because it uses noisy RSSI and AoA sensor streams and active-fashion navigation via wireless communication-based BLE.

Algorithm 1 Docking algorithm

Input: $\hat{AoA}_k$, $RSSI_{diff,k}$
Output: *isDock*

function GoToDockingStation
  while $RSSI_{diff,k} > \alpha$ **do**
    Update $\hat{AoA}_k, \hat{RSSI}_k$ *by BLE − based sensing*

    Re-align process:
    if $RSSI_{diff,k} < \beta$ *and* $\hat{AoA}_k > \gamma$ **then**
      $\Delta\theta_k \leftarrow fixed\ \delta$
      Calculate θ^g_{k+1} *using Equation* (3)
      Rotate to the next goal orientation θ^g_{k+1}
    end if

    $\Delta\theta_k \leftarrow G_a * RSSI_{diff,k} * (\hat{AoA}_k - \hat{AoA}_{k-1})$
    $\Delta r_k \leftarrow G_d * RSSI_{diff,k}$

    Calculate θ^g_{k+1} *using Equation* (3)
    Rotate to the next goal orientation θ^g_{k+1}

    Calculate $(x^g_{k+1}, y^g_{k+1})^\top$ *using Equation* (4)
    Move to the next goal location $(x^g_{k+1}, y^g_{k+1})^\top$

    $\hat{AoA}_{k-1} \leftarrow \hat{AoA}_k$
  end while
end function

4. Results

This section presents the performance of the proposed docking method on a real robot named Jackal in several experiments. This section describes experimental settings (e.g., parameter settings in the navigation stack and actual robot setup). In addition, it presents the results of docking performance and robustness by comparing the odometry values when using the proposed method value and the ground truth, respectively.

4.1. Parameter Settings

(1) Velocity and Acceleration Ranges: The navigation stack provided by the robot operating system (ROS) [24] is used to perform the missions in both stages. This navigation stack is important for mobile robots to move. To reduce the drift errors, it is essential to set the max/min velocity and acceleration correctly. For the experimental results, the translational velocity range is set to 0.1–0.25 m/s, and the rotation velocity is set to −0.52–0.52 rad/s. Thus, setting the maximum translational and rotational acceleration to 1.0 m/s^2 is appropriate. These in Table 1 are also limits for safety purposes.

(2) Constants: We have declared several constants and variables for the docking algorithm, shown in Table 1. First, the window size N for the moving average filter in Equations (1) and (2) is 10. Increasing this value would result in better filtering, but it is a trade-off that would increase the computational load or make it more difficult to respond quickly to changes. The RSSI arrival threshold is the RSSI value measured within a certain distance from the station where docking is considered complete (e.g., within a 1 m distance in front), and in this paper, it is set to −51 [dBm]. In addition, the G_a for calculating $\Delta\theta_k$ is set to 0.25 and the G_d for calculating Δr_k is set to 0.08. G_a and G_d are P-gain values that determine how far the robot moves forward and how much it rotates its heading at each time step during docking. Smaller values result in more precise docking but slower arrival, representing a trade-off. Further, α, a condition for repeating the while loop in Algorithm 1, is set to 1. A smaller α value would result in more iterations of the while loop, leading to more precise docking. However, iterations with too small α might prevent the loop from terminating, so an appropriate value should be tuned experimentally. β and γ, the criteria for the re-alignment process, are set to 8 and 20, respectively, and the angle δ rotated during the re-alignment process is set to 30. In other words, if the robot is close enough to the station according to β, but the AoA is still large enough according to γ, then it only needs to re-align its orientation in place by an amount determined by δ. Each of these values is determined experimentally by the user.

Table 1. Parameter settings used in this study.

Parameter	Value
Transitional velocity range	from 0.1 to 0.25 m/s
Rotational velocity range	from −0.52 to 0.52 rad/s
Maximum acceleration	1.0 m/s^2
G_a	0.25
G_d	0.08
α	1 dBm
β	8 dBm
γ	20°
δ	30°
N	10

4.2. Actual Robot Setup

To demonstrate the performance of the proposed docking algorithm, a mobile robot known as Jackal is used in this study (Figure 7). Jackal is a small and fast robotics research platform with an onboard computer known as the NVIDIA Jetson TX2, Santa Clara, CA, USA. We mounted an extra LiDAR named RPLIDAR-A2 on top of the mobile robot. We used this

LiDAR to create a pre-map (Section 3.1) in the mission zone. The LiDAR has an angular range of 360 degrees and a distance range of 0.12 to 12.0 m. In addition, we experimented with Velodyne's Puck, a 3D LiDAR (Figure 7). However, the disadvantage is that a large amount of load is required owing the unnecessarily large amount of point cloud data used in performing this scenario, so 2D LiDAR is ultimately selected. In addition, a transmitter for Bluetooth communication is attached to the front of the robot (red board). It is experimentally observed that the signal has less noise when the height is similar to that of the receiver attached to the unknown docking station. Therefore, we attach the transmitter to the box to remove the noise as much as possible. The proposed methodology is not limited to a specific robot or specific BLE sensors.

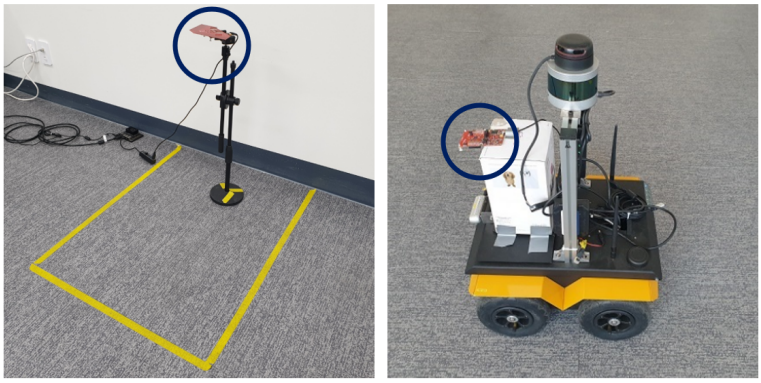


Figure 7. BLE receiver in the docking station (navy circle in left image) and BLE transmitter on the mobile robot (navy circle on right image).

4.3. Experimental Results

In this study, to test the proposed docking method, the docking station with a BLE receiver is located at $x = 0.4$ m, $y = 3.6$ m, $\theta = 90$ deg on the global map (Figure 5), but the robot does not know this location information. The details of the docking station, including the receiver, are shown in Figure 7. We assume that only a single robot platform is operated in the single indoor environment. Docking for multiple robots is out of the scope of this study since it is a scheduling or task assignment problem. Despite being a single indoor environment, the environment was cluttered, and the robot operated with obstacle avoidance capabilities provided by the navigation stack's obstacle detector and path planner (Figure 1). Obstacle avoidance itself in dynamic scenarios is also out of the scope of this study. Monte Carlo trials from various starting points are conducted to test for robustness and generalization to various layouts.

Based on the RSSI value of -73 [dBm], the transition from the mission stage to the docking stage occurs, and the proposed docking algorithm is automatically executed. This reference RSSI value is obtained experimentally by measuring the RSSI value at the boundary between the mission and docking stages we set up. This value may change depending on the field of application. Figure 8 shows the change in the RSSI value during the proposed docking process. The RSSI values separating the mission and docking stages are set to -73 [dBm] and -76 [dBm]. That is, when the RSSI value is higher than -73 [dBm], the docking stage is performed, and when the RSSI value is lower than -76 [dBm], the mission stage is performed. If the value is in between, the current stage is maintained. The RSSI value that divides the mission stage and the docking stage is set to a range (not a constant) because of the error in the RSSI value. Therefore, as shown in Figure 8, even if the RSSI value falls below -73 [dBm] after the docking algorithm is executed, the stage is not changed and docking is still performed. In addition, $RSSI_{arrival}$ is -51 [dBm], and the RSSI value approaches -51 [dBm] as the docking progresses.

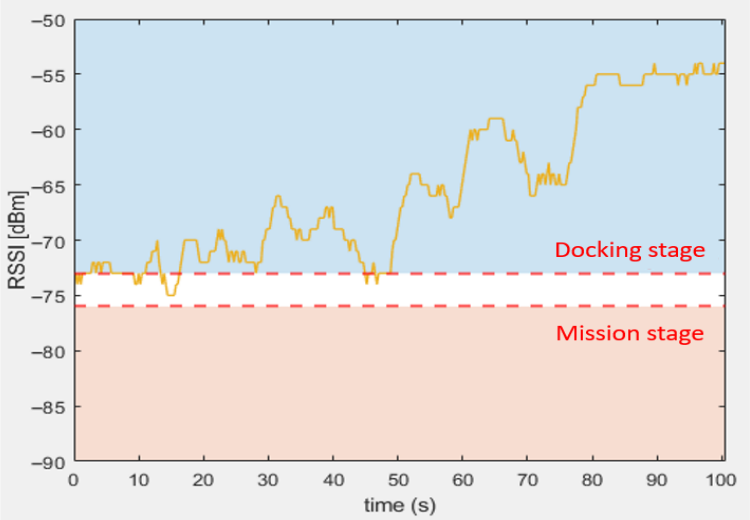


Figure 8. Stage transition from the mission stage to the docking stage based on the filtered RSSI value.

At the docking stage, Figure 9 shows the entire docking process by the proposed approach with an actual robot. As mentioned in Sections 3.2 and 4.1, the mobile robot docks almost straight to the docking station by fine-tuning each parameter. That is, it accurately drops within the desired docking area (yellow square) and stops precisely in front of the docking station based on the RSSI signal (i.e., when the RSSI signal reaches -51 [dBm]).

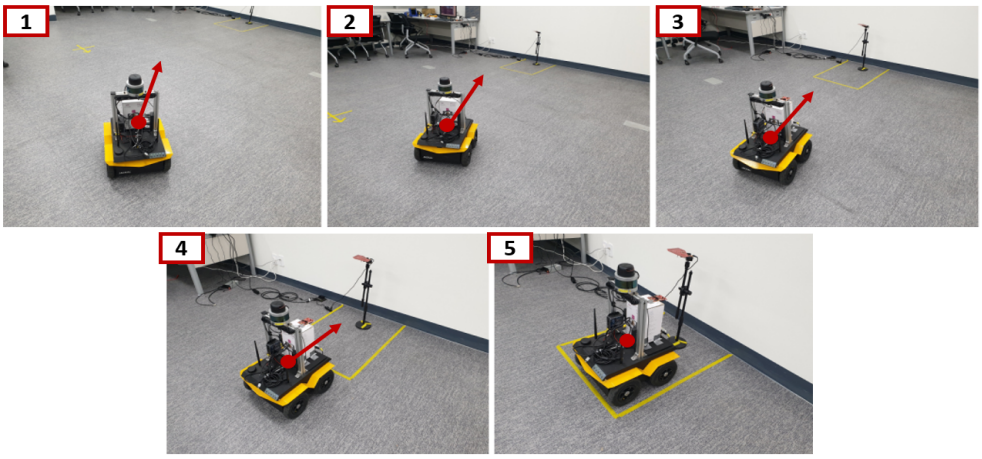
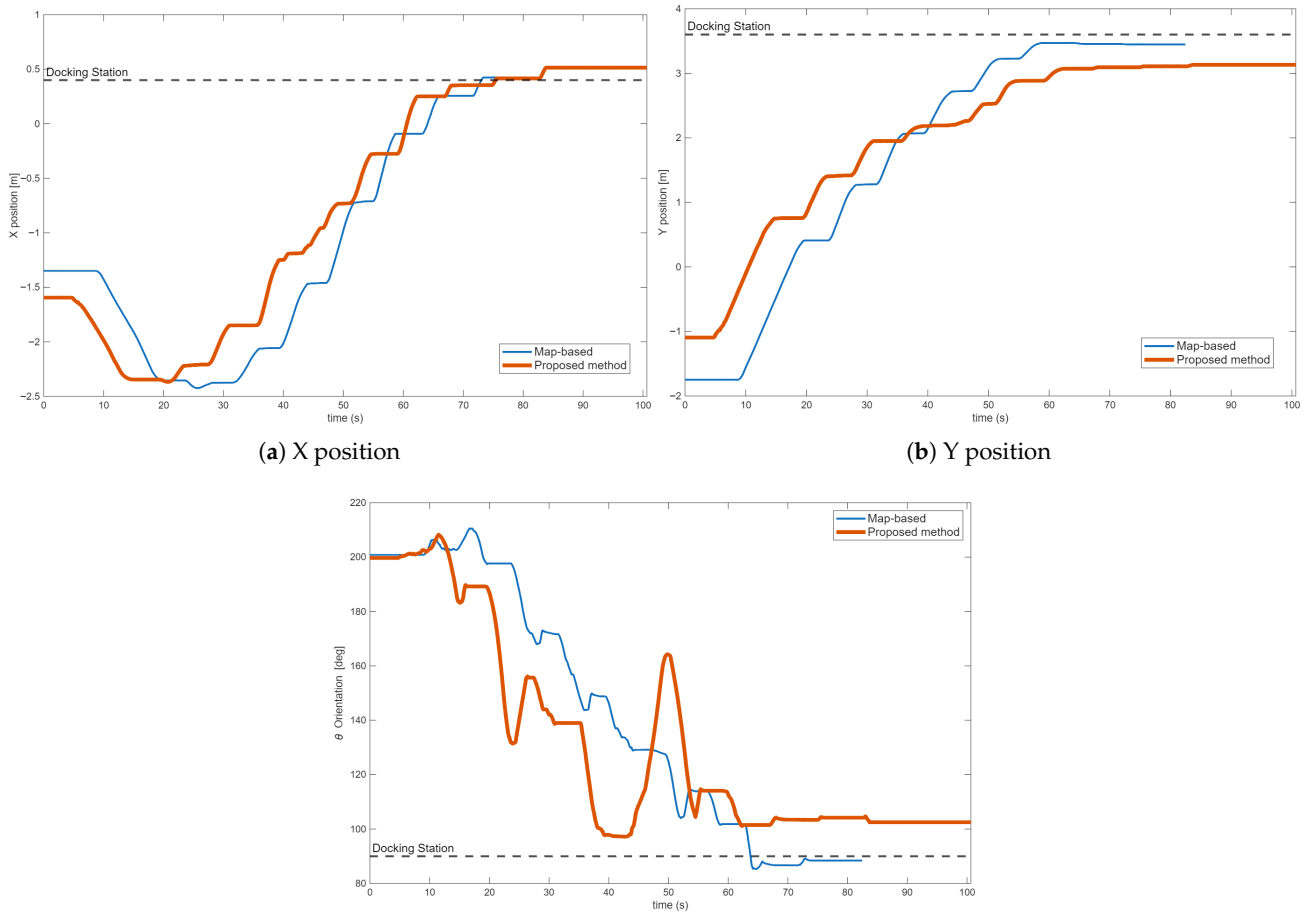


Figure 9. Docking process using the proposed docking solution.

Figure 4 shows the RSSI difference and filtered AoA value when docking to the docking station. As the proposed docking algorithm is executed, the RSSI difference approaches 1 (i.e., α), and the filtered AoA value approaches and converges to 0 (in steady state). The final values of RSSI and AoA in Table 2 show that the docking by our method relatively succeeds. For a comparison study, we run existing adaptive Monte Carlo localization (AMCL) [25] and calculate RSSI and AoA in the map. Green lines in Figure 4 represent their map-based calculated values. Since the AoA and RSSI values are filtered, they are bound to have acceptable errors compared to the map-based calculated references.

Figure 10 graphically illustrates the odometry values (orange line) when the proposed docking algorithm was implemented using filtered AoA and RSSI. For a comparison study, Figure 10 also illustrates the odometry values (blue line) when the same Equations (3) and (4) were implemented using calculated AoA and RSSI in the map (i.e., green line in Figure 4) instead of filtered ones. This "map-based" value is considered a reference close to the ground truth level. In fact, the docking zone lacks a map or is impossi-

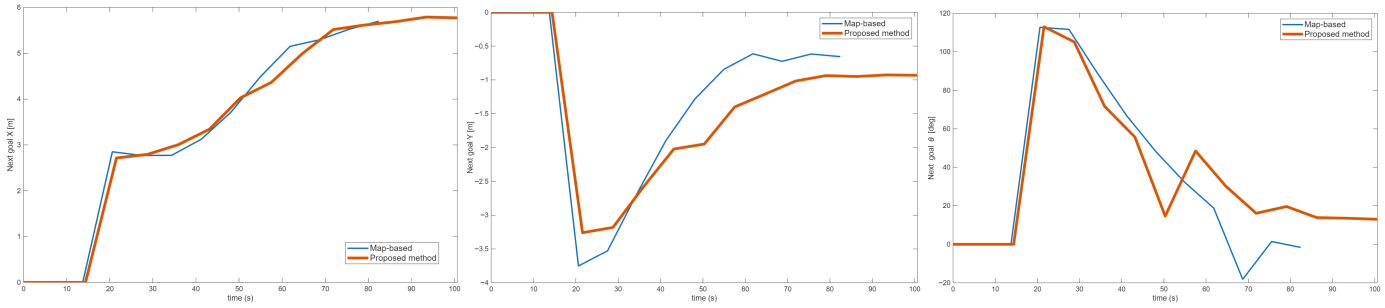
ble to map, so the "map-based" approach is not realistic or practical. The next goal position $(x^g_{k+1}, y^g_{k+1})^\top$ and orientation θ^g_{k+1}, computed by Equations (3) and (4) (or Algorithm 1), are given in Figure 11. All are the average values after performing the experiment for five trials (at a random starting point) using Monte-Carlo analysis. Despite the noisiness of the AoA and RSSI measurements, docking for all trials was successfully performed at the docking station (i.e., success rate is 100%), shown in Figure 10 and Table 2. Table 2 presents each final value after docking is complete. Bold in the table is used to highlight ours.



(**a**) X position (**b**) Y position

(**c**) θ angle

Figure 10. Trajectory of the mobile robot during docking.



(**a**) Next goal X (**b**) Next goal Y (**c**) Next goal θ

Figure 11. Next goal commands for docking.

Table 2. Final values after docking is complete.

	Map-Based	Ours	Docking Station
RSSI [dBm]	0.7654	**3**	1
AoA [deg]	−0.3141	**1**	0
X [m]	0.4245	**0.5149**	0.4
Y [m]	3.448	**3.1324**	3.6
θ [deg]	88.4162	**102.4912**	90

The first graph in Figure 10 shows the local x_k position of the mobile robot for each timestamp. When the x-axis position of the docking station is located at $x = 0.4$ m based on the global origin, the x position of the mobile robot can be observed to gradually approach 0.5149 m. The second graph in Figure 10 shows the local y_k position of the mobile robot for each timestamp. The y-axis position of the docking station is located at $y = 3.6$ m based on the global origin, and the y trajectory of the mobile robot successfully approaches 3.1324 m. Although the docking based on our approach arrives later compared to the reference method (i.e., map-based), our proposed algorithm reliably converges to the docking station (Table 2) within acceptable errors.

The last graph in Figure 10 shows the local heading angle θ_k of the mobile robot in the local coordinates at each timestamp. The antenna direction of the BLE receiver on the docking station is $\theta = 90$ deg in the world coordinate system, and the heading of the mobile robot aligns toward the docking station (in steady-state) as docking progresses. The final θ is 102.4912 deg. The closer a relative angle is to the mobile robot and the docking station, the more significant the change will be, even if the mobile robot moves slightly. Therefore, when the mobile robot approached the docking station, the angle value changed from a value to the left of 90 degrees to a value to the right of 90 degrees. Even if the relative AoA value is noisy, the proposed algorithm re-aligns the local headings of the mobile robot close to 90 degrees through the re-alignment process (Algorithm 1).

In evaluating docking algorithms for the Clearpath Jackal on an Nvidia Jetson TX2, the Cartographer SLAM and AMCL approach (i.e., "map-based") is high-intensity, consuming approximately 35–45 W due to the active power draw of LiDAR (8–18 W) and the heavy CPU/GPU load required for scan matching and particle filtering. In contrast, the Texas Instrument BLE (RSSI/AoA) method (i.e., ours) is significantly more efficient, consuming only 15–20 W, as BLE modules draw negligible milliwatt-scale power and the proposed docking algorithm requires minimal computational resources, allowing the TX2 to operate in a low-power mode. Consequently, while the LiDAR-based stack (Map-based) offers superior precision, the BLE-based docking (proposed) reduces power consumption by over 50%, effectively doubling the energy efficiency during the docking phase and extending the robot's operational battery life.

5. Discussion

In this paper, we propose the design of a mobile robot docking platform with an accurate, low-energy navigation system based on BLE using both RSSI and AoA measurements acquired by an antenna array consisting of three 2.4 GHz dipole antennas [12]. We conducted Monte Carlo experiments using a mobile robot, which transitions to navigation in the docking stage. This novel method was tested several times and demonstrated great performance in real-world scenarios. However, BLE still suffers from numerous inherent shortcomings, which more or less restrict its potential in indoor navigation. In future work, we can continue to improve the results with updated hardware. In addition, explicit signal

propagation or path loss models will be used for analyzing the impact of multipath effects or environmental factors on RSSI and AoA measurements.

Funding: This research was supported by the Regional Innovation System & Education (RISE) Glocal 30 program through the Daegu RISE Center, funded by the Ministry of Education (MOE) and the Daegu, Republic of Korea (2025-RISE-03-001).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors thank Namyoung Kim and Gibeom Kim for their help in the research.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Wardhana, D.A.P.; Happyanto, D.C.; Purwanto, E.; Akbar, G.E.A.; Putra, K.T. AutoDock-IPS: An Automated Docking for Mobile Robot Based on Indoor Positioning System. *J. Electr. Technol. UMY* **2021**, *5*, 32–39. [CrossRef]
2. Kim, N.; Lee, K. Bluetooth Low Energy-Based Guidance Algorithm for a Mobile Robot Docking. In Proceedings of the Institute of Control, Robotics and Systems (ICROS) 2023, Yeosu, Republic of Korea, 17–20 October 2023; pp. 1–2.
3. Yanyan, Z.; Chengren, L.; Jihong, X. The research of the docking application between industrial robot and machine tool. In Proceedings of the 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), Ottawa, ON, Canada, 29–30 September 2014; pp. 1272–1275.
4. Razavi, A.; Gebre-Egziabher, D.; Akos, D.M. Carrier loop architectures for tracking weak GPS signals. *IEEE Trans. Aerosp. Electron. Syst.* **2008**, *44*, 697–710. [CrossRef]
5. Garrido-Jurado, S.; Muñoz-Salinas, R.; Madrid-Cuevas, F.J.; Marín-Jiménez, M.J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* **2014**, *47*, 2280–2292. [CrossRef]
6. Sani, M.F.; Karimian, G. Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors. In Proceedings of the 2017 International Conference on Computer and Drone Applications (IConDA), Kuching, Malaysia, 9–11 November 2017; pp. 102–107.
7. Kim, S.Y.; Yoon, K.S.; Lee, D.H.; Lee, M.H. The localization of a mobile robot using a pseudolite ultrasonic system and a dead reckoning integrated system. *Int. J. Control. Autom. Syst.* **2011**, *9*, 339. [CrossRef]
8. Huang, C.H.; Lee, L.H.; Ho, C.C.; Wu, L.L.; Lai, Z.H. Real-time RFID indoor positioning system based on Kalman-filter drift removal and Heron-bilateration location estimation. *IEEE Trans. Instrum. Meas.* **2014**, *64*, 728–739. [CrossRef]
9. Heydon, R. *Bluetooth Low Energy*; Pearson: London, UK, 2012.
10. Upadhyay, J.; Rawat, A.; Deb, D.; Muresan, V.; Unguresan, M.L. An RSSI-based localization, path planning and computer vision-based decision making robotic system. *Electronics* **2020**, *9*, 1326. [CrossRef]
11. Woolley, M. *Bluetooth Direction Finding, A Technical Overview*; Technical Report; Bluetooth SIG: Kirkland, WA, USA, 2021. Available online: https://www.bluetooth.com/bluetooth-resources/bluetooth-direction-finding/ (accessed on 1 January 2026).
12. Texas Instruments. *CC2640R2 LaunchPad™ Development Kit*, Rev. 3; Texas Instruments: Dallas, TX, USA, 2021. Available online: https://www.ti.com/tool/LAUNCHXL-CC2640R2 (accessed on 1 January 2026).
13. Seco, F.; Jimenez, A.R.; Prieto, C.; Roa, J.; Koutsou, K. A survey of mathematical methods for indoor localization. In Proceedings of the 2009 IEEE International Symposium on Intelligent Signal Processing, Budapest, Hungary, 26–28 August 2009; pp. 9–14. [CrossRef]
14. Shang, F.; Su, W.; Wang, Q.; Gao, H.; Fu, Q. A location estimation algorithm based on RSSI vector similarity degree. *Int. J. Distrib. Sens. Netw.* **2014**, *10*, 371350. [CrossRef]
15. Yang, J.; Chen, Y. Indoor localization using improved rss-based lateration methods. In Proceedings of the GLOBECOM 2009-2009 IEEE Global Telecommunications Conference, Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–6.
16. Siekkinen, M.; Hiienkari, M.; Nurminen, J.K.; Nieminen, J. How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4. In Proceedings of the 2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Paris, France, 1 April 2012; pp. 232–237.
17. Bruno, R.; Delmastro, F. Design and analysis of a bluetooth-based indoor localization system. In Proceedings of the IFIP International Conference on Personal Wireless Communications, Venice, Italy, 23–25 September 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 711–725.
18. Daniş, F.S.; Cemgil, A.T. Model-Based Localization and Tracking Using Bluetooth Low-Energy Beacons. *Sensors* **2017**, *17*, 2484. [CrossRef] [PubMed]

19. Park, J.K.; Kim, J.; Kang, S.J. A situation-aware indoor localization (SAIL) system using a LF and RF hybrid approach. *Sensors* **2018**, *18*, 3864. [CrossRef] [PubMed]
20. Kim, M.; Kim, H.W.; Chong, N.Y. Automated robot docking using direction sensing RFID. In Proceedings of the Proceedings 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 4588–4593.
21. Yin, L.; Wang, Z.; Deng, Z.; Jiang, T.; Sun, Y. Angle-of-Arrival Positioning System Based on CSI Virtual Antenna Array. In Proceedings of the International Conference on Artificial Intelligence for Communications and Networks, Harbin, China, 25–26 May 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 240–250.
22. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
23. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **2002**, *4*, 23–33. [CrossRef]
24. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
25. Thrun, S. Probabilistic robotics. *Commun. ACM* **2002**, *45*, 52–57. [CrossRef]

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