American Academic Scientific Research Journal for Engineering, Technology, and Sciences
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402
http://asrjetsjournal.org/

Autonomous Mobile Robot Motion Control for Hospital Disinfection

Qassim Haichel[a*], Ahmed Rahman[b]

[a]*Researcher, Baghdad University, Al- Khwarizmi College of Engineering, Iraq*
[b]*Assistant Professor. Ahmed Rahman, Baghdad University, Al- Khwarizmi College of Engineering, Iraq*
[a]*Email: Qassim.haichel@gmail.com*
[b]*Email: ahmedalmusawi@kecbu.uobaghdad.edu.iq*

Abstract

As our society develops, robotic systems become indispensable. In producing automation can be used for scanning the impure areas in hospital. In this article MATLAB's "Robotics System Toolbox," to simulate robot navigation. This essay attempts to demonstrate the efficacy of two path planning techniques: pure-pursuit and the probabilistic roadmap (PRM). To compare the performances of four maps, whose difficulty was gradually increased. For PRM, the number of nodes was first set after the map had been loaded. Initial and final positions were then established. Following that, the program built a possible network of links between the nodes at the start and goal locations. Finally, the algorithm analyzed this network of connected nodes to return a collision-free path. In pure-pursuit, the algorithm's main goal is to select a suitable look-ahead distance. In its most basic form, The Pure Pursuit algorithm examines the difference in heading between the current vehicle and the objective point along the course. It is a proportional controller. The effectiveness of the Pure Pursuit algorithm implementations was tested in a variety of situations. The algorithm used in all of the tests followed a straight path between high level waypoints. It's necessary to keep in mind that PRM path position was the only navigation sensor employed in these studies when analyzing their results.

Keywords: mobile robot; Probabilistic Roadmap (PRM); pure pursuit; Kinematic Model.

1. Introduction

Robotic system becomes a necessary part of our society. In producing automaton have utterly modified the approach merchandise are created and reduced the cost of production, can also be used for scanning the impure areas with radiation or venomous materials, however this is often not the top of robotic technology for several reasons like human safety or economics[1].

---

* Corresponding author.

It is fascinating to possess robot ability to finish advanced tasks with lowest or human intervention; however, this is often still far away from being accomplished.Moving an automaton may be a complex operation that needs plenty of information regarding the environment; this information should be analyzed and taken to conduct the required task. In most cases, the goal is to soundly move the robot to a target position to perform associate action; the action of safe travels in steps is termed navigation. the various parameters make the algorithm are effective; attaining an aim position and heading off nearby obstacles[2].The creation of autonomous systems that are capable of interacting with real settings without the aid of a human operator is one of the main goals of mobile artificial intelligence[3]. There are many difficulties in achieving this goal. This is primarily due to the fact that real environments are typically uncertain, unknown, and dynamic; as a result, the knowledge that is already known about these environments may be absent, incomplete, uncertain, or imprecise. In addition, these environments may change over time, becoming more complex and unpredictable [1].The current challenges within the developments of automaton management systems are creating them capable of intelligent and appropriate responses to ever-changing environments. Learning and adaptation methods, moreover as decision-making techniques facilitate to realize these objectives. Nevertheless, it is technologically difficult and probably dangerous to create complicated systems that are self-controlled, since the system can fail if the system does not work. With decentralized control, it is attainable for the system to continue working even once a part of it fails. Although centralized control allows multiple goals and constraints to be coordinated in a coherent manner, decentralization provides flexibility and robustness [4].

Figure 1: Pioneer 3-DX [5]

Kuma and his colleagues [6] present the implementation of the pure pursuit algorithm employing probabilistic roadmaps -PRM- for robot navigation. The surroundings of the robot are visualized using an occupancy grid. The occupancy grid map contains the PRM. Probabilistic roadmaps are used to determine the preferred route for robot navigation from the starting point to the destination. The Gazebo simulator is used for experimental work with a Turtlebot robot simulation. The MATLAB programming language is used for programming. The "Robotic System Toolbox" in MATLAB is also used to program the robot's navigation system. The effectiveness of two path-planning methods, the PRM and the genetic algorithm (GA), is examined in a study by ***Martin C. and his colleagues*** [7]. To evaluate how well a simple map and a complicated map perform in comparison to one another. In the PRM, the number of nodes was first given after the map had been loaded. The initial and final positions were then determined. In order to connect the starting point and the final position, the program then built a network of probable node connections. To find a collision-free path, the program then

searched across this network of connected nodes. First, a map was loaded into GA, after which the GA settings were selected. It was examined to see which combination of values for these GA parameters would best address the problem. The beginning and ultimate placements were then defined. The length or total number of segments was the related cost for each path that was created. Penalties were imposed each time an obstruction appeared on the path that was generated. Results show that both strategies moved in a collision free path within the given environment or map from their starting point to their final position. However, it was discovered that each strategy had both advantages and disadvantages. While -GA- offers smoother routes that make it simpler for mobile robots to travel, it takes more time to analyze information, making it difficult to use in real-time navigation. To present the probable path in a noticeably shorter amount of time, PRM forgoes navigational smoothness; yet, this makes it effective in more reactive settings. Discussing the advantages and disadvantages of the two methods demonstrates how critical it is to select the ideal path planning algorithm for the application at hand.The paper by ***Samuel and his colleagues*** [8] gives a quick summary of a few common path tracking techniques used in the development of autonomous vehicles. A thorough comparison was carried out between these geometric techniques, including pure-pursuit techniques, vector pursuit techniques, and CF-pursuit techniques, all of which are based on pure-pursuit techniques. This review article also emphasizes subjects on which little research has been done. regions include tracking a mobile robot's implicit component and recommending a field for feature research, like tracking a non-holonomic mobile robot's implicit and explicit path.

2. Kinematic Model of The Differential Drive Wheeled Mobile Robot

Two independently driven wheels on this vehicle show in figure (2) allow it to independently manage its forward and angular velocities.

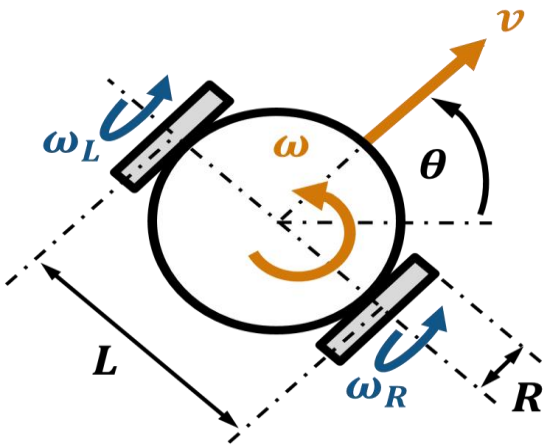

Figure 2: Vehicle[9]

Left and right wheel speeds ω_L and ω_R , in rad/s

Linear velocity v, in m/s

Angular velocity ω, in rad/s

Forward Kinematics

$v = \frac{R}{2}(\omega_R + \omega_L)$, $\omega = \frac{R}{2}(\omega_R - \omega_L)$

Inverse Kinematics

$\omega_L = \frac{1}{R}(v - \frac{\omega_L}{2})$, $\omega_R = \frac{1}{R}(v + \frac{\omega_L}{2})$

By MATLAB Usage:

Create a Differential Drive object

R = 0.1; Wheel radius [m]

L = 0.5; Wheelbase [m]

vehicle = DifferentialDrive(R,L)

Solve forward kinematics

ω_L = 1;

ω_R = -0.5;

[v, w] = forwardKinematics(vehicle, ω_L, ω_R)

Solve inverse kinematics

vRef = 0.2;

wRef = 0.5;

[ω_L, ω_R] = inverseKinematics(vehicle,vRef,wRef)[9].

3. Probabilistic Roadmap

Based on free and occupied places, the Probabilistic Roadmap (PRM) gives a network graph of potential paths in a given environment. Based on the parameters given in the PRM algorithm, nodes are produced and connections are created in this manner [10]. The number of nodes used influences the algorithm's ability to adapt to any level of environmental complexity and depends on the particular challenge at hand. The method uses the network of interconnected nodes to locate a path for the mobile robot that avoids collisions [6,11].

Making a map of the area, storing it in computer-readable form, determining a possible path from point A to point B on the recorded map, and finally moving the robot along the determined path are the four main processes in the robot navigation process[12]. Three steps are involved in the map-building process: choosing the right two-dimensional coordinates in the environment world to cover the entire working area, driving the robot in the direction of these chosen coordinates, receiving sensor readings, and marking the free and occupied spaces in the world by setting the probabilities of occupied spaces as one and the probabilities of free spaces as zero in the occupancy grid framework. This occupancy grid is easily reusable and may be directly stored and used in robot path planning[7].utilized in the robot's navigation the occupancy grid representation of the environment global map. The occupancy grid map is computer readable, thus a computer program can use it to find a path free of obstacles. Finding the best practical route from the starting point to the destination site is the process of path planning in robot navigation. In an occupancy grid map, a path from the starting point to the goal point can be found using the probabilistic roadmaps (PRM) technique. In the PRM technique, the specified number of nodes are formed in the grid's open spaces before being connected to one another within a predetermined connection distance. Additionally, one path among the potential connected paths from the start to the goal is chosen[13].

4. The Pure Pursuit path Tracker

A technique called Pure-Pursuit can be used to geometrically determine the arc required to get a vehicle onto a path. The approach is straightforward, simple, and simple to use. The algorithm's main goal is to select an appropriate look-ahead distance [8]. It is comparable to human driving in that people set a look-ahead distance for their vehicles and then steer to get them to that location. Path tracking has seen extensive application of the Pure Pursuit algorithm. In Figure (3), the algorithm is presented [14].

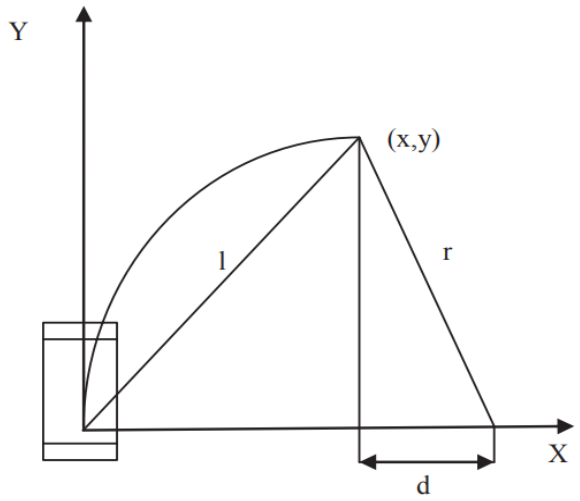

Figure 3: The pure-pursuit mode's geometry [15]

The coordinate system of the machine is built using the x and y axes. The machine is some distance away from the point (x, y). The L represents the arc's cord's length.

linking the point and the source (x, y). The arc's curvature's radius is r. The following is the relationship between x, L, and r [16]:

$$D + x = r \quad (1)$$

$$D^2 + y^2 = r^2 \quad (2)$$

$$x^2 + y^2 = L^2 \quad (3)$$

From Eq. (1), Eq. (2) and Eq. (3),

$$R^2 - 2rx + x^2 + y^2 = r^2 \quad (4)$$

$$r = \frac{L^2}{2x} \quad (5)$$

By choosing a look-ahead distance and calculating the path error x, the radius of the curvature required to get the machine on the required path can be calculated[14].The curvature of an arc is given as $\gamma = \frac{1}{r}$ so we can rewrite Equation 5 as:

$$\gamma = \frac{2x}{l^2} \quad [17](6)$$

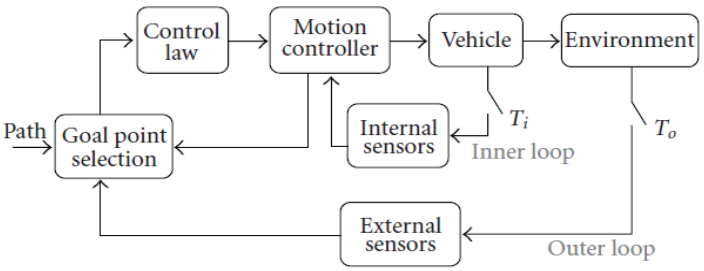

Figure 4: Block diagram of the pure-pursuit path tracker[18]

The Pure Pursuit algorithm, in its simplest form, is a proportional controller that compares the heading of the current vehicle to the heading along the way to the destination location.

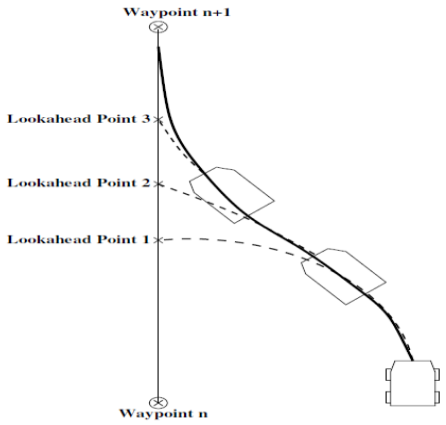

Figure 5: With a longer lookahead distance, there is less overshoot when reentering the path following a substantial separation [17]

This is demonstrated by using a different technique to display the curvature formula. From Figure3, we see that $\sin(\theta err) = \frac{x}{l}$, so for small heading errors $\theta err \approx \frac{x}{l}$. Substituting this into Equation 6, we get:

$$\gamma = \frac{2\theta_{err}}{l} \qquad [17](7)$$

This demonstrates that the lookahead distance, or l, is the algorithm's sole parameter. The algorithm is incredibly simple to implement and fine-tune as a result. The lookahead distance can be tuned to change a variety of performance aspects. A path's maximum curvature can be tracked more accurately and with more precision when the lookahead distance is less. A lower lookahead distance also makes the vehicle return to the path more quickly after becoming separated. But there are some compelling arguments for lengthening it:

- While tracking a path, oscillations are lessened by a greater lookahead distance.
- It enables the mobile start turning before arrive the curve on paths with steep curves, producing smoother trajectories.
- Because the resulting bends are less sharp, the requested steering changes are less sudden, It is excellent for vehicles with a lot of steering latency or those traveling at higher speeds.
- When reentering the path after a significant separation, there is less overshoot with a higher lookahead distance [17].

5. Waypoint Navigation

The robotic vehicle is given a patrol mission by the mission commander in a typical waypoint navigation scenario, and the road to be taken is described as a collection of waypoints separated by tens or hundreds of meters. Without assistance from a driver, the autonomous vehicle travels between successive destinations. For a waypoint to be considered attained, the robot must pass it at a distance that is less than or equal to the waypoint's radial tolerance. The path that the algorithm follows will be the straight section between the previous and next waypoints, Figure (5).

The Pure Pursuit algorithm is typically implemented using a detailed path that is represented in the literature as a collection of closely spaced nodes. However, the user won't want to specify a very specific path for many programs. Each of the waypoints provided to the Pure Pursuit waypoint implementation presented here is regarded as an interim goal location, and the fundamental algorithm monitors the straight-line segments between successive pairs of waypoints. Although a list of all the waypoints is kept, the algorithm only ever works between the current and the following waypoints at any one moment[17].

5.1. Building Map of the Environment

It is possible to create a two-dimensional occupancy grid for the environment map. An occupancy grid can be

made in MATLAB by using the instructions below:

```
image = imread('image.png');

grayimage = rgb2gray(image);

bwimage = grayimage < 0.5;

grid = binaryOccupancyMap(bwimage);

map= grid

show(map)
```

As indicated in table (1), this study used four maps, divided into simple and complex maps:

Table 1: maps parameters

	Case 1	Case 2	Case 3	Case 4
Grid Size	[52 50]	[52 50]	[52 50]	[40 74]
Resolution	1	1	1	1
X World Limits	[0 50]	[0 50]	[0 50]	[0 74]
Y World Limits	[0 52]	[0 52]	[0 52]	[0 40]
Grid Origin in world	[0 0]	[0 0]	[0 0]	[0 0]

PRM were implemented in Matlab R2021a. In order to show how well the methodologies were applied to maps in order to show how well performed at path planning. a map was loaded first, then the quantity of nodes was displayed. Start and goal location were then established. The program then created a net of potential node connections between the starting place and the ending position. In order to return a collision-free path, the program lastly explored this network of connected nodes. The PRM settings used for the maps are listed in Tables (2).

Table 2: PRM parameters

	Case 1	Case 2	Case 3	Case 4
Start Location	[10 20]	[41 10]	[10 10]	[5 5]
Robot Goal	[30 32]	[20 10]	[40 15]	[60,15]
Num Nodes	50	100	150	200

6. Experimental Work & result

To find the probabilistic roadmaps, the occupancy grid map of the environment imported to the MATLAB workspace and using instruction follow:

```
mapInflated = copy(map);
```

```
inflate(mapInflated, robot.TrackWidth/2);

prm = robotics.PRM(mapInflated);

prm.NumNodes = N;

prm.ConnectionDistance = D;

startLocation = [x y];

endLocation = [x  y];

path = findpath(prm, startLocation, endLocation)

show(prm);
```

The mobile robot navigates and avoids obstacles to reach the destination location by employing the PRM and pure pursuit algorithm technique. In any case the first step is got probabilistic roadmap to find the best path to move the robot to the goal point from the home point and navigate through the path by pure-pursuit algorithm. In case of (50) numbers of nodes are used we didn't get the path because nodes don't connect. By trying PRM again to have the best path with nodes connected. In case with (50) numbers of nodes are used we didn't get the path because nodes don't connect as shown in Figure (6 II). By trying PRM again with (100) nodes to have the best path with nodes connected. The map has more doors and lanes in case III with (50) numbers of nodes are used we didn't get the path because nodes don't connect as shown in Figure (6 III). By trying PRM again with (100) nodes didn't have the path with nodes unconnected. By trying PRM again with (150) nodes to have the best path with nodes connected. In case (IV) The most complex map with (50) numbers of nodes are used we didn't get the path because nodes don't connect as shown in. By trying PRM again with (100) nodes didn't have the path with nodes unconnected. So, we trying PRM again with (150) nodes didn't have the path with nodes unconnected. By trying PRM again with (200) nodes result the best path with nodes connected as show in Figure (6 IV).

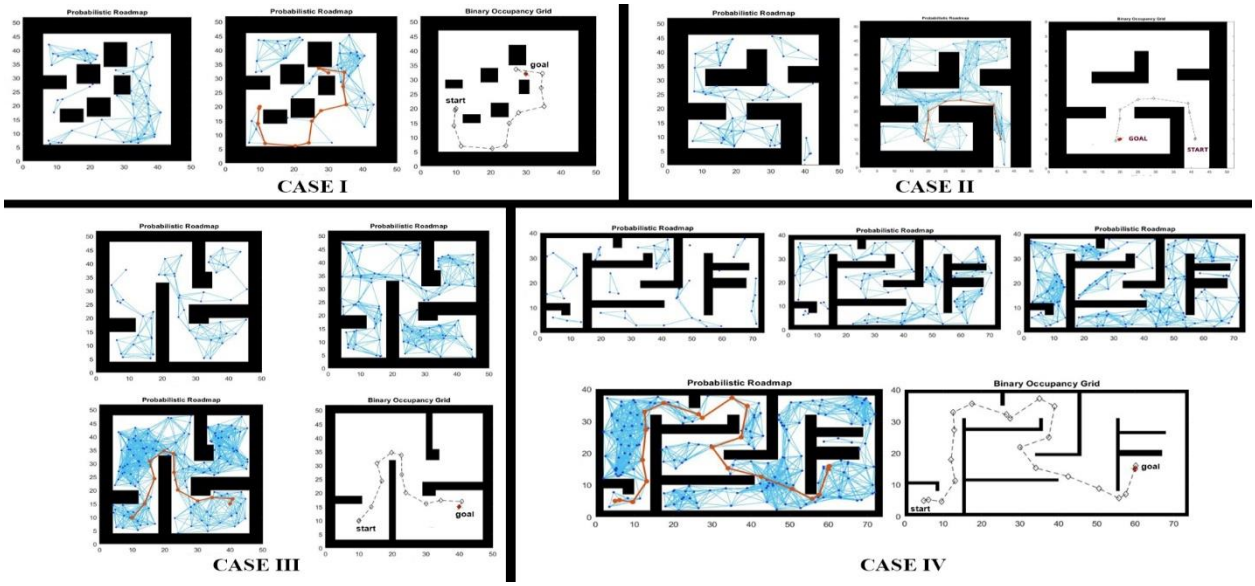

Figure 6: Maps & PRM

7. Conclusions

A solution can be efficiently reached in cases where the environment map is simple (i.e., there are fewer obstacles), has a small number of nodes, and has a long connection distance between nodes. We can see the impact of using a varying number of nodes in PRM by using a fixed connection distance between the nodes.

Both strategies (PRM and pure-pursuit) have successfully shown that they are effective at helping mobile robots plan their routes. They have provided collision-free navigation between the predetermined initial and final locations inside the designated environment or map. This allows Control mobile robot for Hospital Disinfection by using UV light for example.

However, the PRM method observed benefits and drawbacks. The resulting trajectories show that more nods were required, which improves the mobility of the mobile robots but takes more time. Also, the advantage of PRM is accurate results with a large number of nodes.

Once we have a realistic path from the starting point to the goal point in the environment, we can simply follow the waypoints of the path to get to the goal location. Using MATLAB's "Robotics System Toolbox," a path following controller object built on a pure-pursuit process can be produced.

In pure-pursuit, the algorithm's main goal is to select a suitable look-ahead distance. In its most basic form, by measuring the difference in heading between the present vehicle and the objective point along the course, the Pure Pursuit algorithm is a proportional controller. The effectiveness of the Pure Pursuit algorithm implementations was tested in a variety of situations. The algorithm used in all of the tests followed a straight path between high level waypoints, it’s necessary to keep in mind that PRM path position was the only navigation sensor employed in these studies when analyzing their results.

This work's limitation is that it only applies to environments with static obstacles.

References

[1]M. Mucientes, R. Iglesias, C. v Regueiro, A. Bugarã, and S. Barro, "A fuzzy temporal rule-based velocity controller for mobile robotics," 2003. [Online]. Available: www.elsevier.com/locate/fss

[2]F. Diaz-Hermida, M. Pereira-Fariña, J. C. Vidal, and A. Ramos-Soto, "Characterizing Quantifier Fuzzification Mechanisms: a behavioral guide for practical applications," May 2016, doi: 10.1016/j.fss.2017.07.017.

[3]H. Khayyam, B. Javadi, M. Jalili, and R. N. Jazar, "Artificial Intelligence and Internet of Things for Autonomous Vehicles," in *Nonlinear Approaches in Engineering Applications*, Springer International Publishing, 2020, pp. 39–68. doi: 10.1007/978-3-030-18963-1_2.

[4]B. Innocenti, B. López, and J. Salvi, "A multi-agent architecture with cooperative fuzzy control for a mobile robot," *Rob Auton Syst*, vol. 55, no. 12, pp. 881–891, Dec. 2007, doi: 10.1016/j.robot.2007.07.007.

[5]"Pioneer 3 Operations Manual with MobileRobots Exclusive Advanced Robot Control & Operations Software," 2006.

[6]A. Szakál, IEEE Hungary Section, M. IEEE Systems, and Institute of Electrical and Electronics Engineers, *CINTI 2016 : 17th IEEE International Symposium on Computational Intelligence and Informatics : proceedings : 2016 November 17-19, Budapest*.

[7]*2017 IEEE National Aerospace and Electronics Conference (NAECON) : 27-30 June 2017*.

[8]M. Samuel *et al.*, "A Review of some Pure-Pursuit based Path Tracking Techniques for Control of Autonomous Vehicle," 2016.

[9]"matlab." https://www.mathworks.com/help/robotics/ug/path-following-for-differential-drive-robot.html#PathFollowingControllerExample-2 (accessed Aug. 16, 2022).

[10] X. Z. Han *et al.*, "Path-tracking simulation and field tests for an auto-guidance tillage tractor for a paddy field," *Comput Electron Agric*, vol. 112, pp. 161–171, Mar. 2015, doi: 10.1016/j.compag.2014.12.025.

[11] T. Elmokadem and A. v. Savkin, "A hybrid approach for autonomous collision-free uav navigation in 3d partially unknown dynamic environments," *Drones*, vol. 5, no. 3, Sep. 2021, doi: 10.3390/drones5030057.

[12] H. Y. Zhang, W. M. Lin, and A. X. Chen, "Path planning for the mobile robot: A review," *Symmetry*

(Basel), vol. 10, no. 10, 2018, doi: 10.3390/sym10100450.

[13] A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "HPPRM: Hybrid Potential Based Probabilistic Roadmap Algorithm for Improved Dynamic Path Planning of Mobile Robots," *IEEE Access*, vol. 8, pp. 221743–221766, 2020, doi: 10.1109/ACCESS.2020.3043333.

[14] IEEE Staff, *2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)*. IEEE, 2017.

[15] Institute of Electrical and Electronics Engineers, *2016 International Conference on Next Generation Intelligent Systems (ICNGIS)*.

[16] H. Wang, X. Chen, Y. Chen, B. Li, and Z. Miao, "Trajectory Tracking and Speed Control of Cleaning Vehicle Based on Improved Pure Pursuit Algorithm*."

[17] J. Giesbrecht, D. Mackay, J. Collier, S. Verret, and D. Suffield, "Path Tracking for Unmanned Ground Vehicle Navigation Implementation and Adaptation of the Pure Pursuit Algorithm Defence Research and Recherche et développement Development Canada pour la défense Canada," 2005.

[18] J. Morales, J. L. Martínez, M. A. Martínez, and A. Mandow, "Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2D laser scanner," *EURASIP J Adv Signal Process*, vol. 2009, 2009, doi: 10.1155/2009/935237.