BİTLİS EREN ÜNİVERSİTESİ
FEN BİLİMLERİ DERGİSİ

ISSN: 2147-3129 / e-ISSN: 2147-3188

Article Type : Research Article
Received : October 11, 2024
Revised : March 4, 2025
Accepted : March 6, 2025
DOI : 10.17798/bitlisfen.1565609

Year : 2025
Volume : 14
Issue : 1
Pages : 179-197

OBJECT TRACKING USING LIDAR DATA FILTERED BY MINIMIZED KALMAN FILTER ON TURTLEBOT3 MOBILE ROBOT

Kotiba ALDIBS[1], **Oğuz MISIR** [1*]

[1] *Bursa Technical University, Department of Mechatronics Engineering, Bursa, Türkiye*

**** Corresponding Author:*** *oguz.misir@btu.edu.tr*

ABSTRACT

The development of autonomous vehicles requires high accuracy and precision in sensor data for effective interaction with the environment and execution of functions. Processing this data with efficient algorithms positively influences vehicle decision-making. In this study, the TurtleBot3 platform, an ideal simulation model for autonomous vehicles, is used to detect and track nearby objects in the sub-system Robotic Operating System (ROS) Noetic environment. The lidar sensor data from this platform is refined using interpolation and a minimized Kalman filter to remove noise and irregularities. This approach provides clearer and more reliable measurement data, resulting in more stable and fine-tuned responses in the vehicle's motion planning. Compared to the general Kalman filter theory, this method offers faster implementation without relying on the exact error tolerance of the sensor to provide acceptable results.

Keywords: Mobile Robot, Autonomous, Kalman filter, Object tracking.

1 INTRODUCTION

As technology advances, research into mobile robots and autonomous vehicles is becoming increasingly important. Systems capable of performing various functions without human intervention, sensing environmental factors, and processing this information to make decisions have become quite common [1]. Among the systems developed are mobile robots used in logistics, service robotics, cleaning, industrial applications, and healthcare. These robots, which vary according to their purpose, are commonly known as Automated Guided

Vehicles (AGVs) or Autonomous Mobile Robots (AMRs) [2], [3]. While AGVs perform strip tracking by interacting with a magnetic field, AMRs perceive their environment through sensors and apply mapping and navigation processes through algorithms and methods [4]. The basic task for both AMRs and AGVs is to plan a route to a desired destination while avoiding collisions with obstacles in their environment [5]. AMRs typically use lidar sensors for mapping and obstacle detection. While the lidar sensors used are typically 2D, 3D lidar sensors are also used to obtain more geometric information about the environment [6]. In mobile robot applications, lidar is often preferred for simultaneous localization and mapping (SLAM) applications [7]. Researchers have been developing navigation and path planning methods using SLAM for a long time [8]. Lidar is also used in mobile robot applications to detect both dynamic and static obstacles [9], [10]. In environments where mobile robots operate, the objects encountered can be either stationary or moving.

These objects can serve as obstacles or be designated as targets to be followed. Object detection and position estimation are critical requirements for safe path and motion planning in mobile robot navigation [11]. In indoor environments, SLAM applications based on lidar data have developed methods that produce successful results in navigation and path planning. However, the inability to fully detect dynamic changes in the environment and its surroundings can limit the navigation of mobile robots [12], [13]. Lidar data can not only map the environment but also collect data to detect dynamic objects, allowing mobile robots to better understand their environment [14]. The development of these methods provides an opportunity to avoid additional costly requirements, but it requires processing through algorithms tailored to the application using lidar data. In addition, lidar data must be filtered prior to processing due to potential measurement errors inherent in electronic devices [15].

In this study, a 2D lidar-based dynamic object detection and tracking method has been developed that is capable of identifying objects of any class and geometry in a model-independent manner. Information obtained from lidar sensors in real-world conditions can lead to erroneous measurements due to reflections and noise, so the proposed method employs an interpolation-based process on the distance data from the lidar sensor to minimize these erroneous measurements, introducing a new approach that minimizes the noise of lidar data by reducing the reliance on Kalman filters. To implement this method, the Turtlebot3 Waffle pi version development platform was used, which is ideal for such problems and research. This platform also supports Robotic Operating Systems (ROS), which are used for environmental data processing for mobile robots, autonomous navigation, and mapping [16]. We made it

compatible with a developed method using ROS, enabling its application on both the real TurtleBot3 platform and the Gazebo robot simulator. Our method aims to minimize the Kalman filter, thereby reducing the noise in the real data obtained from the TurtleBot3's lidar sensor. We have also developed an approach to address this issue in the Gazebo simulator environment. We took measurements at different distances and conditions to enable the mobile robot to detect and track objects regardless of their geometry. We processed these measurements using the defined method to ensure appropriate filtering. The results showed successful results both in real conditions and in simulation environments.

2 RELATED WORKS

Researchers address tasks such as object recognition and tracking in mobile robot applications and develop different solution algorithms. In the object recognition method developed in [17], a trained CNN structure was used and implemented in the Gazebo environment. This study provided the robot with the ability to interpret the objects it encounters; however, the mode of operation was limited by the camera angle and the classes within the training set. Similarly, [18] used cameras, but at a more advanced level. Instead of mere object recognition, the study focused on real-time object detection by slicing the video data from the camera into frames and evaluating them. Algorithms capable of discriminating objects were used to feed a Kalman filter model with changes in the object's position across frames. The result was a system capable of predicting the next object position. In another study, we observe a low-cost model for human tracking. The measurement and detection device used in this model is a low-cost laser scanner [19].

This study focused on the identification and tracking of human footprints using an algorithm based on footprint detection in lidar data. According to this algorithm, the midpoint between two different footprints was designated as the target point, and the model showed good success results. An important feature of the model is its ability to track a single person without confusion, even when multiple people are present in the tracking area. In addition, the footprints of different people vary on the lidar data, and the distance between footsteps changes when walking. Special adjustments can be made for such cases. In [20], a Kalman filter was used to deal with irregularities in the readings of electronic measurement devices. Specifically, a fixed Kalman gain was used to analyze differences between the signal curves at the input and output of the filter, effectively mitigating irregularities in the sensor data. The study reported promising results in correcting measurement irregularities using this method.

3 APPROACH SYSTEM OVERVIEW

The scheme of the developed method for the detection of dynamic objects using 2D Lidar is shown in Figure 1. In the method, the distance information perceived by the lidar on the mobile robot is sometimes inaccurately or incompletely assigned due to noise and reflections on the real environment platform; therefore, the sensor data read from the real TurtleBot3 and the simulator environment may differ. In order to deal with these differences, two approaches have been taken in this study due to problems arising from different sensor structures between the physical system and the simulation environment, which involve the integration of different algorithms. In this study, the TurtleBot3 Waffle Pi platform manufactured by Robotis uses a 360-degree lidar (LDS-2) to measure distances to physical obstacles, with software components based on ROS (Robot Operating System) in Linux Ubuntu 20.04.

We chose to use ROS1 Noetic for this platform, which is compatible with both ROS1 and ROS2 versions. Besides the physical environment, the Gazebo software will be used for the virtual environment. As mentioned before, the readings of virtual and physical sensors may differ, and the sensor resolution may change too. In the physical platform, the linear velocity of the robot ranges from 0 to 0.26 m/s and the angular velocity from 0 to 1.82 rad/s, and the lidar sensor resolution is defined as 1 degree [16]. For the virtual platform, the properties are the same with a lower tolerance and higher confidence about the resolution of the lidar data. Robotis provides a ready-made system for TurtleBot3, and this software is primarily designed for the basic functions of the robot. Therefore, any additional applications require a ROS package and control from an external master.

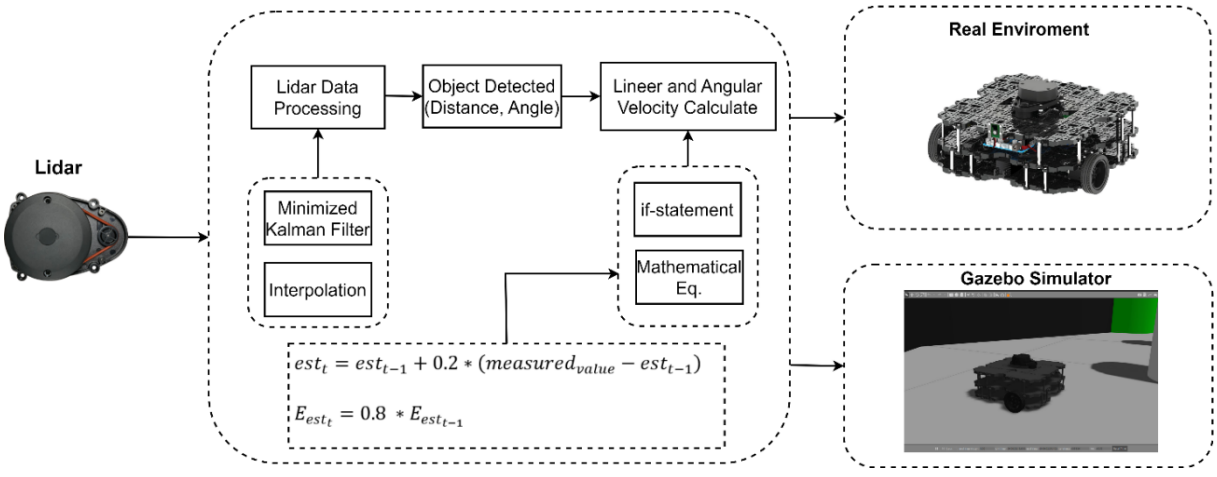


Figure 1. Method developed to detect dynamic objects using 2D Lidar.

In object tracking, the accuracy of the angle and distance information obtained from the lidar sensor is critical. During our research on the TurtleBot3 physical platform using the LDS-2 lidar sensor, we identified instances where distance measurements resulted in erroneous "zero" values due to reflections or incorrect readings. To avoid this problem, we compared the index numbers of erroneous distance measurements from the lidar sensor with neighboring index values and implemented an interpolation to mitigate these erroneous measurements. For the proposed method in the simulation environment, unlike the physical system, the distance information from the lidar sensor did not have zero values due to reflections, and the angles of the detected distances were sequentially synchronized. Therefore, there was no need for solutions to eliminate zeros in the simulation environment, allowing direct focus on the object detection method using the lidar.

3.1 Minimized Kalman Filter

Since Rudolf Kalman introduced the Kalman filter theory, it has been at the forefront of advanced research. Designed to overcome uncertainties in predicting the final state of a variable during its monitoring and measurement, this model fundamentally involves predicting a new value in place of the measured value, using the equations of the Kalman model [21], [22]. There exist improved and generalized versions with common equation variables. Equation (1) shows the Kalman Gain (KG), calculated based on parameters of the prediction error and the measurement error. KG is used in predicting the current value. Equation (2) expresses the current predicted value (est_t), which is obtained by adding the product of the KG coefficient and the difference between the measured value and the previous predicted value to the previous predicted value. Equation (3) calculates the prediction error (E_{est_t}) [20], [21], [23], [24], [25].

$$KG = \frac{E_{est}}{E_{est} + E_{measurement}} \tag{1}$$

$$est_t = est_{t-1} + KG * (measured - est_{t-1}) \tag{2}$$

$$E_{est_t} = (1 - KG) * E_{est_{t-1}} \tag{3}$$

So basically, the traditional Kalman filter could be defined with these three equations. The term est_{t-1} denotes the previous predicted value, while est_t denotes the current predicted value. E_{est} denotes the current prediction error, while $E_{measurement}$ denotes the measurement error. In the Kalman model, the relationship (measurement - old estimate) continuously facilitates new value predictions. Subsequently, the Kalman Gain (KG) coefficient is updated

and tends toward zero, allowing the predicted value to continuously converge to the actual value [20]. Figure 2 shows an example that graphically illustrates the prediction of the Kalman filter. Assuming a measurement of a value at 95 degrees with measurement errors in the range [-6, +6], an initial estimation error of -5 and an initial estimated value of 90 are chosen for consideration. This data is used to demonstrate how convergence to the true value occurs within 10 iterations.

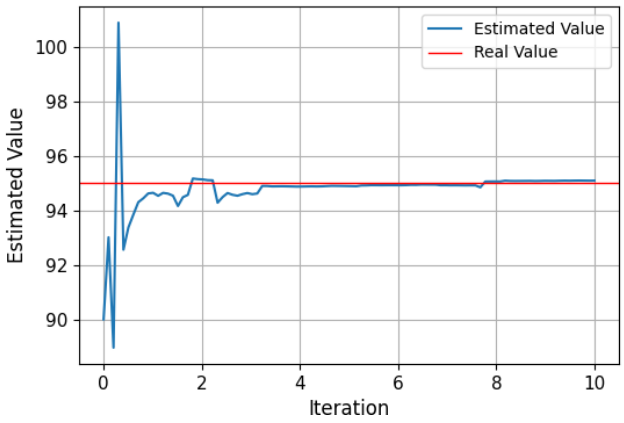
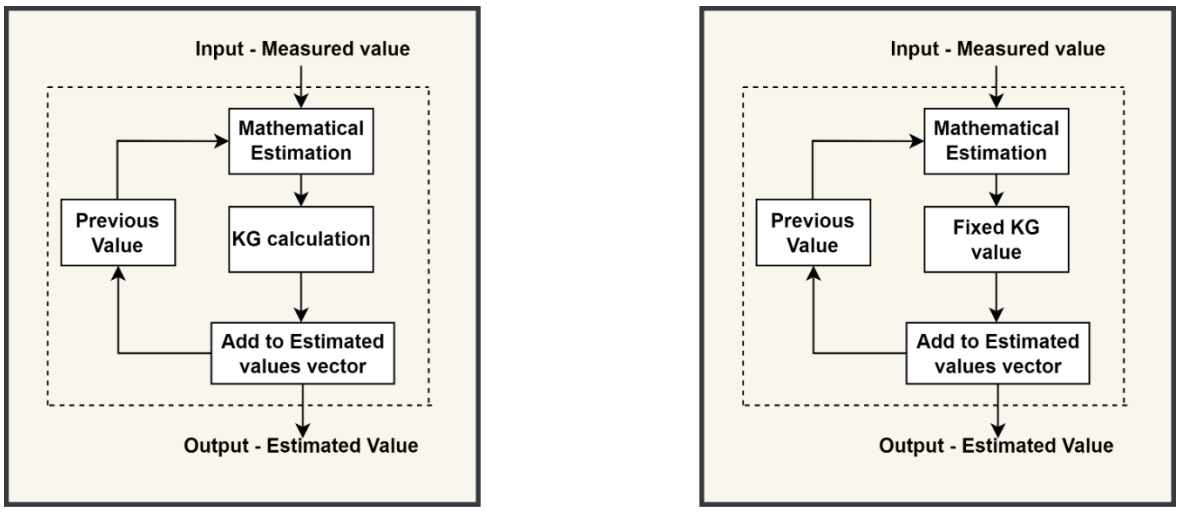


Figure 2. Comparison of actual and Kalman filter prediction graphs.

Considering the physical environment and the accuracy level of the LDS-2 lidar, the goal is to smooth the robot's responses. The response of the robot is highly dependent on the distance to the nearest object, so this parameter is prioritized. The measured distance from the lidar data is sometimes set to zero and at other times varies due to measurement tolerances. In such cases, instead of relying on the measured minimum, estimation is used to reduce the sensitivity to fluctuations. To produce an estimate at any time t, the previous predicted value is required; if this value is not available, it is initialized to the current measured value to start the algorithm. Figure 3 shows both the normal Kalman filter and the improved minimized Kalman filter in comparison.

The smoothing aims to achieve independence from the values of the estimation and measurement errors, as indicated by the Kalman Gain (KG) coefficient in equation (1), which depends on the ratio of the prediction and measurement errors. To achieve this, the change in estimated values is kept constant relative to the difference between the current measured state and the previous state values, thus ensuring continuous convergence of the estimated value to the measured value. Therefore, as expressed in equation (4), the KG coefficient is kept constant. Through experimentation, this value was determined to be 0.2 to suit the robot model. Equation (5) expresses the difference between the current estimated value and the previous value. If

incorrect measurement errors occur, they are absorbed by the improved Kalman filter without causing abrupt responses.



(a) Kalman Filter (b) Minimized Kalman Filter

Figure 3. Kalman Filter and Minimized Kalman Filter Diagram.

$$KG = \frac{\Delta est}{measured\ value\ - est_{t-1}} \tag{4}$$

$$\Delta est = est_t - est_{t-1} \tag{5}$$

When the KG value is fixed, the relationship between the measurement error and the estimation error is also fixed to a constant value. This means that each position update of the object depends on the influence of the previous estimated state and 20% of the change in the measured value, thus achieving smoother responses, which lead our modifications on equations (2) and (3) to produce the following (6) and (7) equations.

$$est_t = est_{t-1} + 0.2 * (measured - est_{t-1}) \tag{6}$$

$$E_{est_t} = 0.8\ * E_{est_{t-1}} \tag{7}$$

During the movement of the mobile robot, there are sometimes inappropriate responses such as sudden changes in speed and direction. The causes of this problem include abrupt transitions caused by simple if-else statements, sudden changes in distance from the lidar sensors, and measurement errors. To solve these problems, we propose variable velocity and angular velocity coefficients that can absorb sudden changes in linear and angular velocity. We can achieve changes in linear velocity through a mathematical function where the variable is the position angle of the object. Since the angle values range from 0 to 180 degrees, we can

map them to the range [0, 1], which is called the mapped_theta variable, so that the output of this function is in the range [0, 1]. In other words, using this coefficient, we can scale the speed value as a percentage of the maximum speed value. So, our linear velocity changing rules can be listed as follows: (1) maximum when the angle of the object is less than 30 degrees; (2) decreases exponentially after 30 degrees; and (3) tapers to zero near 90 degrees. The appropriate mathematical function that combines these three points is expressed in equation (8), which uses the mapped_theta as its independent variable. Here, $kv(\theta)$ refers to the linear velocity controller, and θ represents the mapped_theta. In addition, the graph of this function is shown in Figure 4.

$$kv(\theta) = \frac{1}{1 + 0.2 * e^{(20*(\theta - 0.167))}} \tag{8}$$

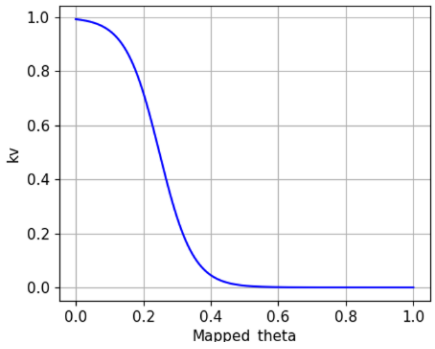


Figure 4. kv function.

According to this graph, the coefficient kv begins a rapid descent when the mapped_theta variable reaches 0.167 (corresponding to a 30-degree angle). At a value of 0.34, corresponding to a 60-degree angle of the object, the coefficient becomes 0.15 when multiplied by the maximum speed, giving a value of 15%. As mapped_theta approaches 0.5, or as the object angle approaches 90 degrees, the output speed converges to zero. This satisfies these criteria and provides a nonlinear change in linear velocity. Using this method, we can control the angular velocity. However, as the object angle decreases, we need a function where the angular velocity value also decreases-a reverse version of the linear velocity criteria. The function used for the change in angular velocity is expressed in equation (9). Here, $kw(\theta)$ corresponds to the angular velocity coefficient change.

$$kw(\theta) = \frac{1}{1 + e^{(-15*(mapped_{theta} - 0.334))}} \tag{9}$$

Figure 5 displays the graph of the function that handles the change in angular velocity. As the object's angle approaches zero, it ensures that the angular velocity converges to zero. In addition, a function was obtained that non-linearly increases the angular velocity as the object angle exceeds 30 degrees.

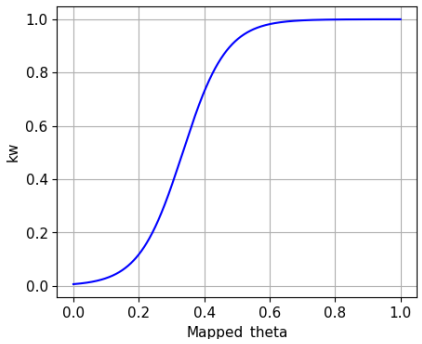


Figure 5. kw function.

3.2 Proposed Approach for The Virtual (Gazebo) Environment

Lidar data was used to implement object tracking using the Gazebo simulator as the simulation environment. Gazebo includes the Turtlebot3 robot model, which is equipped with sensors and actuators similar to the physical hardware. To run this method in the Gazebo simulator environment, a turtle_world environment was used. To obtain lidar data, a data subscription was performed on the "/scan" topic within a ROS package node, allowing access to the transmitted data. The content of the data was examined using the "/rostopic info" command to identify and utilize parameters critical to the mobile robot. Figure 6 illustrates the subscribed topics within the Gazebo environment.

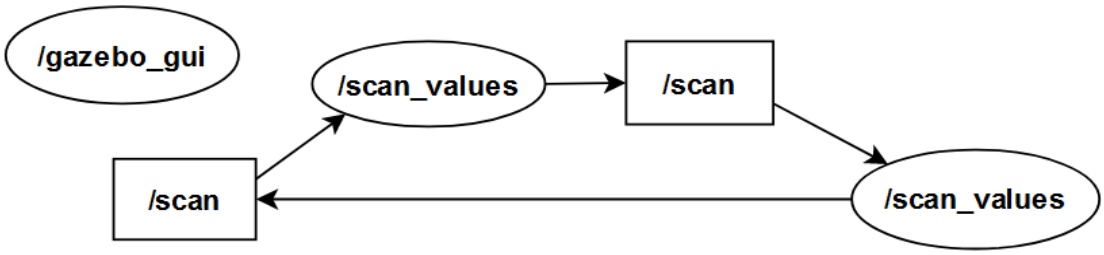


Figure 6. Data flow in the Gazebo environment.

Figure 7 shows the object detection method diagram for the Gazebo environment. It shows the process where the nearest detected lidar distance data and its index are identified from the "/scan" scan data obtained with the help of ROS. To guide or move the mobile robot to the nearest object, this process is facilitated by themes defined by ROS. The control of linear

and angular velocities is transferred to the right and left motors of the mobile robot using the Twist function. We ensured successful navigation of the mobile robot to the detected object based on specified angle and distance tolerances.

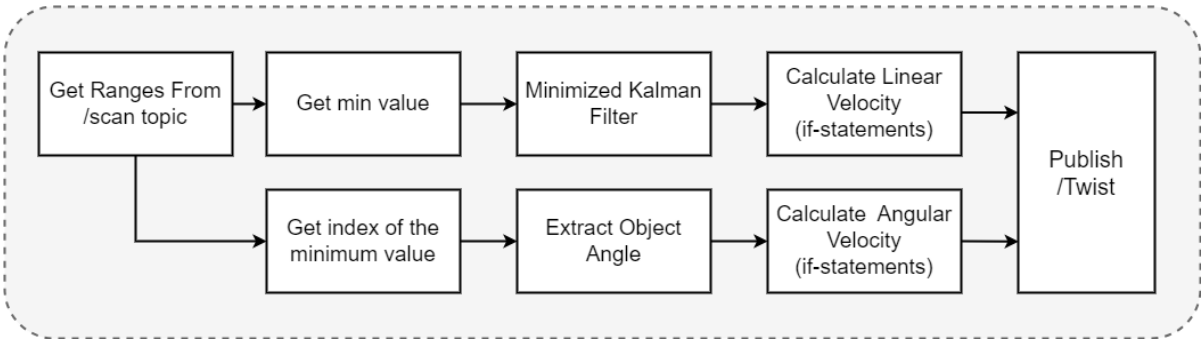


Figure 7. Developed object detection algorithm for virtual environment.

The pseudo code expressed in virtual_env1 for calculating linear and angular velocities can be explained in terms of multiple steps. According to this code, the closest vector distance data is determined from the lidar sensor. Then the index number of the closest distance is extracted, which allows the relative angle of the closest distance with respect to the robot to be calculated. To orient the robot to the object based on the calculated angle and distance, the angular difference between the robot's heading angle and the angle of the detected object is calculated. If this angular difference is within +/- 8 degrees, the angular velocity is set to 0 to minimize oscillations during orientation. If the angular difference is -90 or 90 degrees, indicating that the object is directly to the left or right, the angular velocity is set to maximum in the opposite direction of the object. At the same time, if the distance to the object is less than 0.35 units, the linear velocity is set to 0, causing the robot to stop. Conversely, if the distance is greater than 0.35 units, the linear velocity is maximized, allowing the robot to search for or track the object. A notable drawback of this method is the use of numerous if-else structures. In a simplified way, in the following Algorithm 1. table a summary of the operations in the form of codes.

Algorithm 1. Coded representation of the algorithm's flow.

Virtual-Env-1

Subscribe /scan
Subscribe /Twist
Get /scan
$lidar\_min\_range \leftarrow \min(laserData.ranges)$
$lidar\_min\_ranges\_angle \leftarrow detect\_angle(lidar\_min\_range)$
$angle\_diff \leftarrow heading\_angle - lidar\_min\_ranges\_angle$
if $-8 \leq angle\_diff \leq 8$ **then**
    $angular\_vel \leftarrow 0$
else if $angle\_diff < -90$ **then**
    $angular\_vel \leftarrow max$
else if $angle\_diff > 90$ **then**
    $angular\_vel \leftarrow -max$
end if
if $lidar\_min\_range < 0.35$ **then**
    $linear\_vel \leftarrow 0$
else
    $linear\_vel \leftarrow max$
end if

3.3 Proposed Approach for The Physical Environment

Unlike the virtual environment, the mobile robot in the physical environment has a different number of topics, as shown in Figure 8. These topics are used to share sensor information and monitor changes in the hardware. The noise generated by the Lidar sensor has resulted in some distances being assigned a value of "0" and in the detection of sudden noisy distance values during measurements.

Figure 9 outlines the solutions to these problems, with the steps summarized as follows: first, the indices of the zeros were identified. Since these are not fixed indices, new values were assigned by interpolating between the real values before and after the zeros. To detect the nearest object, the nearest distance from the noise-filtered data is used as a reference. The index of this nearest distance is also used to calculate the angle of the object. To avoid errors in the measured minimum values, an estimation is made using the proposed minimized Kalman filter. This method also helps to avoid sudden speed changes and wrong turns. To navigate to the object at the determined angle and distance, the linear and angular velocities are calculated. These velocities are then sent to the motors using the '/Twist' publisher.

Figure 10 shows the comparison between the raw lidar data containing zeros and erroneous measurements and the data after the interpolation process. It is evident that the values initially assigned as zero ("0") have been absorbed after interpolation.

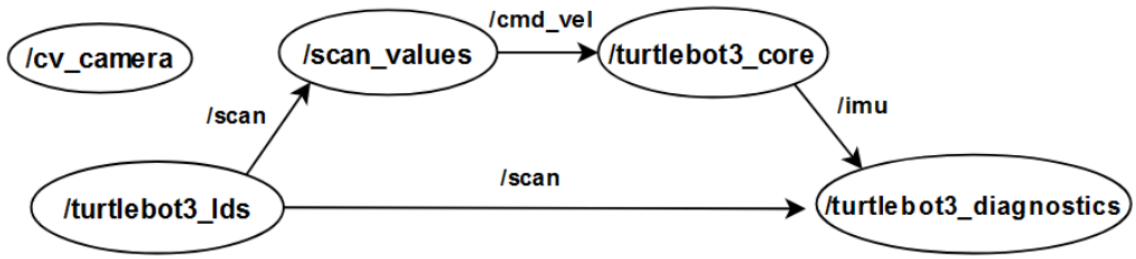


Figure 8. Data flow in Turtlebot3 Waffle Pi.

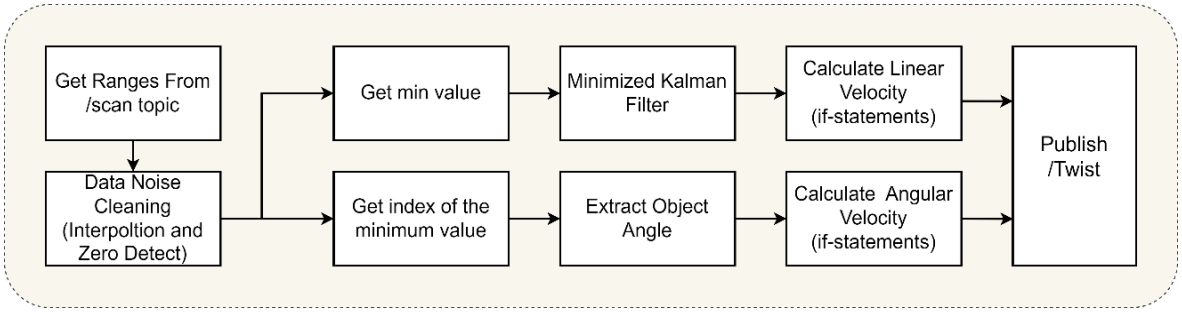


Figure 9. Developed object detection algorithm for physical environment.

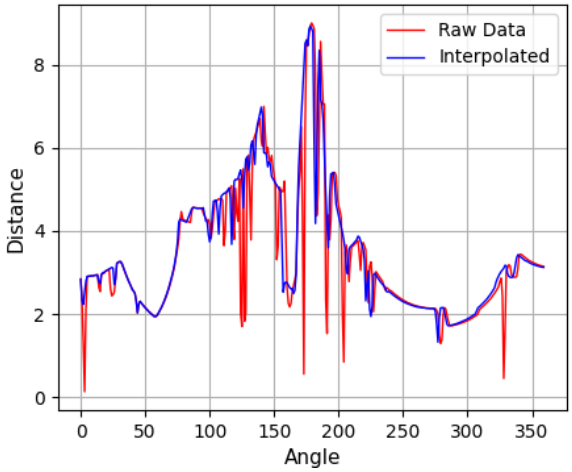


Figure 10. The impact of interpolation on raw lidar data.

4 EXPERIMENTAL RESULTS

In this study, an improved Kalman filter-based method for the detection and tracking of dynamic objects using 2D Lidar is proposed. The proposed method provides different solutions for simulated and physical Turtlebot3 environments. Applications have been developed for detecting objects at different distances and angles both in the physical environment and on the

Turtlebot3 platform. Furthermore, the developed minimized Kalman filter method addresses how improvements were made to noisy raw lidar data.

4.1 Results for Physical Turtlebot3 Environment

The physical environment experiments used the Waffle Pi version of the Turtlebot3. In the first experiment, objects placed at angles of 30, 60, and 90 degrees were detected, and their angles relative to the Turtlebot3 were determined. Figure 11 shows objects placed at angles of 30, 60, and 90 degrees relative to the Turtlebot3.

In the experiments conducted, objects placed at a distance of less than 1 meter were analyzed for velocity variations using two different methods. Figure 12 (a) shows the velocity variations based on different angles using a simple if-else structure. Similarly, Figure 12 (b) shows the velocity variations obtained using the proposed method aimed at smoothing changes based on different angle values.

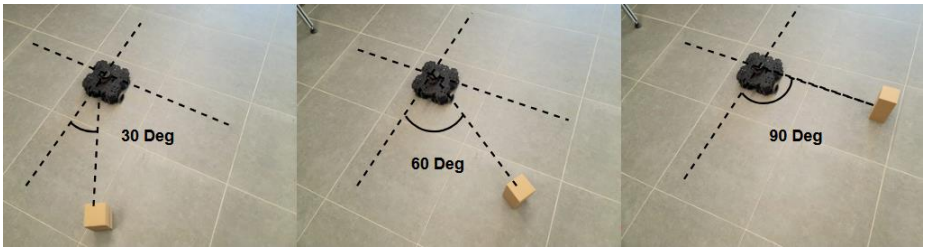


Figure 11. Detection of objects placed at different angle positions.

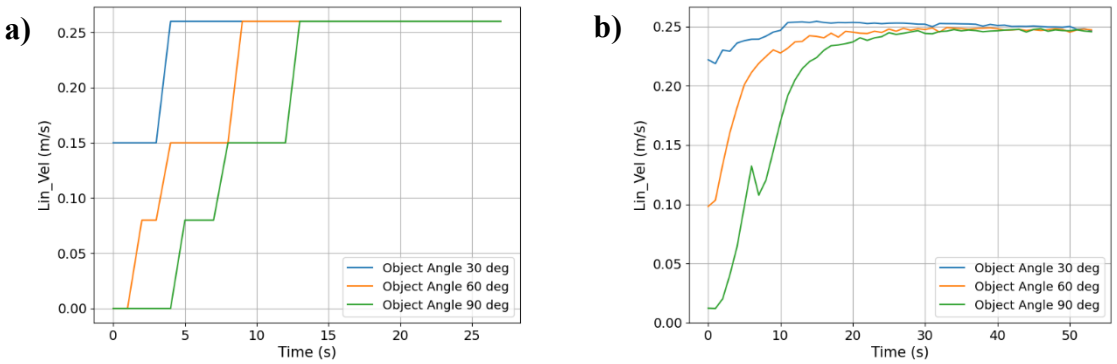


Figure 12. Velocity responses of Robot speed relative to object placed at different angles for physical environment. a) Linear velocity changes using if-else conditions, b) Linear Velocity changes using math equations.

The velocity responses to objects placed at different angles using a simple if-else structure show abrupt changes, resulting in the robot's jerky movements. As shown in Figure 12(a), the velocity changes are characterized by high accelerations. To reduce these high

accelerations, the proposed method uses exponential equations depending on the acceleration coefficients, as shown in Figure 12(b). Acceleration rates are reduced in steps of 30 degrees, resulting in smoother velocity changes. To verify the object detection based on the minimum detected distance, measurements were taken for three different distances in front of the robot as shown in Figure 13, by defining three distances of 35 cm, 65 cm, and 95 cm, in this test we used a static and fixed object in order to observe the efficiency of the designed filter.

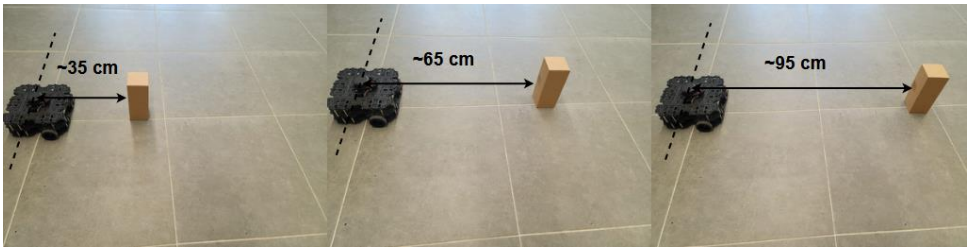


Figure 13. Distance measurement application of object at different distances.

As a result, the effects of changes in the minimum distance are shown in Figures 14 (a) and (b), comparing the outputs of noisy data and the proposed minimized Kalman filter. According to the comparative results obtained based on variations in different distances, it can be seen that the object distances obtained with the minimized Kalman filter are denoised compared to the raw lidar data. It is observed that the use of the minimized Kalman filter reduces sudden accelerations and abrupt oscillations caused by noisy data during object tracking by the mobile robot

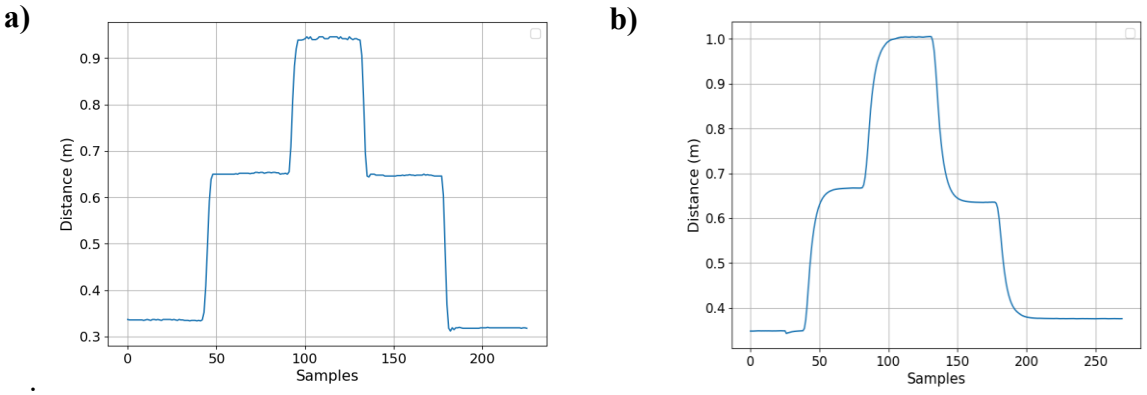


Figure 14. Object detection distances based on raw and minimized Kalman filtered data for the physical environment. a) Object distance detection based on raw lidar data b) Object detection distance obtained by using minimized Kalman filter on lidar data.

In addition, the distance of a stationary object was monitored over a period of time, and comparative data including raw lidar data, data processed through the Kalman filter, and data processed through the enhanced minimized filter are shown in Figure 15. This collection includes lidar data obtained from a stationary object at a given distance. According to the results obtained, the data processed through the developed minimized Kalman filter shows a better absorption of fluctuations compared to both the noisy raw data and the general Kalman filter.

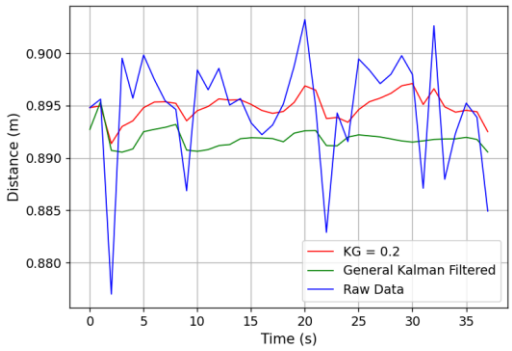


Figure 15. Comparison of raw lidar data, data passed through the general Kalman filter, and data passed through the minimized Kalman filter.

4.2 Results for Gazebo Environment

Experiments conducted in the Gazebo environment involve examining changes in velocity by placing different objects at different angles and fixed distances, similar to those in the physical environment. Unlike the real environment, the virtual environment is free of zeros. Therefore, we wanted to observe the accuracy of our method in both virtual and physical environments. The experiments conducted in the Gazebo environment are shown in Figure 16.

The speed variation of the object placed at 30, 60, and 90 degrees in the virtual environment is shown in Figure 17, as specified in the method, showing both the speed variations based on the simple if-else structure and those adjusted using exponential speed coefficients. According to the results obtained, smoother speed transitions were observed in the virtual environment due to the absence of physical environment noise.

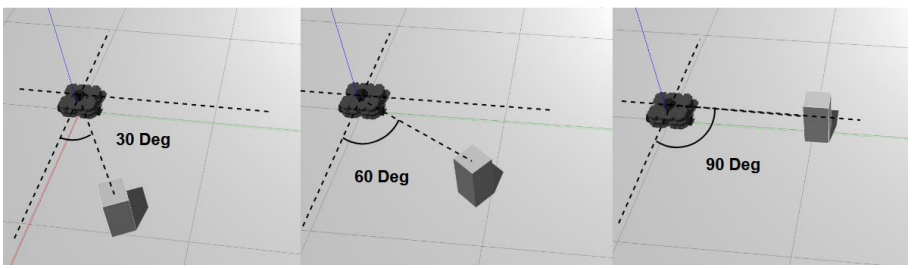


Figure 16. Experiment on object distance with angles of 30, 60, and 90 degrees in the Gazebo environment.

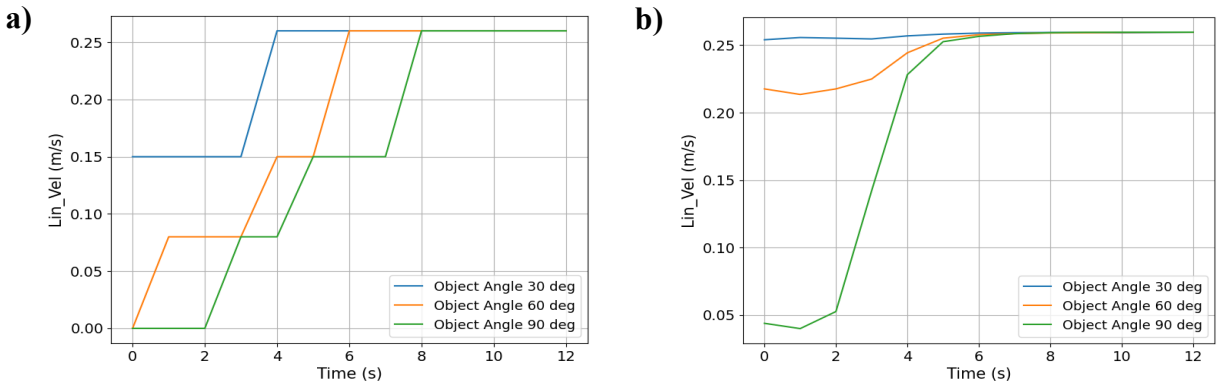


Figure 17. Robot speed responses according to object placed at different angles for virtual environment. a) linear speed changing using if-else conditions for virtual environment b) linear velocity changing using mathematical equations for virtual environment.

Figure 18 illustrates experiments conducted in the virtual gazebo environment with objects placed at different fixed distances. This experiment focuses on the distances perceived by the lidar for objects placed at 3 different distances.

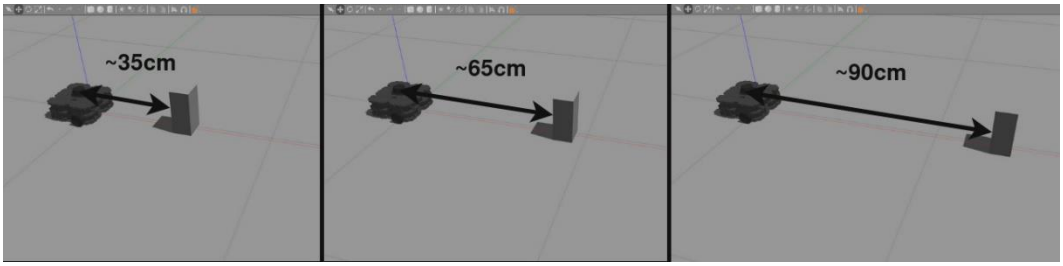


Figure 18. Experiment applied for different fixed distances for the virtual environment.

Figure 19 shows the closest object distances obtained from lidar data with and without application of the minimized Kalman filter in the virtual environment at 3 different distances. In the results with the minimized Kalman filter applied, the distance measurement transitions are smoother compared to the unfiltered lidar data in the virtual environment, where the lidar data is free from noise-induced sudden distance transitions.

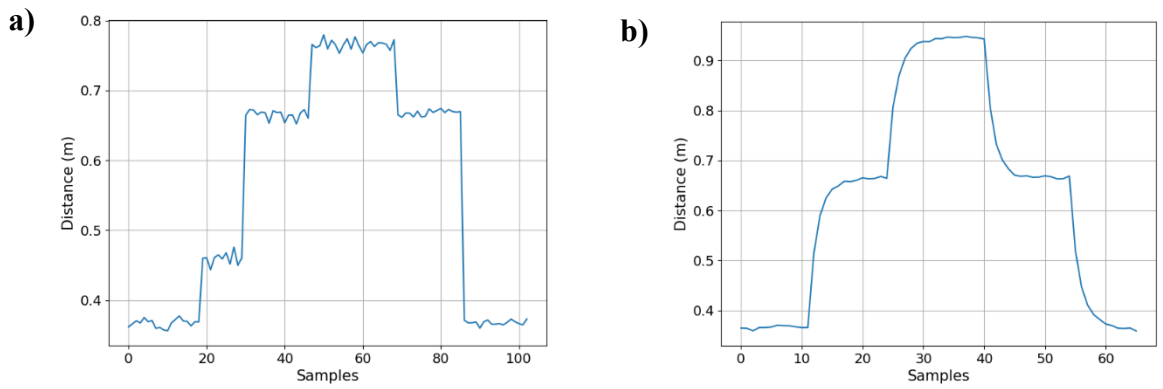


Figure 19. Object detection distances based on raw and minimized filtered data in the gazebo environment. a) Object distance detection based on raw lidar data b) Object detection distance obtained by using minimized Kalman filter on lidar data.

5 CONCLUSION

This paper proposes a method based on an extended Kalman filter for tracking objects without specifying geometries and shapes, so it is focused on enhancing the quality of lidar data intended for use in autonomous mobile robots; however, as an examples, a 35x20x20 cm box and observer legs have been used in different tests. Real-world lidar data often produce erroneous measurements due to reflections and noise. To mitigate these inaccuracies, a minimized Kalman filter methodology incorporating interpolation and Kalman filtering techniques has been developed. The proposed method is adaptable to both the Turtlebot3 physical robot and the Gazebo simulation environment, presenting two different approaches. Experimental trials for object tracking have been conducted using both the physical Turtlebot3 and the Gazebo environment. Instead of a simple control mechanism where linear and angular velocities change based on the perceived angle and distance of the object, the study uses well-tuned mathematical functions with intervals and breakpoints to achieve smoother velocity transitions. In addition, the minimized Kalman filter was observed to protect against sudden jumps in measurements and to mitigate state noise during discrete-time measurements. Object detection and tracking are performed using lidar information, and future work will focus on improving object classification using new lidar data, considering both computational performance and object tracking accuracy.

Acknowledgment

The authors would like to acknowledge Bursa Technical University, specializing in Robotics and Intelligent Systems, for providing research opportunities and support.

Conflict of Interest Statement

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The study is complied with research and publication ethics.

Artificial Intelligence (AI) Contribution Statement

This manuscript was entirely written, edited, analyzed, and prepared without the assistance of any artificial intelligence (AI) tools. All content, including text, data analysis, and figures, was solely generated by the authors.

Contributions of the Authors

Conceptualization, K.A. and O.M.; methodology, O.M. and K:A.; software, K.A.; validation, O.M. and K.A.; supervision, O.M.

REFERENCES

[1] M. A. K. Niloy *et al.*, "Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review," *IEEE Access*, vol. 9, pp. 35338–35370, 2021, doi: 10.1109/ACCESS.2021.3062557.

[2] Z. Bai, H. Pang, Z. He, B. Zhao, and T. Wang, "Path Planning of Autonomous Mobile Robot in Comprehensive Unknown Environment Using Deep Reinforcement Learning," *IEEE Internet Things J*, Jun. 2024, doi: 10.1109/JIOT.2024.3379361.

[3] F. Li, J. Cheng, Z. Mao, Y. Wang, and P. Feng, "Enhancing Safety and Efficiency in Automated Container Terminals: Route Planning for Hazardous Material AGV Using LSTM Neural Network and Deep Q-Network," *Journal of Intelligent and Connected Vehicles*, vol. 7, no. 1, pp. 64–77, Mar. 2024, doi: 10.26599/JICV.2023.9210041.

[4] A. Bhargava, M. Suhaib, and A. S. Singholi, "A review of recent advances, techniques, and control algorithms for automated guided vehicle systems," Jul. 01, 2024, *Springer Science and Business Media Deutschland GmbH*. doi: 10.1007/s40430-024-04896-w.

[5] E. A. Oyekanlu *et al.*, "A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications," 2020, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2020.3035729.

[6] H. Kim, H. Kim, S. Lee, and H. Lee, "Autonomous Exploration in a Cluttered Environment for a Mobile Robot With 2D-Map Segmentation and Object Detection," *IEEE Robot Autom Lett*, vol. 7, no. 3, pp. 6343–6350, Jul. 2022, doi: 10.1109/LRA.2022.3171069.

[7] H. Dong, C. Y. Weng, C. Guo, H. Yu, and I. M. Chen, "Real-Time Avoidance Strategy of Dynamic Obstacles via Half Model-Free Detection and Tracking with 2D Lidar for Mobile Robots," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 2215–2225, Aug. 2021, doi: 10.1109/TMECH.2020.3034982.

[8] Z. Xu, J. Y. Wu, and Q. L. Liu, "Research on mobile robot indoor positioning mapping based on front-end and back-end optimization," *Journal of Mechanical Science and Technology*, vol. 38, no. 5, pp. 2555–2561, May 2024, doi: 10.1007/s12206-024-0434-0.

[9] Y. Chen, B. Pervan, and M. Spenko, "Quantifying the Risk of Unmapped Associations for Mobile Robot Localization Safety," *IEEE Transactions on Robotics*, 2024, doi: 10.1109/TRO.2024.3401093.

[10] O. Misir and L. Gökrem, "Dynamic interactive self organizing aggregation method in swarm robots," *BioSystems*, vol. 207, Sep. 2021, doi: 10.1016/j.biosystems.2021.104451.

[11] X. Li, L. Wang, Y. An, Q. L. Huang, Y. H. Cui, and H. S. Hu, "Dynamic path planning of mobile robots using adaptive dynamic programming," *Expert Syst Appl*, vol. 235, Jan. 2024, doi: 10.1016/j.eswa.2023.121112.

[12] F. Li, C. Fu, D. Sun, J. Li, and J. Wang, "SD-SLAM: A Semantic SLAM Approach for Dynamic Scenes Based on LiDAR Point Clouds," *Big Data Research*, p. 100463, May 2024, doi: 10.1016/j.bdr.2024.100463.

[13] V. Vaquero, E. Repiso, and A. Sanfeliu, "Robust and real-time detection and tracking of moving objects with minimum 2d lidar information to advance autonomous cargo handling in ports," *Sensors (Switzerland)*, vol. 19, no. 1, 2019, doi: 10.3390/s19010107.

[14] Z. Liu *et al.*, "Positive and Negative Obstacles Detection Based on Dual-LiDAR in Field Environments," *IEEE Robot Autom Lett*, 2024, doi: 10.1109/LRA.2024.3414256.

[15] H. Zhang, R. Xiao, J. Li, C. Yan, and H. Tang, "A High-Precision LiDAR-Inertial Odometry via Invariant Extended Kalman Filtering and Efficient Surfel Mapping," *IEEE Trans Instrum Meas*, vol. 73, pp. 1–11, 2024, doi: 10.1109/TIM.2024.3382751.

[16] Robotis, "Turtlebot3," *https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/*.

[17] C. Nandkumar, P. Shukla, and V. Varma, "Simulation of Indoor Localization and Navigation of Turtlebot 3 using Real Time Object Detection," in *Proceedings of IEEE International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications, CENTCON 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 222–227. doi: 10.1109/CENTCON52345.2021.9687937.

[18] P. R. Gunjal, B. R. Gunjal, H. A. Shinde, S. M. Vanam, and S. S. Aher, "Moving object tracking using kalman filter," in *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, IEEE, 2018, pp. 544–547.

[19] Y. Ge and W. Li, "Human following of mobile robot with a low-cost laser scanner," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, IEEE, 2019, pp. 3987–3992.

[20] Z. Khan, H. Bugti, and A. S. Bugti, "Single dimensional generalized kalman filter," in *2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, IEEE, 2018, pp. 1–5.

[21] D. Simon, "Kalman filtering," *Embedded systems programming*, vol. 14, no. 6, pp. 72–79, 2001.

[22] H. Liu *et al.*, "Uncertainty-Aware UWB/LiDAR/INS Tightly Coupled Fusion Pose Estimation via Filtering Approach," *IEEE Sens J*, vol. 24, no. 7, 2024, doi: 10.1109/JSEN.2024.3362741ˆx.

[23] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *Proceedings - 8th International Conference on Intelligent Networks and Intelligent Systems, ICINIS 2015*, Institute of Electrical and Electronics Engineers Inc., Aug. 2016, pp. 74–77. doi: 10.1109/ICINIS.2015.35.

[24] G. Welch and G. Bishop, "An introduction to the Kalman filter," 1995.

[25] H. T. Madan, V. Akram, S. Reddy, K. N. Mohan Gowda, and L. D. Uday Kumar, "An Error-State Extended Kalman Filter Based State Estimation and Localization Algorithm for Autonomous Systems," in *International Conference on Smart Systems for Applications in Electrical Sciences, ICSSES 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICSSES58299.2023.10199535.