PERTANIKA
JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Analysis of Pure-pursuit Algorithm Parameters for Nonholonomic Mobile Robot Navigation in Unstructured and Confined Space

Izzati Saleh[1], Azwati Azmin[2], Azan Yunus[3] and Wan Rahiman[1,4,5]*

[1]School of Electric & Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, 14300 USM, Nibong Tebal, Pulau Pinang, Malaysia
[2] Faculty of Electrical Engineering, Universiti Teknologi Mara, Cawangan Pulau Pinang, 13500 UiTM, Permatang Pauh, Malaysia
[3]CL Solution Sdn Bhd., Wisma Rampai, 59200 Kuala Lumpur, Malaysia
[4]Cluster of Smart Port and Logistics Technology (COSPALT), Universiti Sains Malaysia, Engineering Campus, 14300 USM, Nibong Tebal, Pulau Pinang, Malaysia
[5]Daffodil Robotics Lab, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh

ABSTRACT

This research analyses Pure-pursuit algorithm parameters for nonholonomic mobile robot navigation in unstructured and constrained space. The simulation-based experiment is limited to the mobile robot arrangement. The Look Ahead Distance parameter is adjusted so the mobile robot can navigate the predefined map closely following the waypoints. The optimal Look Ahead Distance value is combined with the VFH+ algorithm for obstacle avoidance. The method is enhanced by adding the λ weight so the robot returns to its waypoints after avoiding an obstacle. The investigation reveals that λ influences the mobile robot's capacity to return to its predetermined waypoints after avoiding an obstacle. Based on the simulation experiment, the optimal LAD value is 0.2 m, and the optimal λ value is 0.8.

Keywords: Mobile robot navigation, obstacle detection, path following, VFH+ algorithm

INTRODUCTION

Typical mobile robot navigation starts with the mobile robot taking input from sensors from the surroundings. From these data, the mobile robot can localise itself and generate a feasible path to navigate itself. The navigation path could be planned based on a predefined global path or from a local

ARTICLE INFO

Article history:
Received: 29 December 2022
Accepted: 20 March 2023
Published: 06 November 2023

DOI: https://doi.org/10.47836/pjst.32.1.06

E-mail addresses:
izzatisaleh@student.usm.my (Izzati Saleh)
azwati3025@uitm.edu.my (Azwati Azmin)
mkay@clsolution.com.my (Azan Yunus)
wanrahiman@usm.my (Wan Rahiman)
* Corresponding author

ISSN: 0128-7680
e-ISSN: 2231-8526

© Universiti Putra Malaysia Press

planner. Based on the generated path information, the mobile robot will send signals to the microcontroller to move the robot.

Path-following is defined as a vehicle following a globally determined geometric path using steering motions to direct it along that path (Snider, 2009). Several path-following methods for ground vehicles have been adapted to mobile robots. However, a simpler yet effective mobile robot path-following method would be the geometric path-following method. One of the most used path-following methods would be the pure pursuit algorithm (Pure-pursuit).

The Pure-pursuit was initially used for missiles to pursue its target (hence the name). The method was then translated to the application for a ground vehicle by Coulter (1992), and in recent years the application expanded to nonholonomic mobile robot and autonomous vehicle applications (Ahn et al., 2021; Girbés et al., 2011; Huang et al., 2020; Li et al., 2019; Qinpeng et al., 2019; Shan et al., 2015; Wang et al., 2019; Wang et al., 2017; Yang et al., 2022). Since then, several improvements and modifications of Pure-Pursuit have been undertaken based on specific configurations of their respective mobile robot or autonomous vehicles. Research by Girbés et al. (2011) proposed a multi-level control scheme by considering different dynamics with different sampling frequencies, Shan et al. (2015) replaced circles on traditional Pure-Pursuit with clothoid C curve to reduce fitting error and Wang et al. (2017) who calibrated the heading and steering angle of the vehicle and reduced lateral error when the vehicle was following an ideal path with an improved accuracy by 54.54%.

Both research by Chen et al. (2018) and Li et al. (2019) proposed to include a PI (Proportional Integral) controller to address tracking errors in extreme drive conditions. Chen et al. (2018) also paired the algorithm with a low-pass filter to smooth the final output steering angle. Meanwhile, Li et al. (2019) used PID (Proportional Integral Derivative) to facilitate steering angle calculation when using the Pure-Pursuit approach.

Research by Wang et al. (2019) proposed a dual-stage fuzzy logic controller to adjust the mobile robot speed and Look-ahead distance in Pure-Pursuit to ensure the robustness and stability of the system. On top of that, Ahn et al. (2021) proposed a method of selection of Pure-Pursuit Look-ahead point heuristically based on the relationship between the vehicle and the path. Finally, Yang et al. (2022) proposed an algorithm that deduced the Look-ahead behaviour and scanned the area for the ideal goal point based on the evaluation function. The research objective is to minimise lateral and heading errors to achieve adaptive optimisation of the target location.

Paper Objective

This paper aims to determine the optimal value of parameters affecting the trajectory of a differential drive mobile robot (later referred to as a mobile robot) in an unstructured and confined space. The first parameter to be tuned is the Look-Ahead distance from Pure-Pursuit. The best value is then tested on the pure pursuit controller and integrated with an obstacle avoidance algorithm, Vector Field Histogram (VFH+).

The second parameter, λ, is introduced to ensure the mobile robot converges into the designated waypoints immediately after avoiding an obstacle. The integration of these two parameters is further tuned, and the best parameter is deduced. The tuning of the parameters is done in simulation and is limited to the mobile robot configuration and inside a pre-set map.

MOBILE ROBOT NAVIGATION

Pure-pursuit Algorithm

Pure-pursuit is a vehicle tracking algorithm that measures the curvature that drives a vehicle from its current location to a target position. The pure pursuit algorithm geometrically evaluates the curvature that will move the mobile robot to a target point. Figure 1 shows the geometry diagram of a Pure-pursuit adapted from Coulter (1992).

From Figure 1, by assuming the current location of the mobile robot in the global coordinate system is in origin $(x_{robot}, y_{robot}) = (0,0)$, let l be the Look-ahead Distance and (l_x, l_y) be the current Look-ahead point, whereby l is the hypotenuse of the right-angled triangle (Equation 1). The relationship between the radius of the arc r that joins the current location of the mobile robot (x_{robot}, y_{robot}) with the Look-Ahead Distance point was explained in Equation 2, whereby d is represented by Equation 3.

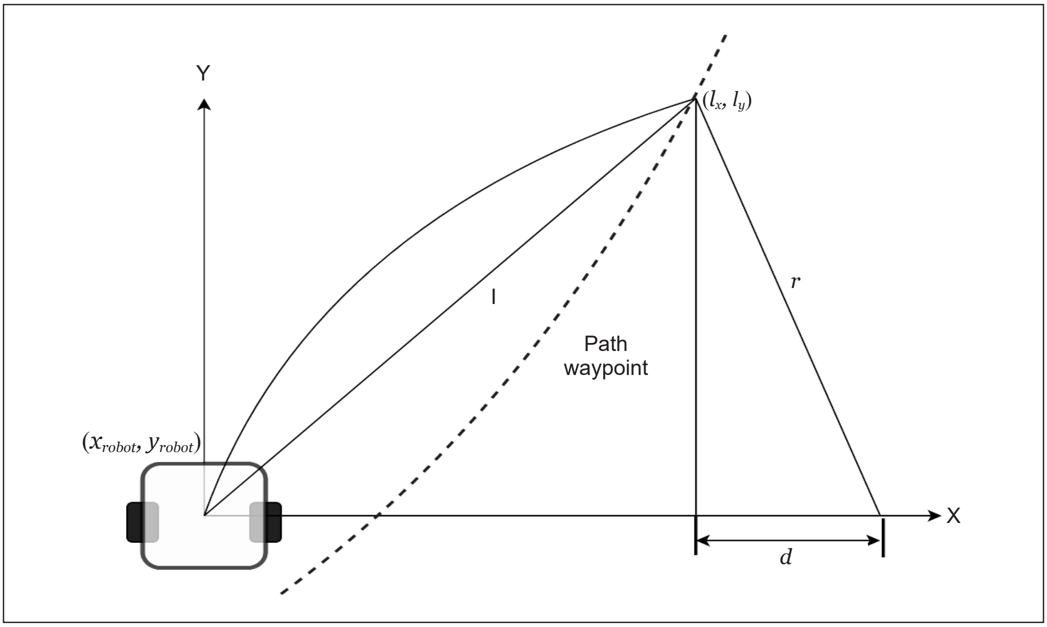

Figure 1. Geometry diagram of Pure-pursuit

$$l_x^2 + l_y^2 = l^2 \quad [1]$$

$$d^2 + l_y^2 = r^2 \quad [2]$$

$$l_x + d = r \quad [3]$$

By substituting *d* from Equation 3 to Equation 2, we can obtain arc radius *r* (Equations 4, 5 and 6):

$$(r - l_x)^2 + l_y^2 = r^2 \quad [4]$$

$$r^2 - 2rl_x + l_x^2 + l_y^2 = r^2 \quad [5]$$

$$r = l^2/2l_x \quad [6]$$

The r value determines the actual arc radius the vehicle will follow. The curvature of that radius is its reciprocal value $\left(\frac{1}{r}\right)$.

Figure 2 shows the Pure-Pursuit. Firstly, the waypoints of the mobile robot are established. The mobile robot will localise itself inside a global map. During every loop, the algorithm iterates to find the current position of the mobile robot, a new look-ahead point, and, consequently, the current arc radius. The algorithm will transform the goal point (l_x, l_y) to the mobile robot coordinate and consequently calculate the steering angle for the mobile robot to steer back into its path.

```
Data: Waypoints
Result: Robot Trajectory
Initialization;
while distanceToGoal>goalRadius do
    find current robot position ($x_{robot}$, $y_{robot}$);
    find path point closest to robot position;
    find lookhead point ($l_x$, $l_y$);
    transform goal point into robot coordinates;
    calculate the nagular velocity ($\omega$) to steer back into path;
    update new robot position;
    update distanceToGoal;
    if distanceToGoal ≤goalRadius then
        end search;
    else
        go back to the beginning of loop;
    end
end
```

Figure 2. Pure-pursuit algorithm

Look Ahead Distance

There are two major goals when the mobile robot tracks the path: regaining the mobile robot's position to the designated path and maintaining its position in the path. In the pure pursuit algorithm, one crucial parameter needs to be tuned: Look-ahead Distance (LAD). This parameter affects how far the mobile robot perceives the sets of waypoints. A small LAD is used for the mobile robot to follow the path of the waypoints closely. However, when the LAD is too small, the robot will overshoot the path and oscillate along the desired path. A larger LAD can be chosen for the mobile robot to converge to produce a smoother path gradually, but the robot might have difficulty manoeuvring into a small area due to larger curvatures near the corners**.**

Vector Field Histogram (VFH+)

Vector Field Histogram (VFH) is an algorithm that calculates a mobile robot's obstacle-free steering direction (Bolbhat et al., 2020; Díaz & Marín, 2020; Dong et al., 2021; Pappas et al., 2020; Ulrich & Borenstein, 1998; Ulrich & Borenstein, 2000). To identify the location and proximity of obstacles, range sensor readings are used to compute polar density histograms. Unlike VFH, which is very goal-oriented and provides only one solution of steering direction, VFH+ determines a set of possible candidate directions based on all openings in the masked polar histogram. There is another extension of VFH+, which is VFH*, that plans the waypoints based on the A* approach; however, since, in this case, the waypoints were predefined, the former one was used instead. These candidate directions are then subjected to a cost function considering more than just the difference between the candidate and target directions. An opening is considered wide if the difference between its borders is larger than the maximum number of sectors s_{max}. For a narrow opening, there is only one candidate direction, c_n, and this can be represented by Equation 7:

$$c_n = \frac{k_r + k_l}{2} \qquad [7]$$

There are two candidate directions for a wide opening: either on the left side c_l or on the right side c_r. Should the target direction lie between these two candidates, it can also be considered the third candidate c_t (Equation 8).

$$\begin{cases} c_r = k_r + s_{max}/2 \\ c_l = k_l - s_{max}/2 \\ c_t = k_t \quad , k_t \in [c_r, c_l] \end{cases} \qquad [8]$$

The cost function for a candidate g(c) can be represented with Equation 9:

$$g(c) = \mu_1 . \Delta(c, k_1) + \mu_2 . \Delta\left(c, \frac{\theta_n}{\alpha}\right) + \mu_3 . \Delta(c, k_{d,n-1}) \qquad [9]$$

and $\Delta(c_1, c_2)$ is a function that computes the absolute angle difference between two sectors, c_1 and c_2 (Equation 10):

$$\Delta(c_1, c_2) = min\{|c_1 - c_2|, |c_1 - c_2 - 360°/\alpha|, |c_1 - c_2 + 360°/\alpha|\} \qquad [10]$$

where α is the angular resolution of the histogram, θ_n is the current orientation, k_t is the target direction divided by α and $k_{d,n-1}$ is the previously selected direction of motion/α.

The terms μ_1, μ_2, and μ_3 are accountable for how the mobile robot directs when facing an obstacle. The higher the μ_1, the closer the mobile robot's steering direction to the goal point. Also, a higher μ_2 value produces an efficient path, while a higher μ_3 value ensures less oscillation in steering movements. The following condition must be satisfied to ensure the steering direction follows the goal direction (Equation 11):

$$\mu_1 > \mu_2 + \mu_3 \qquad [11]$$

METHODOLOGY

Differential Drive Mobile Robot Kinematics

The simulation was coded using MATLAB software. To simulate a simplified vehicle model of a differential-drive mobile robot, a *differentialDriveKinematics* object (Figure 3) creates a differential-drive vehicle model. The model approximates a vehicle with a single fixed axle and wheels separated by a specified track width. For differential drive, the wheels were controlled independently. The speed and heading are defined from the axle centre. The vehicle state is defined as $[x_{robot} \quad y_{robot} \quad \theta_{robot}]$, the global coordinate inside a map measured in metres, whereas the heading was measured in radians.

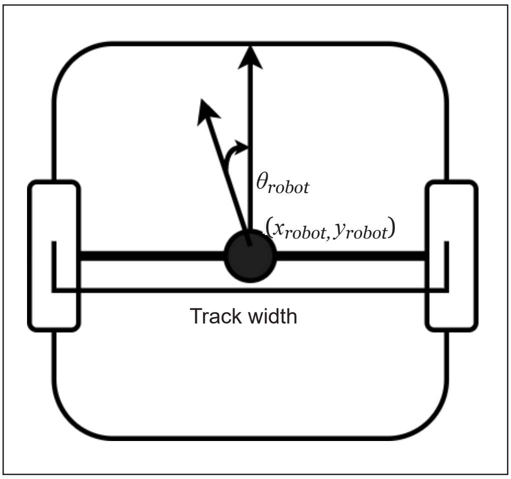

Figure 3. Differential drive robot kinematics

Simulation Experiment

Table 1 shows the Mobile Robot Hardware configurations. These parameters were input into the algorithm to simulate the behaviour of the differential drive robot.

Table 1
Mobile robot hardware configurations

Parameter	Value
Mobile robot radius	0.2 m
Track width	0.3 m
Minimum turning radius	0.15 m
Maximum angular velocity, *v*	1.82 rad/s
Maximum translational velocity, *v*	0.26 m/s
Range sensor angle range	0-360^o
Range sensor max range	1.5m

Test Environment. Figure 4 shows the occupancy map of the test environment. The selected environment layout is unstructured, with non-symmetrical occupied space and a narrow pathway.

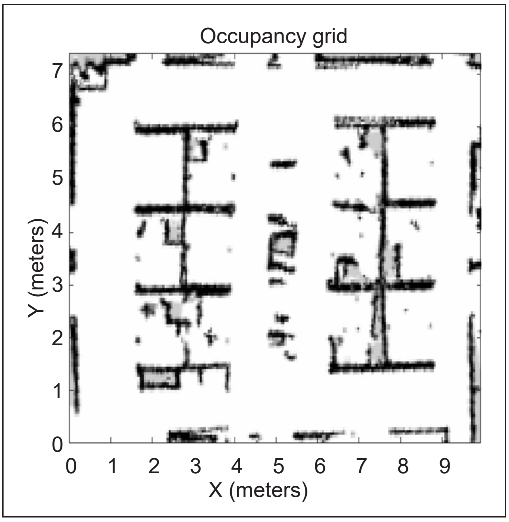

Figure 4. Map of test environment

Path Following. Table 2 shows pure pursuit algorithm parameters set constant in the experiment. The maximum angular velocity *ω* is capped at 1 rad/s, and a set of waypoints were defined. These points were set to pass through wide and narrow gaps on the map. It should be noted that the initial waypoints did not intersect with any of the occupied spaces. Table 3 shows the simulation parameters to be tested to observe the effect of LAD and translational velocity, *v*, on the trajectory of the mobile robot. Since the simulated hardware's maximum translational velocity, *v* is 0.26 m/s, the test parameters are capped at a translational velocity, *v*, of 0.2 m/s.

Table 2
Pure pursuit algorithms parameters

Parameter	Value
Maximum angular velocity, *ω*	1 rad/s
Initial waypoints (m)	[(1,6), (1,0.6), (4.5,0.6), (4.5,6), (5.8,6), (5.8,0.6), (9.2,0.60), (9.2,6)]

Table 3
Look Ahead Distance (LAD)

Translational Velocity, *v* (m/s)	LAD (m)
0.1	0.2
	0.4
	0.6
	0.8
	1.0
	2.0
0.2	0.2
	0.4
	0.6
	0.8
	1.0
	2.0

Obstacle Avoidance. Table 4 shows the selected parameters for the VFH+ algorithm. Safety Distance is the parameter to ensure safe navigation of the mobile robot. The obstacle-free space calculation considers the mobile robot's radius with an added Safety Distance value. Histogram Threshold was used to compute binary histograms from the polar obstacle density. Any values higher than the upper threshold are considered occupied (1), whereas values smaller than the lower threshold are considered free space (0). Values between the threshold limit are set to follow the previous binary histogram with the initial value of free space (0).

Table 4
VFH+ parameters

Parameter	Value
Safety Distance	0.1 m
Histogram Threshold	[3,10]
Distance Limits	[0.05 m, 1.5 m]
μ_1	5
μ_2	2
μ_3	2
α	360°

The third parameter, Distance Limit, was set to consider only meaningful readings from the range sensor. The lower limit prevents false positive readings, while obstacles beyond the upper limit are ignored to accelerate computation. The consequent parameters μ_1, μ_2, and μ_3 values were taken from Ulrich and Borenstein (1998, 2000), and the α value is based on the property of the range sensor.

Table 5
λ *parameters*

Parameter	Value
λ	0.5
	0.6
	0.7
	0.8
	0.9
	1.0

Heuristic Function. A heuristic function is proposed to ensure the robot returns to its designated waypoints after avoiding obstacles (Equation 12). The angular velocity ω will consider the magnitude of the calculated angular velocity from pure pursuit algorithm ω_{path} and the calculated angular velocity from VFH+. A weight parameter λ is introduced to prevent the mobile robot from steering too far from the waypoints, specifically waypoints with sharp turns. Table 5 shows the selection values of λ to be tested.

$$\omega = \begin{cases} \lambda * \omega_{pat\,h} - \omega_{VFH}\ , \ if\ \omega_{pat\,h} < 0 \\ \lambda * \omega_{pat\,h} + \omega_{VFH},\ if\ \omega_{pat\,h} > 0 \\ \omega_{VFH}, \qquad if\ \omega_{pat\,h} = 0 \end{cases} \qquad [12]$$

RESULTS AND DISCUSSION

Path Following

Figures 5 and 6 show plots of the trajectory of different LADs, where the translational velocity is fixed at 0.1 m/s and 0.2 m/s, respectively. In this simulation, mobile robot movement is based solely on waypoints, and the range sensor is disabled. From the plot, it could be observed that as the LAD value increases, the mobile robot will have a larger curvature. At LAD = 2.0 m, the curvature is too large that it collides with occupied space. While the larger LAD produces a smoother transition, the robot does not closely follow the designated waypoints. It could also be observed in Figure 7 that due to an increase in velocity, at LAD = 0.2 m, there is slight oscillation when the mobile robot turns at sharp corners. Based on these two plots, it could be concluded that the best LAD parameter for the mobile robot configuration would be at 0.2 m for the velocity of 0.1 m/s and 0.4 m for 0.2 m/s (refer to marked points in Figures 5 and 6). Of the two, the best overall path following trajectory parameters is at a translational velocity of 0.1m/s and LAD 0.2 m. It is due to its ability to follow the waypoints at a sharp corner closely.

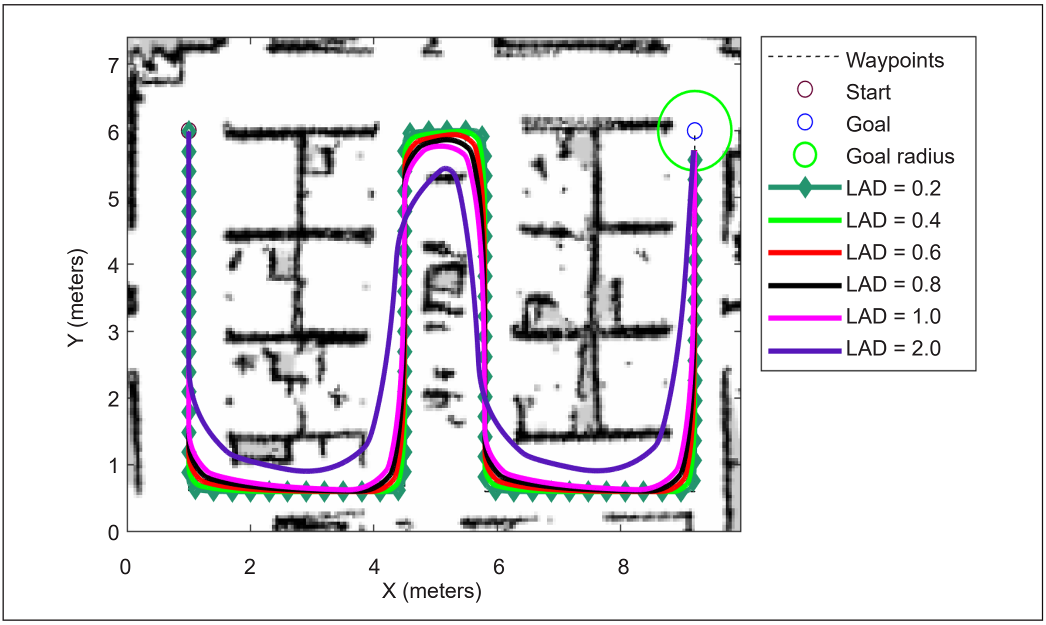

Figure 5. Plot of waypoints vs trajectory of different LAD (velocity = 0.1 m/s)

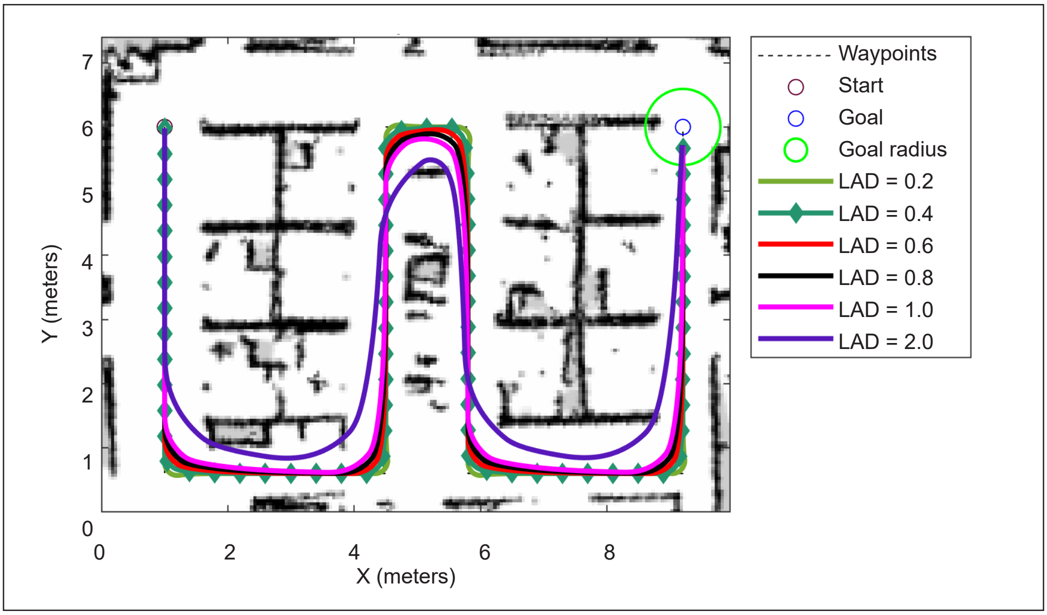

Figure 6. Plot of waypoints vs trajectory of different LAD (velocity = 0.2 m/s)

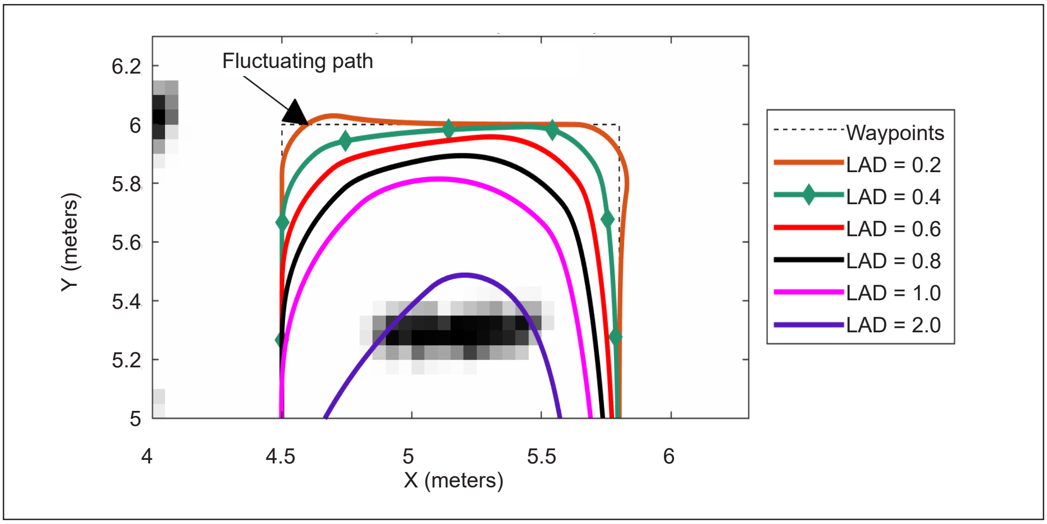

Figure 7. Plot of waypoints vs trajectory of different LAD [zoomed] (velocity =0.2 m/s)

Path Following with Obstacle Avoidance

Figure 8 shows an obstacle placed inside the map. It overlaps with the positioning of the waypoints. In this simulation, the range sensor is enabled. When the mobile robot senses an obstacle, the VFH+ algorithm will calculate the Masked Polar Histogram and compute an obstacle-free steering direction (Figure 9). It should be noted that the forward direction of the mobile robot is considered 0^o. Moreover, the positive angle is measured counterclockwise. In this case, the computed steering direction is 334^o.

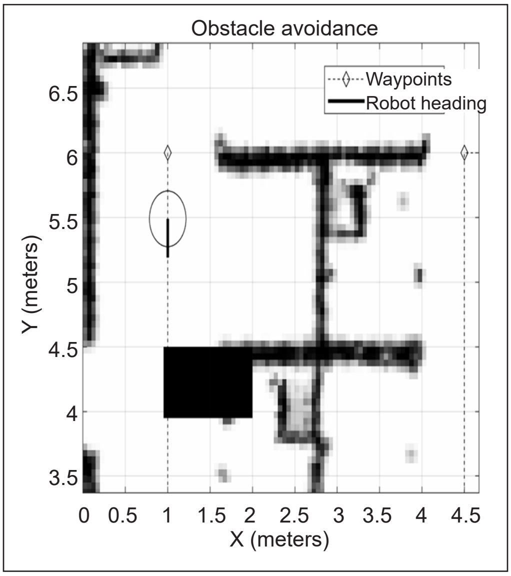

Figure 8. Mobile robot encountering obstacle

Figures 10 and 11 show the trajectory of the mobile robot using Pure-pursuit and VFH+. It could be seen that the λ parameter affects the mobile robot's ability to return to its designated waypoints after avoiding an obstacle. By implementing a low value of λ = 0.5, the mobile robot curved too far away from the waypoints and could not complete its navigation. At λ = 0.6, although initially, it was able to avoid the obstacle and return to the designated waypoints, it also failed to complete its navigation. Similar results were obtained by implementing high values of λ = 0.9 and λ = 1.0. In both cases, the mobile robot could not steer far from the obstacle, causing a collision. The mobile robot could complete the navigation at λ = 0.7 and λ = 0.8. Based on the simulation data, the best value of λ is 0.8, as the mobile robot managed to avoid the obstacle and safely return to its designated waypoints. The trajectory of λ = 0.8 is also relatively smoother compared to λ = 0.7.

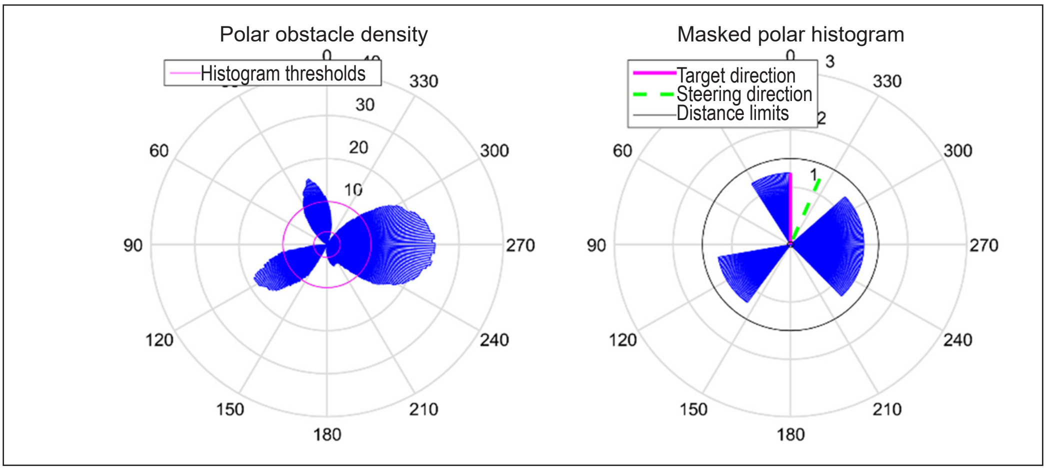

Figure 9. VFH+ histogram

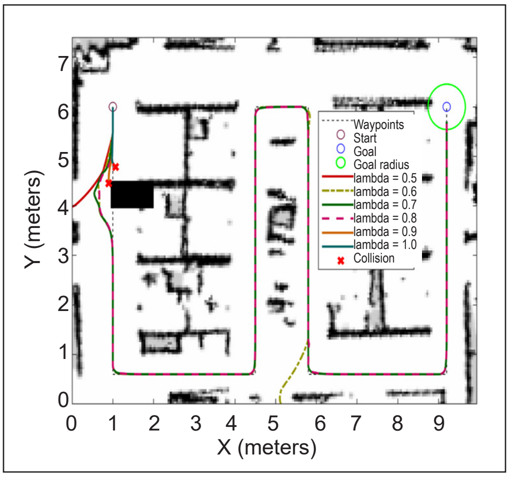

Figure 10. Plot of trajectory using Pure-pursuit and VFH+

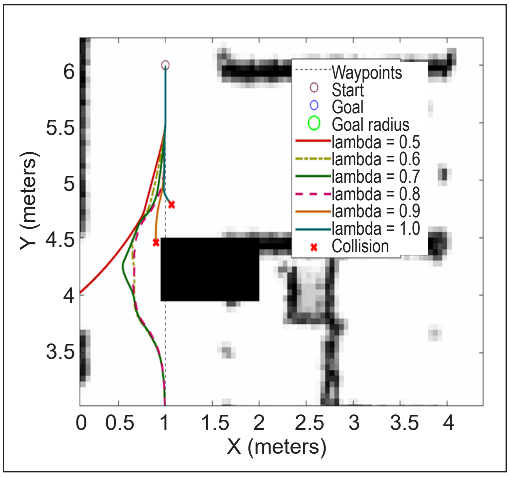

Figure 11. Plot of trajectory using Pure-pursuit and VFH+ (zoomed)

CONCLUSION

In conclusion, Pure-pursuit is an effective geometric path-following algorithm. The simulation shows that by tuning the value of LAD, the mobile robot can navigate closely to its waypoints inside an unstructured and confined space. By integrating the VFH+ algorithm as an obstacle avoidance method and introducing a weight parameter λ, the mobile robot can avoid an obstacle and return to its designated waypoints.

FUTURE WORKS

As for future works, the author plans to expand the work by applying the navigation concepts to industrial hardware and drones (Ibrahim et al., 2017). The work can also be significantly improved using metaheuristic optimisation, as proposed by Wang et al. (2020).

ACKNOWLEDGEMENT

This work was supported by Collaborative Research in Engineering, Science, and Technology (CREST), Malaysia, with grant no. 304/PELECT/6050423/C121.

REFERENCES

Ahn, J., Shin, S., Kim, M., & Park, J. (2021). Accurate path tracking by adjusting look-ahead point in pure pursuit method. *International Journal of Automotive Technology*, *22*(1), 119-129. https://doi.org/10.1007/s12239-021-0013-7

Bolbhat, S. S., Bhosale, A. S., Sakthivel, G., Saravanakumar, D., Sivakumar, R., & Lakshmipathi, J. (2020). Intelligent obstacle avoiding agv using vector field histogram and supervisory control. In *Journal of Physics: Conference Series* (Vol. 1716, No. 1, p. 012030). IOP Publishing. https://doi.org/10.1088/1742-6596/1716/1/012030

Chen, Y., Shan, Y., Chen, L., Huang, K., & Cao, D. (2018). Optimization of pure pursuit controller based on pid controller and low-pass filter. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 3294-3299). IEEE Publishing. https://doi.org/10.1109/ITSC.2018.8569416

Coulter, R. C. (1992). *Implementation of the Pure Pursuit Path Tracking Algorithm*. Defense Technical Information Center.

Díaz, D., & Marín, L. (2020). VFH+ D: An improvement on the VFH+ algorithm for dynamic obstacle avoidance and local planning. *IFAC-PapersOnLine*, *53*(2), 9590-9595. https://doi.org/10.1016/j.ifacol.2020.12.2450

Dong, X., Wang, M., Kang, C., Zhang, Y., Pan, W., & Gao, S. (2021). Improved auto-regulation VFH algorithm for obstacle avoidance of unmanned vehicles. In *2021 China Automation Congress (CAC)* (pp. 6467-6472). IEEE Publishing. https://doi.org/10.1109/CAC53003.2021.9727888

Girbés, V., Armesto, L., Tornero, J., & Solanes, J. E. (2011). Smooth kinematic controller vs. pure-pursuit for non-holonomic vehicles. In *Towards Autonomous Robotic Systems: 12th Annual Conference, TAROS 2011* (pp. 277-288). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-23232-9_25

Huang, Y., Tian, Z., Jiang, Q., & Xu, J. (2020). Path tracking based on improved pure pursuit model and pid. In *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)* (pp. 359-364). IEEE Publishing. https://doi.org/10.1109/ICCASIT50869.2020.9368694

Ibrahim, I. N., Izhevsk, K., Pavol, B., Aiman, A. A. M., Izhevsk, K., & Karam, A. (2017). Navigation control and stability investigation of a hexacopter equipped with an aerial manipulater. In *Proceedings of the 2017 21st International Conference on Process Control, PC 2017* (pp. 204-209). IEEE Publishing. https://doi.org/10.1109/PC.2017.7976214

Li, H., Luo, J., Yan, S., Zhu, M., Hu, Q., & Liu, Z. (2019). Research on parking control of bus based on improved pure pursuit algorithms. In *2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)* (pp. 21-26). IEEE Publishing. https://doi.org/10.1109/DCABES48411.2019.00013

Pappas, P., Chiou, M., Epsimos, G.-T., Nikolaou, G., & Stolkin, R. (2020). Vfh+ based shared control for remotely operated mobile robots. In *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (pp. 366-373). IEEE Publishing. https://doi.org/10.1109/SSRR50563.2020.9292585

Qinpeng, S., Zhonghua, W., Meng, L., Bin, L., Jin, C., & Jiaxiang, T. (2019). Path tracking control of wheeled mobile robot based on improved pure pursuit algorithm. In *2019 Chinese Automation Congress (CAC)* (pp. 4239-4244). IEEE Publishing. https://doi.org/10.1109/CAC48633.2019.8997258

Shan, Y., Yang, W., Chen, C., Zhou, J., Zheng, L., & Li, B. (2015). CF-Pursuit: A pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles. *International Journal of Advanced Robotic Systems*, *12*(9), Article 134. https://doi.org/10.5772/61391

Snider, J. M. (2009). *Automatic Steering Methods for Autonomous Automobile Path Tracking*. Carnegie Mellon University.

Ulrich, I., & Borenstein, J. (1998). VFH+: Reliable obstacle avoidance for fast mobile robots. *IEEE International Conference on Robotics and Automation, 2*, 1572-1577. https://doi.org/10.1109/ROBOT.1998.677362

Ulrich, I., & Borenstein, J. (2000). VFH*: Local obstacle avoidance with look-ahead verification. *Proceedings-IEEE International Conference on Robotics and Automation, 3*, 2505-2511. https://doi.org/10.1109/ROBOT.2000.846405

Wang, H., Chen, X., Chen, Y., Li, B., & Miao, Z. (2019). Trajectory tracking and speed control of cleaning vehicle based on improved pure pursuit algorithm. In *2019 Chinese Control Conference (CCC)* (pp. 4348-4353). IEEE Publishing. https://doi.org/10.23919/ChiCC.2019.8865255

Wang, R., Li, Y., Fan, J., Wang, T., & Chen, X. (2020). A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller. *IEEE Access, 8*, 166525-166540. https://doi.org/10.1109/ACCESS.2020.3023071

Wang, W. J., Hsu, T. M., & Wu, T. S. (2017). The improved pure pursuit algorithm for autonomous driving advanced system. In *2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)* (pp. 33-38). IEEE Publishing. https://doi.org/10.1109/IWCIA.2017.8203557

Yang, Y., Li, Y., Wen, X., Zhang, G., Ma, Q., Cheng, S., Qi, J., Xu, L., & Chen, L. (2022). An optimal goal point determination algorithm for automatic navigation of agricultural machinery: Improving the tracking accuracy of the pure pursuit algorithm. *Computers and Electronics in Agriculture, 194*, Article 106760. https://doi.org/10.1016/j.compag.2022.106760