Journal of Nonlinear and Convex Analysis
Volume 26, Number 7, 2025, 2335–2344

Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

THE DEVELOPMENT OF THE PURE PURSUIT WAYPOINT ALGORITHM WITH A FUNCTION OF THE DYNAMICAL OBSTACLE-AWARE PATH FOLLOWING

HEONJONG YOO AND SEONGGON CHOI*

Abstract. The paper focuses on the pure pursuit method and adjust linear and angular velocities design for the waypoint following platform. Firstly, the pure pursuit design is developed and it is incorporated into Robot Operating System(ROS) publish and subscribe block and adjust velocities design for 4 wheel mobile platform. In addition, the Ranger mini mobile platform can be moved through ROS connection. Lastly, the pure pursuit method is incorporated into the ROS publish and subscribe mechanism and adjusted velocities design. In this paper, the development of pure pursuit logic incorporating into the adjust velocities design is experimented based on the designated waypoint path with the Ranger-mini mobile platform with ROS connection.

1. Introduction

The way-point following algorithm have been studied for years in [6, 10]. In such papers, there are pure pursuit algorithm that is used for following designated way-point. In this presentation, we address a pure pursuit way-point following problem for an Ranger mini mobile platform. Our previous work used state-flow method [11] to develop a laptop-based experiment. The state flow method performs path tracking of the generated path and checks whether the Ranger-mini mobile platform accurately tracks the generated path dynamics by inspecting observed longitude and latitude, heading angle information in real time. Furthermore, the state flow algorithm is revised for way-point following briefly described in [4]. On the other side of the research, trajectory tracking method for 4 wheel mobile platform have been researched in several papers [5–8]. In recent years, Reinforcement Learning has been studied in various fields, particularly in path following problem in mobile platform, see in [2,7]. More recently, a 3D path-following control method is proposed using a linear active disturbance rejection control and Deep Deterministic Policy Gradient(DDPG) algorithm, explained in [12].

2020 *Mathematics Subject Classification.* 93C40.

Key words and phrases. Pure pursuit method, waypoint following logic, adjust linear and angular velocities.

This work was partly supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2025-RS-2020-II201462, 50 %) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. RS-2020-NR049604, 50 %).

∗Corresponding author.

Here, we use way-point following method rather than trajectory control, since the waypoint following is more accurate at the end point of arrival. To summarize that, We make the following contribution in this paper.

- Despite the various aforementioned control and estimation methods, the experiment was conducted focusing on the waypoint following logic incorporated into ROS publish and subscribe block introduced in MATLAB/SIMULINK recently.
- The real environment experiment is implemented based upon the way-points from ROS subscribe block.
- Furthermore, the dynamic obstacle avoidance experiment is implemented based upon the way-points from ROS subscribe block similar to the experiment in [1].

2. Pure pursuit algorithm design

In this section, the pure pursuit method described in [10] is used to incorporate the aforementioned design into adjust velocities part.

The angle of the mobile platform's body is set as α named as look ahead distance angle given as

$$\alpha = \theta - tan^{-1}\left(\frac{y_{look\ ahead} - y}{x_{look\ ahead} - x}\right). \tag{2.1}$$

The radius of curvature R is given as

$$R = \frac{l_d}{2\sin(\alpha)} \tag{2.2}$$

where l_d is look ahead distance.

$$a_i(t) = 2\alpha_i(t)R_i(t) \tag{2.3}$$

where $\alpha_i(t)$ is the look-ahead distance angle at i-th time point, and $R_i(t)$ is the radius of curvature at the i-th time point. The linear velocities are given as

$$v_i(t) = \frac{a_i(t)}{\delta T} = \frac{\alpha_i(t)l_d}{\delta T\sin(\alpha_i(t))}. \tag{2.4}$$

Fig. 2 is set as the look-ahead point at the i-th time point. This look-ahead point is also the reference path point at the n-th time point tn, then the time interval δT is shown in

$$\delta T = t_n - t_i. \tag{2.5}$$

The pure pursuit geometric model is described in Fig. 1. where L is wheel base of the mobile platform, R is the radius of the curvature. The ranger-mini mobile platform's steering angle ϕ at each sampling time is given as

$$\phi = \frac{L}{R}. \tag{2.6}$$

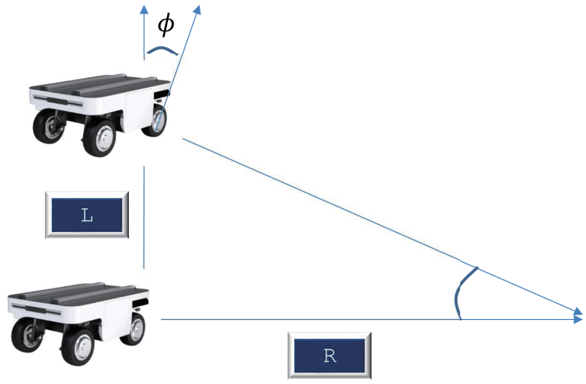

FIGURE 1. The pure pursuit geometric model

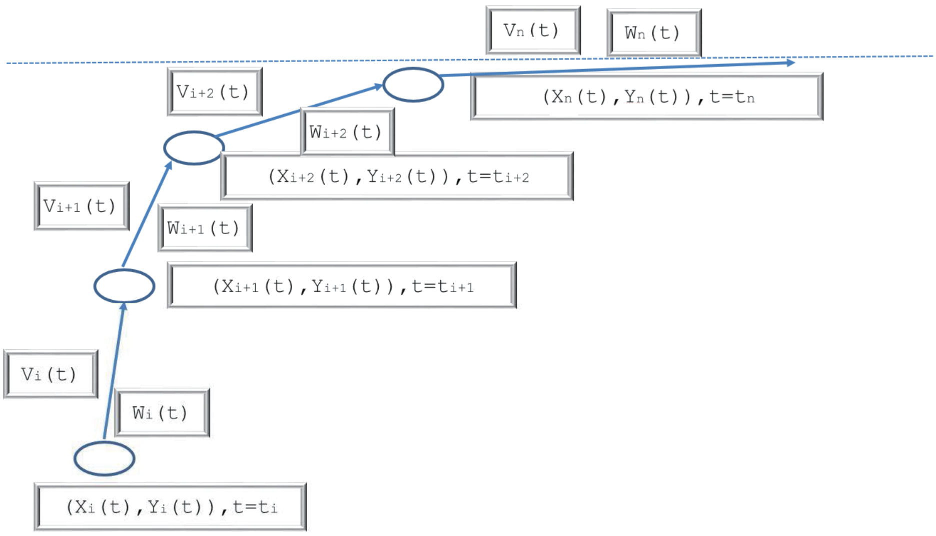

FIGURE 2. Tracking parameters of the pure pursuit with adjusted velocities

3. Adjust velocities part

A scan data set enters this block, compares the range values, and generates linear and angular velocities that can avoid dynamic obstacles. The adjusted linear and angular velocities design in the Simulink is provided in Fig. 3.

The recovery subsystem design is described in Fig. 4.

The path following subsystem design is described in Fig. 5.

The following code is written in the Matlab function box, given as

```
Algorithm I
function w = exampleHelperComputeAngularVelocity
(steeringDir, wMax)
```

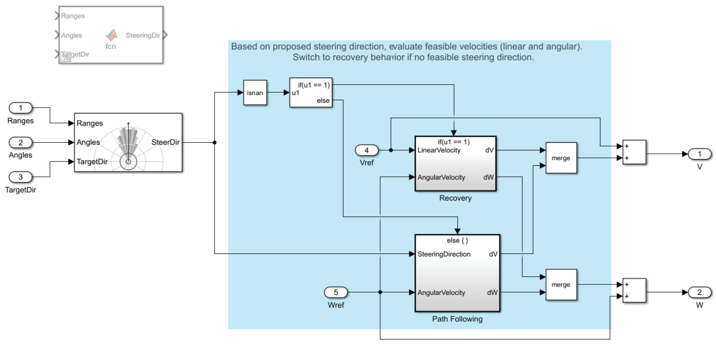

FIGURE 3. The design of the adjusted linear and angular velocities.

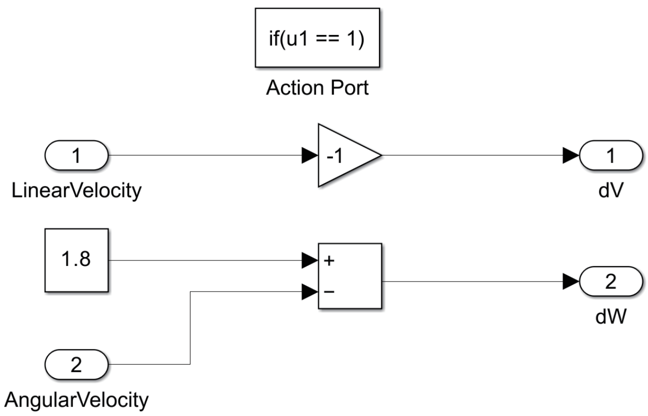

FIGURE 4. The subsystem design of the recovery part

```
if nargin == 1
    wMax = inf;
end

validateattributes(steeringDir, {'double'},{'real'},...
    'exampleHelperComputeAngularVelocity', 'STEERINGDIR', 1);
validateattributes(wMax, {'double'},{'real', 'positive'},...
    'exampleHelperComputeAngularVelocity', 'WMAX', 2);

curPose = [0 0 0];
```

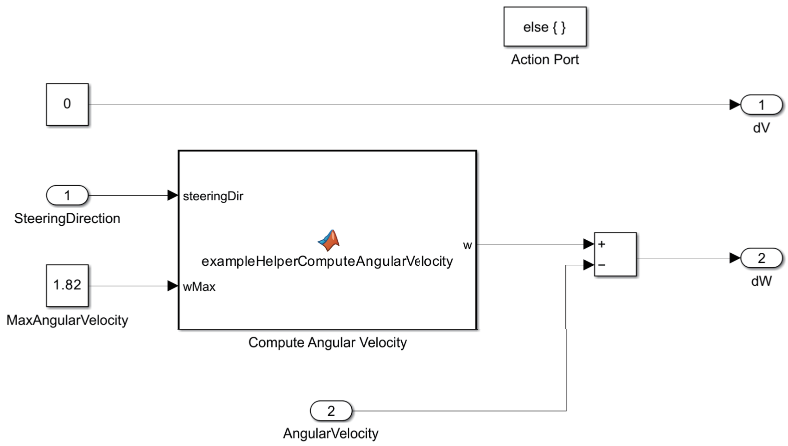

FIGURE 5. The subsystem design of the path following part

```
lookaheadPoint = [cos(steeringDir), sin(steeringDir)];
slope = atan2((lookaheadPoint(2) - curPose(2)), ...
    (lookaheadPoint(1) - curPose(1)));
alpha = angdiff(curPose(3), slope);

w = (2*sin(alpha));

if abs(abs(alpha) - pi) < 1e-12
    w = sign(w)*1;
end

if abs(w) > wMax
    w = sign(w)*wMax;
end
end
```

The adjusted velocities part is incorporated into the previous pure pursuit design. Hence the dynamical obstacle avoidance is implemented during the experiment, since the angular velocities is changed when the Ranger-mini platform encountered unexpected obstacle, by displaying range value in the path following subsystem in Fig. 5.

4. Experiment through Ranger mini mobile platform

The sensor and hardware configuration is given as in Fig. 6. We conduct experiment in an range mini mobile platform introduced in [11] To deal with the dynamic obstacle of a generated path, we use ROS scan data sets that is sent to MATLAB

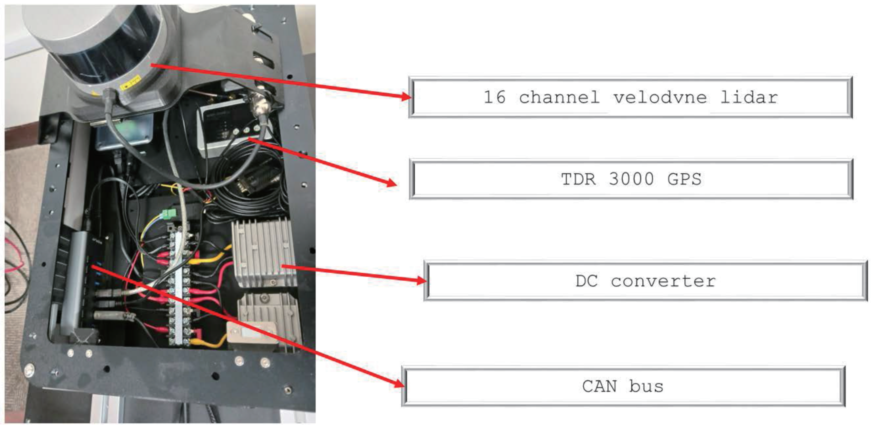

FIGURE 6. Picture view of the hardware test bench of the Ranger-mini

laptop in a real time. [4] discussed that mobile robots can sense and navigate their environments using various sensors, here, we use 16 channel Velodyne Lidar(VLP 16 Puck model). The sensors structure is given as Fig. 6. In addition, we use software Adjust incorporated pure pursuit control algorithms to decide, plot generated way-point paths, avoid obstacles. Furthermore, most mobile robots can learn from their surroundings and enhance their performance over time owing to adjust linear and angular velocities algorithm. The overall structure in MATLAB SIMULINK is shown in Fig. 7.

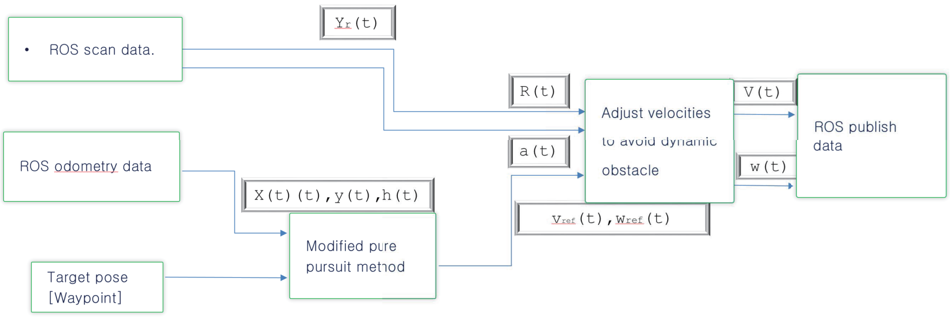

FIGURE 7. The overall structure using modified pure pursuit and adjusted velocities to avoid dynamic obstacle

4.1. **Calculate the initial heading angle from ROS Odometry Block.** Firstly, we moved Ranger-mini platform from starting point to end point using ROS publish and subscribe block. The starting point and end point from ROS odometry block is given as

$$x_1 = 0, x_{end} = 0.85, y_1 = 0, y_{end} = 0.84. \tag{4.1}$$

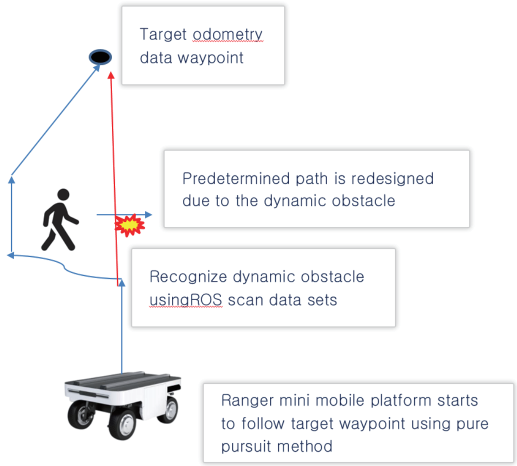

FIGURE 8. The dynamic obstacle avoidance and way-point following experiment scenario

TABLE 1. Experimental Condition

Initial $(x(1), y(1))$	End point $(x(end), y(end))$	Initial heading angle $h(0)$
(0,0)	(0.85,0.84)	0.7854 (rad)

The heading angle calculation is given as

$$\begin{aligned} \delta x &= x(end) - x(1), \\ \delta y &= y(end) - y(1), \\ h(0) &= atan(\delta y, \delta x). \end{aligned} \tag{4.2}$$

Where $h(0)$ is initial heading angle, x, y are longitude and latitude values from ROS odometry information.

The following table are shown in experimental condition. In addition to the condition in Table 1, the sampling time is set as 0.01 and mobile platform's linear velocities are set as 0.1 (m/s). The dynamic obstacle which means a human was interrupted during the experiment, then the Ranger mini platform stops and tracks another way to avoid a human. The result of dynamic obstacle avoidance scenario is given in Fig. 8

The 16 channel velodyne lidar recognizes the dynamic obstacle denoted as red circle, then adjusted velocity part recalculated linear and angular velocities to avoid obstacle. If we use only pure pursuit, the green line implies that it could not avoid dynamic obstacle. The result explains that the adjust velocities incorporated into pure pursuit well tracks the final point, simultaneously, the aforementioned method could avoid dynamic obstacle bypassing the direction, described in Fig. 9. In the existing design method using the only pure pursuit, the ranger-mini mobile platform could not avoid the dynamical obstacle described in green line in Fig. 9.

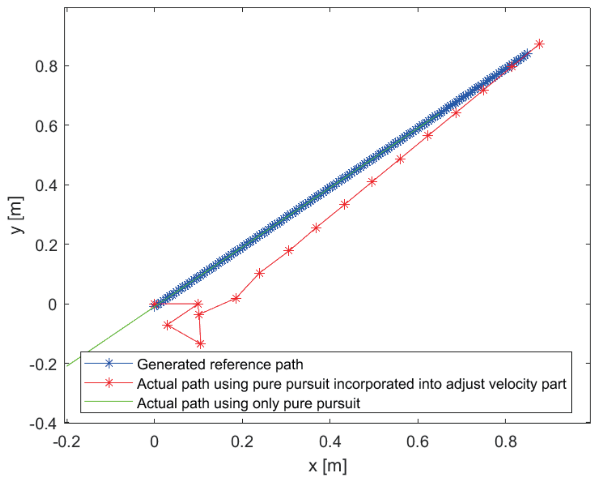

FIGURE 9. The experiment result avoiding dynamic obstacle, simultaneously following starting point to end point of ROS odometry

By adjusting the angular velocity in the adjusted part introduced in Section 3, the mobile platform could be avoided before hitting dynamic obstacles. In the real environment, the plot in Fig. 9 is replotted in MATLAB web map, described in Fig. 10.

FIGURE 10. The experiment result avoiding dynamic obstacle, simultaneously following starting point to end point of ROS odometry in MATLAB web map

5. CONCLUSION

In this paper, the pure pursuit based adjust control was modeled to a real 4 wheel independent steering system for the waypoint following of an the generated designated waypoint. In addition to that, the dynamic obstacle avoidance experiment was conducted by designing adjust control combining with ROS scan data sets. The experimental result is shown in which the mobile robot follows designated waypoints based on ROS odometry data sets. Simultaneously, the 4 wheel mobile platform (Ranger mini) can avoid dynamic obstacle (In the experiment, a person was interrupted during experiment time).

References

[1] K. Almazrouei, I. Kamel and T. Rabie, *Dynamic obstacle avoidance and path planning through reinforcement learning*, Applied Science **13** (2023: 8174.

[2] Y. Cao, K. Ni, X. Jiang, T. Kuroiwa, H. Zhang, T. Kawaguchi, S. Hashimoto and W. Jiang, *Path following for autonomous ground vehicle using DDPG algorithm: A reinforcement learning Approach*, Applied Science **13** (2023): 6847.

[3] J. Chen, Z. Shuai, H. Zhang and W. Zhao, *Path following control of autonomous four-wheel-independent-drive electric vehicles via second-order sliding mode and nonlinear disturbance observer techniques*, IEEE Transactions on Industrial Electronics **68** (2021), 2460–2469.

[4] M. H. Hong, H. J. Yoo and S. G. Choi, *Incorporation of waypoint following logic into ROS publish and subscribe mechanism*, in: Proceedings of 2024 26th International Conference on Advanced Communications Technology (ICACT), IEEE, 2024, pp. 373–375.

[5] Y. Huang, H. Ding and Y. Zhang, *A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach*, IEEE Transactions on Industrial Electronics **67** (2020), 1376–1386.

[6] Q. Lin, S. Mitsch, A. Platzer and J.M. Dolan, *Safe and Resilient Practical Waypoint-Following for Autonomous Vehicles*, IEEE Control Systems Letters, 6(2021), 1574-1579.

[7] S. S. Mousavi, M. Schukat and E. Howley, *A deep reinforcement learning: An overview*, in: Proceedings of SAI Intelligent Systems Conference (IntelliSys), IEEE, 2017, pp. 426–440.

[8] Y. Tong, H. Jing, B. Kuang, G. Wang, F. Liu and Z. Yang, *Trajectory tracking control for four-wheel independently driven electric vehicle based on model predictive control and sliding model control*, in: Proceedings of 5th CAA International Conference on Vehicular Control and Intelligence (CVCI), IEEE, 2021, pp. 1–8.

[9] J. Wang, T. Yan, Y. Bai, Z. Luo, X. Li and B. Yang, *Assistance quality analysis and robust control of electric vehicle with differential drive assisted steering system*, IEEE Access **8** (2020), 136327–136339.

[10] R. Wang, Y. Li, J. H. Fan, T. Wang and X. Chen, *A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and Velocity Controller*, IEEE Access **8** (2020), 166525–166540.

[11] H. J. Yoo, D. Baek and S. G. Choi, *Navigation path following platform for a greenhouse shuttle robot using the state-flow method*, Mathematical Problems in Engineering, **2024** (2024): Article ID 8810990.

[12] Y. Zheng , J. Tao, Q. Sun, X. Zheng, H. Sun, M. Sun and Z. Chen, *DDPG-based active disturbance rejection 3D path-following control for powered parafoil under wind disturbances*, Nonlinear Dynamics **111** (2023), 11205–11221.

Manuscript received September 10, 2024
revised December 3, 2024

Heonjong Yoo
Research Institute for Computer and Information Communication(RICIC), Chungbuk National University, Cheongju-city, Republic of Korea
E-mail address: 622061@chungbuk.ac.kr

Seonggon Choi
Information and Communication Engineering,Chungbuk National University, Cheongju-city, Republic of Korea
E-mail address: sgchoi@chungbuk.ac.kr