Environmental Localization and Detection Using 2D LIDAR on a Non-Holonomic Differential Mobile Robot

Muhammad Azriel Rizqifadiilah [1*], Trihastuti Agustinah [1], Achmad Jazidie [1]

[1] Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[*]azriel.imanto@gmail.com

Abstract: This research presents a novel approach for environmental localization and multi-object detection using a 2D LIDAR system integrated into a non-holonomic differential mobile robot. The proposed methodology combines SLAM real-time localization and Euclidean Clustering for detecting surrounding objects, enabling the robot to localize effectively within its working environment and identify multiple obstacles. By segmenting LIDAR data into discrete object clusters, the system can accurately determine the size and shape of detected obstacles, providing a detailed understanding of the environment. Simulation results demonstrate that the proposed approach effectively addresses the challenges of localization and multi-object detection. The Euclidean Clustering approach has shown to be a lot more effective. With a time performance of 443.1677 seconds, it finishes the mission. The Euclidean clustering is significantly faster than the K-Nearest Neighbors (K-NN) technique, which required 11256 seconds with a K value of 100 and 16542 seconds with a K value of 200. The performance of autonomous mobile robots is improved because Euclidean Clustering provides a better time performance in these specific scenarios.

1. Introduction

Environmental localization and detection are essential for mobile robots to navigate and perform tasks in complex and dynamic environments. Some dynamic environments include a wall, a static object, and a person or pedestrian. In the previous study [1] pedestrian pursuers were referred to as pedestrians with a kinetic movement. In recent years, the incorporation of two-dimensional LiDAR sensors in non-holonomic differential mobile robots has attracted considerable interest, as it can enhance the robots' sensing and positioning abilities [2].

Two-dimensional LiDAR sensors employ laser pulses to determine distances and create a 2D representation of the area, aiding the robot in understanding its environment and evading obstructions [3]. These sensors have been extensively employed in autonomous vehicles, drones, and other robotic applications because of their accuracy, real-time performance, and reliability under various environmental conditions [4].

Non-holonomic differential mobile robots are wheeled robots with motion constraints, rendering them incapable of moving directly and sideways [5]. These robots typically have two independently driven wheels, allowing them to change direction by adjusting each wheel's speed [6]. This design offers enhanced manoeuvrability in confined spaces and increased stability on uneven terrain.

Applications, including search and rescue, environmental monitoring, and warehouse automation, have all used the pairing of 2D LiDAR sensors and non-holonomic differential mobile robots [7]. Research in this area has concentrated on developing algorithms and techniques for efficient localization, mapping, and path planning [6]. Moreover, studies have explored methods for fusing LiDAR data with other sensors, like cameras and inertial measurement units (IMUs), to boost the system's overall performance and robustness.

Several algorithms have been developed for processing raw LiDAR data for effective localization and detection. Such algorithms can generally be categorized into two main types: feature-based and grid-based methods [8]. Feature-based methods involve extracting salient features from LiDAR data, such as corners, edges, and lines, and then matching these features to a known map or database [9]. Grid-based methods, in contrast, typically involve dividing the environment into a grid and updating the grid cells based on LiDAR measurements [10]. Previous research can be accomplished using occupancy grid mapping. Grid-based techniques model the world as a probabilistic grid, where each cell denotes the likelihood that an object will occupy it [11].

A key aspect of localization and detection is addressing the inherent uncertainty in the robot's motion and sensor measurements. Probabilistic techniques, such as particle filters and Kalman filters, enable the robot to maintain a probability distribution over its potential locations and update this distribution as new sensor measurements become available.

SLAM has emerged as a popular technique for localization and mapping in mobile robots, allowing them to navigate unknown environments. SLAM algorithms use sensor data to map the environment and estimate the robot's position within it at the same time [12]. Several SLAM variants, such as Extended Kalman Filter (EKF)-SLAM, Fast-SLAM, and Graph-SLAM, have been developed and applied to diverse robotic applications, including 2D LiDAR-based systems [13].

Many clustering algorithms exist, including Euclidean clustering, K-Nearest Neighbours (K-NN), Hierarchical Clustering, and many others. This research suggests Euclidean clustering, widely adopted for object detection due to its simplicity and effectiveness. This method segments point cloud data into clusters based on the Euclidean distance between points. The clusters represent individual objects in the environment, which can be further processed for recognition and tracking [14]. Object detection was discus in autonomous driving, warehouse robotics, and environmental monitoring [15].

Combining SLAM for localization and Euclidean clustering for detection has effectively addressed

environmental localization and detection problems in mobile robotics. For instance, [16] used SLAM for mapping and localization in an urban environment while employing Euclidean clustering to detect and track dynamic objects such as pedestrians and vehicles. Similarly, [17] presented an integrated system for an autonomous mobile robot that used Graph-SLAM for localization and Euclidean clustering for obstacle detection.

This integration has proven effective in various challenging scenarios, outperforming other methods. This research seeks to address dynamic and complex environments more effectively. The environment in this study is made as complex as feasible by incorporating static obstacles, such as walls and tables, and dynamic obstacles, such as pedestrians pursuer.

2. Material and Method

2.1. Non-Holonomic Mobile Robot and Obstacle Modelling

The robot used a two-wheeled differential drive to move around. The system enables the robot to travel in a straight path within the Cartesian plane, as depicted in Fig. 1. The robot's center of mass is situated in its middle, labelled as C, and its width measures $2L$. The wheel diameter is represented by $2R_a$, while the distance between the center of mass C and the wheel axis A has a value of α.

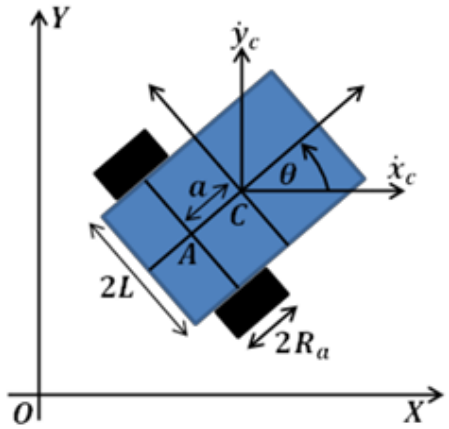

Fig. 1. *Non-holonomic Differential Mobile Robot* [18].

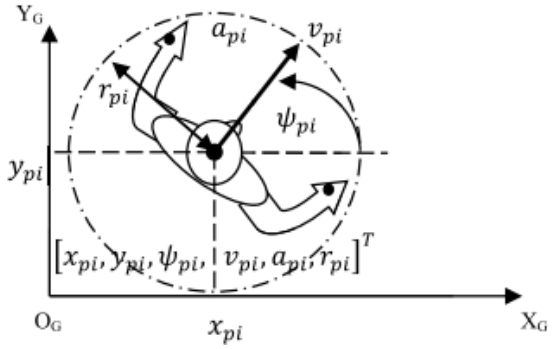

Fig. 2. *Pose of a single pedestrian in X-Y coordinates* [1]

Table 1. System Parameter

Parameter	Symbol	Value
Mass	m	4
Wheelbase to the robotic rotary shaft	a	0,25
Total inertia	I	2
Wheel radius	R_a	0.15
Half the width of the robot	L	2

The differential drive robot's dynamics and kinematics model are formulated with the following equation.

$$M(q)\ddot{q} + V(q,\dot{q}) = B(q)\tau - A^T(q)\lambda \tag{1}$$

$$\begin{bmatrix} m & 0 & ma\sin\theta \\ 0 & m & -ma\cos\theta \\ ma\sin\theta & -ma\cos\theta & Ic + 2ma^2 \end{bmatrix}\ddot{q} + \begin{bmatrix} ma\dot{\theta}^2 cos\theta \\ ma\dot{\theta}^2 sin\theta \\ 0 \end{bmatrix} = \frac{1}{R_a}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix}\begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} - \begin{bmatrix} -\sin\theta \\ \cos\theta \\ -a \end{bmatrix}\left[-m(\dot{x}_c\cos\theta + \dot{y}_c\sin\theta)\dot{\theta}\right] \tag{2}$$

Simplification of the dynamics and kinematics model using the transformation matrix s so that it becomes the following equation.

$$\dot{q} = Sv \tag{3}$$

$$S^T MS\dot{v} + S^T\left(M\dot{S} + V_m S\right)v + F + \tau_d = S^T B\tau \tag{4}$$

$$\bar{M}\dot{v} + \bar{V}_m v = \bar{B}\tau \tag{5}$$

$$S = \begin{bmatrix} \cos\theta & -a\sin\theta \\ \sin\theta & a\cos\theta \\ 0 & 1 \end{bmatrix} \quad q = \begin{bmatrix} x_c \\ y_c \\ \theta \end{bmatrix} \quad v = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{6}$$

So that the results of the simplification of the kinematics model are as follows

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -a\sin\theta \\ \sin\theta & a\cos\theta \\ 0 & 1 \end{bmatrix}\begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \end{bmatrix} \tag{7}$$

The parameters of the differential drive robot model used in this study have a value in Table 1.

The previous research in [1] concerned pedestrian pursuers pose. It moves with variable speeds and nonlinear trajectories, giving it the same characteristics as a single pedestrian. Fig. 2 shows a single property for pedestrians. The human individual can be expressed using pose information in coordinates $X - Y$ $[x_{hi},\ y_{hi},\ \psi_{hi}]^T$, speed v_{hi}, acceleration a_{hi} and radius of the body from the individual human to-i^{th} annotated with r_{hi} which can be affected by the reach of the hand or walker can be formulated as follows

$$\boldsymbol{S}_{hi} = [x_{hi}, y_{hi}, \psi_{hi}, v_{hi}, a_{hi}, r_{hi}\ ]^T \tag{8}$$

In these simulations, the representation of the pursuer pedestrian is manifested as a point within a two-dimensional space. The pursuer of the pedestrian is shown and analyzed under the assumption that it is a circle, and the pose of the pursuer's movement is expressed as an equation (10).

Obstacles with nonlinear trajectories frequently occur in environments such as offices, hospitals, and campuses, as they involve multiple modes of transportation. These obstacles may include individuals engaging in diverse

activities, including walking, running, or using assistive devices like canes or wheelchairs. These obstacles' combined behavior and motion create a distinctive state that defines their interactions.

$$\boldsymbol{S}_{h-so_m} = \left[x_{h-so_m}, y_{h-so_m}, r_{h-so_m}\right]^T \quad (9)$$

The geometry of human movement is approached by a non-holonomic model as expressed in the following equation.

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \dot{\psi}_h \end{bmatrix} = \begin{bmatrix} \cos\psi_h & 0 \\ \sin\psi_h & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_h \\ \omega_h \end{bmatrix} \quad (10)$$

The speed of human motion is expressed as $\boldsymbol{v}_h = [v_h \ \omega_h]^T$. The position and orientation (pose) of the midpoint of the human shoulder are denoted $[x_h, y_h \psi_h]^T$ By serving as a representation of a human being in global coordinates. This equation can structure the transition of human configuration from body coordinates to global.

2.2. Simultaneous Localization and Mapping (SLAM)

SLAM is a technology used in robotics and autonomous systems to map the environment and simultaneously determine an agent's position. SLAM is applicable in various domains, including mobile robotics, autonomous vehicles, and augmented reality. SLAM has proven critical for enhancing the navigation and decision-making capabilities of robots and autonomous systems. The SLAM step illustrates in Fig. 3.

The basic theory of SLAM revolves around two fundamental problems [19]:

1. Localization: Estimating the agent's position and orientation (pose) within the environment.

2. Mapping: Building a depiction of the surroundings, commonly in a 2D or 3D map.

Robots' position (a two-dimensional vector) and orientation (a three-dimensional vector) in relation to their surroundings make up x_k. It is taken from time $k = 0$ and defined as the robot path x_k.

$$X_k = \{x_0, x_1, x_2, x_3, \dots, x_k\} \quad (11)$$

For the relative motion U_k between two-time steps, $k-1$ and k, it is defined as

$$U_k = \{u_0, u_1, u_2, u_3, \dots, u_k\} \quad (12)$$

Relying solely on a single robot's odometry U_k Determining its position within a plane is unsuitable, as in real-world applications, it lacks the necessary precision for accurate localization. The series of sensor measurements Z_k Each time step is defined as follows:

$$Z_k = \{z_0, z_1, z_2, z_3, \dots, z_k\} \quad (13)$$

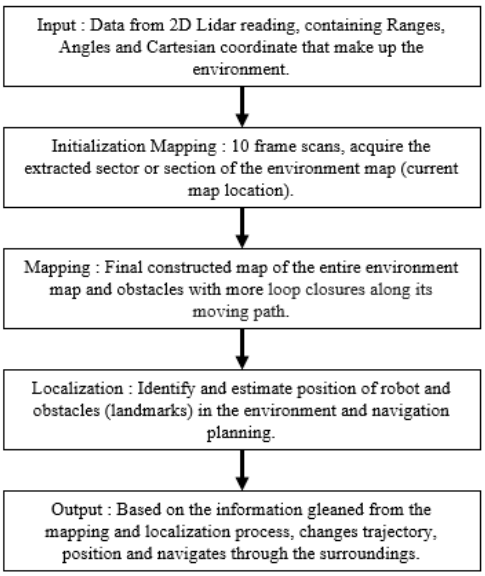

Fig. 3. *General steps SLAM is performed in autonomous mobile robots.*

The following steps entail creating environment representation and predicting the location after collecting and defining all the essential data. By incorporating a probability distribution to estimate the positions of the robot and landmarks based on the created map, the SLAM methodology uses a probabilistic approach. The following diagram illustrates this probability distribution. In order to provide robust and flexible localization and mapping, it captures the uncertainty and variability in the predicted positions. As defined by *P*

$$P(x_K, m | Z_k, U_k) \quad (14)$$

Considering the historical measurements and odometry data, the likelihood of the position at time k and the map may be comprehended. It also calls for a second connection known as the observation model. The observation model determines the following relationship between the robot's position x_k and its odometry u_k.

$$P(x_K | x_{k-1}, u_k) \quad (15)$$

In most applications, a robot can discern the distance to landmarks, their relative orientation, and unique identifiers within a typical environment. The foundation for building a measuring model is this understanding. It can be described as follows by using a probability distribution in the measurement model:

$$P(z_K | x_k, m) \sim N(h(x_k, m), Q_k) \quad (16)$$

Where N is the two-dimensional normal distribution, Q_k is the two-dimensional noise covariance, and $h(x_k, m)$ is an arbitrary function representing the operation of sensory equipment. The following variable is defined below:

k : Time instant.
x_k : Robot location.
X_k : Sequence of robot location or path.
u_k : Odometry between time $k-1$ and k.
U_k : Sequence of robot odometry or relative motion.
Z_k : Sequence of measurements between robot and landmarks.

The SLAM process uses sensor measurements and odometry data as input. It then creates a map of the

environment and finds important points called landmarks. It also figures out where the robot is on this map. The result is a combined map with the exact location of the robot, which helps in efficient navigation.

2.3. Euclidean Clustering

Euclidean clustering is crucial in detecting obstacles by grouping points within a point cloud according to their Euclidean distance. This approach is often used in robotics, autonomous vehicles, and computer vision to identify and separate distinct objects within an environment.

The fundamental theory behind Euclidean clustering involves partitioning the input point cloud into subsets or clusters, each representing a separate object or obstacle. These are achieved by measuring the pairwise Euclidean distance between points and grouping them based on a specified distance threshold.

The Euclidean distance calculates the linear distance between two points in Euclidean space. Given two points, $P(x_1, y_1, z_1)$ and $Q(x_2, y_2, z_2)$, the Euclidean distance between them is calculated as follows:

$$d(P,Q) = sqrt((x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2) \quad (17)$$

Obstacles and environment detected, the gradient for each point in the cluster must be computed. Let xy be a point cloud from LIDAR data. The gradient is calculated as the difference between consecutive points in xy. If G is the gradient matrix of size $(m-1) \times 2$:

$$G_i = xy_{\{i+1\}} - xy_i, for\ i = 1, 2, \ldots, m-1 \quad (18)$$

Then normalize the gradient vectors. Normalize each row of matrix G by dividing it by its Euclidean norm. If N is the normalized gradient matrix of size $(m-1) \times 2$:

$$N_i = \frac{G_i}{||G_i||_2}, for\ i = 1, 2, \ldots, m-1 \quad (19)$$

ALGORITHM 1: Classify Clusters

***Input**: xy - matrix of cluster points, c - cluster index vector, centroids - matrix of cluster centroids.*
***Output:** classification - list of cluster classifications, dimensions - matrix of cluster dimensions.*
***Initialization**: Initialize empty lists 'classification' and 'dimensions.'*
***For** each cluster i in the input data:*
  ***If** the number of points in cluster $i > 1$*
    Compute gradient and angles between consecutive points in xy
    Calculate total_angle and point_count
    ***If** abs(total_angle - pi/2) < 0.1 or point_count >= 10*
      Classify cluster as 'Wall.'
      Compute wall dimensions
    Else
      Classify cluster as 'Square.'
      Compute wall dimensions
  Else
    Classify cluster as 'Circle.'
    Compute mean circle radius
  Else
    Classify cluster as 'Empty.'
***Return** 'classification' and 'dimensions.'*

Compute the angles between consecutive gradient vectors. Calculate the angles between consecutive rows of the normalized gradient matrix N. If A is a vector of size $(m-2) \times 1$, containing the angles:

$$A_i = arccos(N_i \bullet N_{\{i+1\}}), for\ i = 1, 2, \ldots, m-2 \quad (20)$$

Classify the cluster as a wall or circle based on the sum of angles and point count. Calculate the total angle by summing the elements of vector A, and determine the number of points in the cluster, which is equal to the number of elements in the cluster index vector c.

$$total_{angle} = \Sigma A_i, for\ i = 1, 2, \ldots, m-2 \quad (21)$$

$$PC = n, n\ the\ number\ of\ element\ c \quad (22)$$

The algorithm for the detection environment using Euclidean clustering is shown in Algorithm 1.

3. Result and Discussion

Original research papers submitted to the IET Research Journals should conform to the IET Research Journals Length Policy [2]. The length guidelines include the abstract, references and appendices but do not include figure captions, equations, or table content.

This research uses SLAM and Euclidean clustering methods to assess the effectiveness of a differential drive mobile robot navigating a changing environment. The simulation uses the MATLAB toolbox, emphasizing 2D LIDAR sensors for gathering data about the surroundings. The test scenario includes a mobile robot, two moving obstacles, a stationary object as the target, and two squares as the static obstacles. The robot travels at a speed of $0.10\ m/s$, while the dynamic obstacles move at $0.30\ m/s$, enabling the assessment of SLAM and Euclidean clustering algorithms' effectiveness in real-time navigation tasks.

This simulation's primary objective is to evaluate the robot's capacity to create a map, determine its location inside the map, and recognize moving obstacles using LIDAR data. The SLAM technique aids the robot in both determining its location and simultaneously building a map. The Euclidean clustering algorithm, meanwhile, enables the robot to distinguish between moving and stationary objects. The objective is to expand the robot's capacity to travel safely, improve its performance under varied settings, and make it more versatile for real-world applications.

The environment localization and detection of shape and size using SLAM and Euclidean clustering involves acquiring data from a LIDAR 2D sensor, localizing the robot, mapping the environment, and detecting obstacles. SLAM algorithms estimate the robot's pose and create a map, while Euclidean clustering groups point in the point cloud data to identify unique barriers. Shape recognition techniques categorize obstacles based on geometric properties, and the size of each obstacle is determined by computing the dimensions of their bounding boxes or other geometric properties. With this information, the robot can make informed decisions regarding its trajectory and motion, avoiding detected obstacles and safely navigating the environment.

After collecting the data, the subsequent process involves pre-processing by implementing SLAM. The parameter maximum lidar range and map resolution are 18

and 20, respectively. For looping closure threshold and search radius are 320 and 8, respectively. Localization is carried out with initial ten scans from the beginning of the simulation in Fig. 4. For SLAM final building of the environment trajectory of the robot illustrated in Fig. 5. SLAM creates a map of the environment, showing obstacles, landmarks, and other features. This map helps the robot navigate, plan its path, and avoid collisions. The map's accuracy depends on sensor quality, the robot's motion model, and the SLAM algorithm. Two keyframes were identified in a simulation of 1485 frames in one real-time simulation, demonstrating optimal detection performance. These results have been illustrated in Fig. 6 and Fig. 7, where the robotic system successfully detected and identified the obstacles present in the simulation.

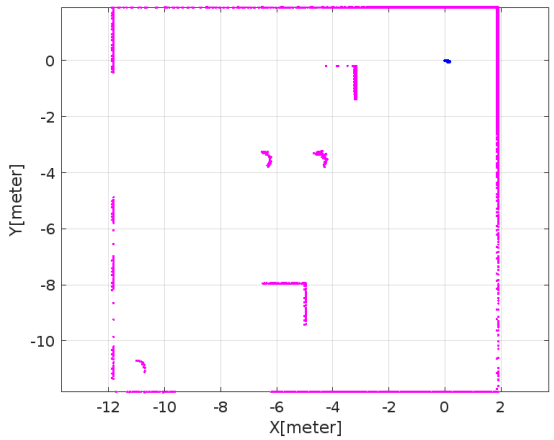

Fig. 4. *Map of the environment pose graph for initial ten scans.*

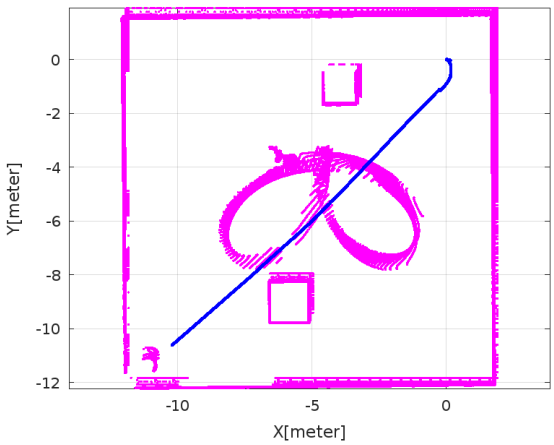

Fig. 5. *Final building of the environment trajectory of the robot.*

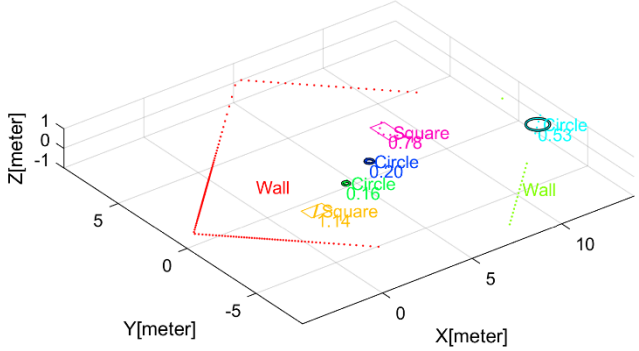

Fig. 6. *Obstacle and environment detection in frame 230*

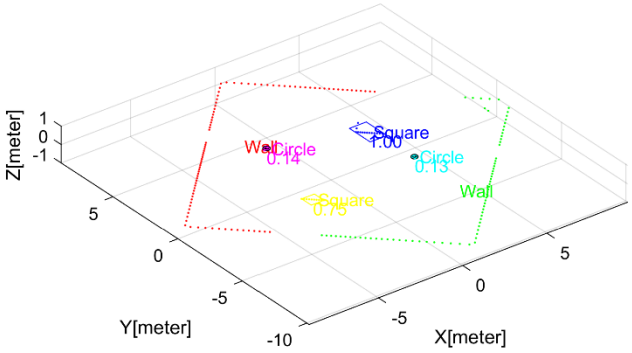

Fig. 7. *Obstacle and environment Detection in frame 700*

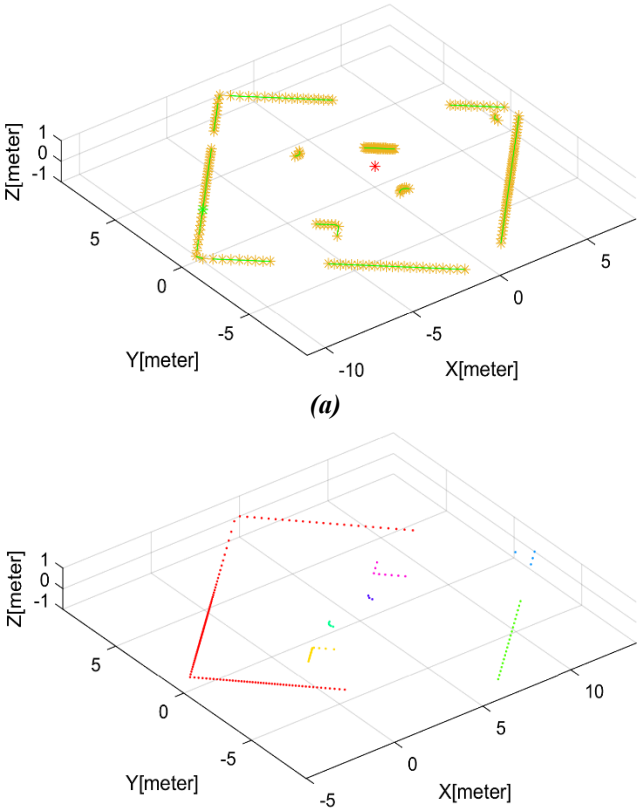

Fig. 8. *Comparison of the two clustering methods* ***(a)*** *K-NN,* ***(b)*** *Euclidean Clustering*

Table 2. Time performance in two different methods

Method	Time (s)
K-NN with K 100	11256
K-NN with K 200	16542
Euclidean Clustering	443.1677

In this simulation, the algorithm was able to detect two distinct objects. The first object is a wall, which serves as a static obstacle delineating the boundaries of the working area. The second object is a circle, representing a dynamic obstacle the robot must navigate. Based on the detected radian values, the algorithm detects the radian values for each dynamic obstacle and approximates their shapes as circles. Fig. 6 demonstrates that dynamic obstacles 1 and 2 were detected with radian values of 0.16 and 0.20, respectively, and two square shapes with sizes 1.14 and 0.78 in centimeters were detected. In contrast, Fig. 7 shows that the same obstacles were detected with radian values of 0.13

and 0.14, respectively, in the two shapes were detected too with different sizes of 0.75 and 1 in centimeters. These variations were caused by the fact that the lidar emission was further away, and the reflected light influenced the item detected during frame 320 and frame 700.

This outcome highlights the effectiveness of the SLAM process and the Euclidean clustering algorithm in identifying and detecting obstacles with varying shapes and sizes. With the successful detection of both static and dynamic obstacles, the robot can safely navigate and operate within its environment. These findings also showcase the potential of these algorithms to be applied in other robotic systems, ultimately improving their performance and adaptability in complex and dynamic environments.

This simulation also simulates clustering with different methods like K-NN. Here, attempting to demonstrate the efficiency and quickness of each method from frame 1 to frame 1485. In K-NN, the value of k, or the number of nearest neighbours to be considered, must be determined. Incorrect k selection can have negative consequences. On the other hand, Euclidean Clustering usually requires setting a threshold distance for grouping points into clusters.

The K value was tested with two types, 100 and 200; however, with Euclidean clustering, point clouds are automatically grouped into clusters based on distance. Table 2 shows that using Euclidean clustering has the best time when processing up to the last frame. The comparison can be seen in Fig. 8.

4. Conclusion

In conclusion, this study presents a comprehensive framework designed explicitly for differential drive mobile robots, addressing the challenges of environmental localization and detection. By modelling the kinematics and dynamics of both the robot and obstacles, we have successfully harnessed the potential of LiDAR data to achieve accurate and robust localization and detection capabilities.

Completing the environmental localization and detection objective for differential drive mobile robots involves a well-defined process. Starting with collecting data from LiDAR sensors, the acquired information undergoes preprocessing to ensure its quality and usability. Localization techniques are then applied to accurately estimate the robot's position and orientation within the environment. Various objects are recognized and segmented using environment detection algorithms, improving the robot's situational awareness. Object classification, enabled by machine learning, further enhances the robot's ability to understand its surroundings. Finally, mapping and visualization techniques are employed to generate informative representations of the environment.

Based on the simulations that have been carried out, it can be proven that this method effectively detects obstacles by knowing the shape and size of each clustering.

5. Acknowledgments

The authors express their gratitude for the support from the Electrical Engineering Department ITS on research facilities and financial support during the completion of this project.

References

[1] T. Agustinah, A. Jazidie, M. Fuad, and F. A. Setiawan, “Evading of Pedestrian Pursuer and Avoiding Obstacles Using Path-Velocity Planner,” *IEEE Access*, vol. 9, pp. 162476–162486, 2021, doi: 10.1109/ACCESS.2021.3133383.

[2] Y. Liu, C. Wang, H. Wu, Y. Wei, M. Ren, and C. Zhao, “Improved LiDAR Localization Method for Mobile Robots Based on Multi-Sensing,” *Remote Sens (Basel)*, vol. 14, no. 23, p. 6133, Dec. 2022, doi: 10.3390/rs14236133.

[3] D. Ghorpade, A. D. Thakare, and S. Doiphode, “Obstacle Detection and Avoidance Algorithm for Autonomous Mobile Robot using 2D LiDAR,” in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, IEEE, Aug. 2017, pp. 1–6. doi: 10.1109/ICCUBEA.2017.8463846.

[4] J. Wu, J. Gao, J. Yi, P. Liu, and C. Xu, “Environment Perception Technology for Intelligent Robots in Complex Environments: A Review,” in *2022 7th International Conference on Communication, Image and Signal Processing (CCISP)*, IEEE, Nov. 2022, pp. 479–485. doi: 10.1109/CCISP55629.2022.9974277.

[5] N. Uddin, H. Nugraha, A. Manurung, H. Hermawan, and T. M. Darajat, “Kinematics Modeling and Motions Analysis of Non-holonomic Mobile Robot,” in *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, IEEE, Aug. 2022, pp. 220–225. doi: 10.1109/ICOIACT55506.2022.9971911.

[6] A. A. Panchpor, S. Shue, and J. M. Conrad, “A survey of methods for mobile robot localization and mapping in dynamic indoor environments,” in *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, IEEE, Jan. 2018, pp. 138–144. doi: 10.1109/SPACES.2018.8316333.

[7] D. Chikurtev, N. Chivarov, S. Chivarov, and A. Chikurteva, “Mobile robot localization and navigation using LIDAR and indoor GPS,” *IFAC-PapersOnLine*, vol. 54, no. 13, pp. 351–356, 2021, doi: 10.1016/j.ifacol.2021.10.472.

[8] A. Elfes, “Using occupancy grids for mobile robot perception and navigation,” *Computer (Long Beach Calif)*, vol. 22, no. 6, pp. 46–57, Jun. 1989, doi: 10.1109/2.30720.

[9] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Boston, MA: Springer US, 1992. doi: 10.1007/978-1-4615-3652-9.

[10] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers, pp. 116–121. doi: 10.1109/ROBOT.1985.1087316.

[11] P. K. Panigrahi and S. K. Bisoy, “Localization strategies for autonomous mobile robots: A review,” *Journal of King Saud University - Computer and*

Information Sciences, vol. 34, no. 8, pp. 6019–6039, Sep. 2022, doi: 10.1016/j.jksuci.2021.02.015.

[12] C. Cadena *et al.*, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.

[13] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics," *MA: MIT Press.*, Cambridge, 2005.

[14] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, May 2011, pp. 1–4. doi: 10.1109/ICRA.2011.5980567.

[15] D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2017, pp. 5067–5073. doi: 10.1109/ICRA.2017.7989591.

[16] B. Douillard *et al.*, "On the segmentation of 3D LIDAR point clouds," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, May 2011, pp. 2798–2805. doi: 10.1109/ICRA.2011.5979818.

[17] S. Pendleton *et al.*, "Perception, Planning, Control, and Coordination for Autonomous Vehicles," *Machines*, vol. 5, no. 1, p. 6, Feb. 2017, doi: 10.3390/machines5010006.

[18] F. A. Setiawan, T. Agustinah, and M. Fuad, "Modified Extremum Seeking Control for Target Tracking and Formation Control in Pursuit-Evasion Game," *JAREE (Journal on Advanced Research in Electrical Engineering)*, vol. 6, no. 2, Oct. 2022, doi: 10.12962/jaree.v6i2.320.

[19] A. R. Khairuddin, M. S. Talib, and H. Haron, "Review on simultaneous localization and mapping (SLAM)," in *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, IEEE, Nov. 2015, pp. 85–90. doi: 10.1109/ICCSCE.2015.7482163.