A GPS-Free Human-Following Framework for Caddy Robots Based on Safety Corridor Obstacle Avoidance Strategy

Jongbin Kim* Non-member, Bumyeon Lee** Non-member, Junghyun Choi*a) Non-member

Human-following functionality of autonomous mobile robots is a core technology for service robots to collaborate with humans in real environments. However, existing studies have limitations such as expensive sensor dependency, high computational complexity, and GPS-based position estimation. These limitations have restricted the application of low-cost embedded platforms in outdoor environments where GPS shadowing occurs. This paper proposes a human-following system that operates in real-time on a low-cost MCU using only a single LiDAR and IMU in outdoor environments where GPS is unavailable. The proposed system adopts a distributed control architecture that separates the perception unit and the drive unit. Follower recognition is performed through a lightweight distance-based clustering algorithm. The following control is designed with a dual-mode structure: Direct Following mode, which directly follows the follower in environments without obstacles, and Avoidance Following mode, which follows while avoiding obstacles. In particular, the Safety Corridor concept was introduced to implement real-time obstacle avoidance without complex path planning. This paper demonstrates stable following performance while maintaining a safe distance not only in straight following in open areas but also in corner sections and environments with obstacles. Through this, it aims to solve the high cost and high complexity problems of existing systems and apply to the commercialization of golf caddy robots.

Keywords: Autonomous following robot, Human following, Obstacle avoidance, Embedded system, Caddy robot, Low-cost sensor

1. Introduction

Human-robot interaction has become a core component of service robot technology. In particular, following robots that assist humans have attracted significant attention for their potential applications in various service sectors, including worker assistance in logistics warehouses, patient accompaniment in hospitals, and baggage transportation in airports.

Robot technology is also being actively applied in the field of golf. A caddy carries golf bags and clubs while providing advice on course information and club selection. However, the high cost of hiring professional caddies makes it difficult for all golfers to afford. Therefore, autonomous caddy robots have been developed to reduce costs and popularize golf. However, most remain at the experimental level, and even commercialized products lack sufficient technical disclosure.

Research on human-following for autonomous mobile robots is classified into vision-based and range sensor-based approaches according to the sensing method.

Vision-based approaches utilize visual information acquired through cameras. Studies include gait recognition that identifies specific individuals by analyzing the periodicity of walking patterns [1], tracking methods that learn appearance features such as color and shape through probabilistic models [2, 3], depth estimation using binocular disparity from stereo cameras[4], and deep learning-based trajectory prediction [5]. However, these methods suffer from performance degradation under lighting changes and backlight conditions, and are vulnerable to occlusion phenomena. Additionally, they share a common limitation that real-time implementation on low-cost embedded systems is difficult due to high computational loads.

In range sensor-based research, the accurate distance measurement capability of LiDAR is utilized. Research implementing following in crowded environments by detecting human legs with 2D LiDAR and applying particle filters has been reported [6]. However, real-time implementation on low-specification MCUs is difficult due to complex probabilistic computations. Distributed sensor network-based position estimation has also been proposed [7]. However, prior infrastructure construction is essential, reducing adaptability to new environments and making outdoor application impossible.

Multi-sensor fusion systems have also been developed for practical applications. Representative examples include golf caddy robots fusing LiDAR, camera, and GPS [8], multi-sensor integration in omnidirectional robots [9], and RGB-D based Map-Free following [10]. However, synchronization and calibration of multiple sensors is complex, and position estimation accuracy degrades in GPS-shadowed areas. Additionally, RGB-D sensors have limitations for outdoor application due to sunlight interference. Special application research such as front-following Push-Cart [11] and field operation of shopping guide robots [12] has also been conducted, but has limitations of high computational load from complex trajectory prediction and indoor environment specialization.

a) Correspondence to: jh.choi@kmu.ac.kr
* Department of Automotive Engineering, Keimyung University, Daegu 42601, Korea
** Department of Mechanical Engineering, Keimyung University, Daegu 42601, Korea

In summary, existing studies have the following common limitations. First, high dependence on expensive sensors such as 3D LiDAR and stereo cameras makes commercialization difficult [13]. Second, deep learning-based algorithms have high computational loads, making real-time processing on low-specification MCUs difficult [3, 11]. Third, existing obstacle avoidance algorithms such as RRT* have high complexity, limiting implementation in embedded environments [3, 5]. These limitations restrict application of low-cost embedded platforms in GPS-shadowed areas [13, 14].

This study aims to solve the high cost, high complexity, and GPS dependency problems of existing systems. A low-cost sensor configuration using only a single 2D LiDAR and IMU is adopted in unpaved outdoor environments where GPS is unavailable. A distance-based clustering algorithm with reduced complexity was developed to enable real-time processing on a low-cost MCU (STM32F7). For obstacle avoidance, the Safety Corridor concept was introduced to implement effective avoidance actions without complex path planning.

The contributions of this paper are as follows. First, a lightweight clustering algorithm enabling real-time following with only a single 2D LiDAR is proposed. Second, a following control strategy adapting to various environments through a dual-mode structure of Direct Following and Avoidance Following is developed. Third, a low-computation obstacle avoidance algorithm is implemented by introducing the Safety Corridor concept.

The remainder of this paper is organized as follows. Section 2 introduces the hardware structure of the caddy robot and the follower recognition algorithm. Section 3 describes the control algorithm of Direct Following and Avoidance Following. Section 4 verifies the performance of the proposed algorithm through scenario-based experiments. Finally, Section 5 concludes the paper.

2. Overall Structure of Caddy Robot

This chapter describes the hardware structure of the proposed autonomous following caddy robot and the overall structure of the driving controller.

Fig. 1 shows the overall structure of the caddy robot developed in this study. The robot adopts a differential drive system, with two driving motors mounted on the rear wheels enabling independent speed control. The two front wheels are designed as casters to smoothly support the robot's directional changes. The front wheels are equipped with suspension to absorb impacts on unpaved terrain and tilting joints to prevent sudden rolling. This enables stable driving even in lawn environments such as golf courses.

The handle is equipped with a button for switching between manual mode and autonomous mode, and an electromagnetic-based tactile sensor is embedded so that the system automatically switches to manual mode when the user grasps the handle. This design ensures safety by allowing the user to directly control the robot at any time.

The main controller incorporates an STM32F7 microcontroller and IMU. It processes scan data received from the 2D LiDAR to recognize the follower and transmits appropriate velocity commands to the driving motors of each wheel according to the situation. The battery is positioned at the center of the robot to lower the center of gravity and improve

Fig. 1 Structures of Caddy Robots

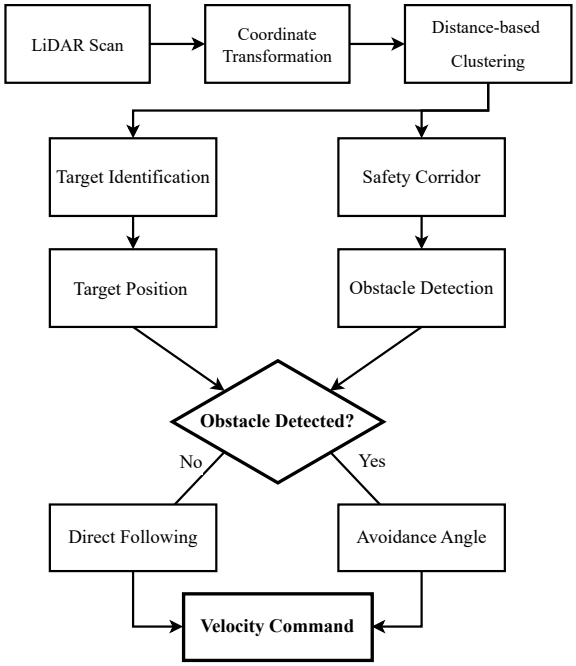

Fig. 2 Follow Diagram

driving stability.

As shown in the block diagram presented in Fig. 2, scan data acquired from the LiDAR undergoes a coordinate transformation process to be converted into a Cartesian coordinate system. The transformed point cloud is classified into individual objects through a distance-based clustering algorithm. Among these, objects satisfying predefined conditions are selected as following targets. Simultaneously, the Safety Corridor algorithm detects obstacles existing on the path between the robot and the follower. Based on the obstacle detection results, the system selects the appropriate mode between Direct Following and Avoidance Following to generate the final velocity command.

This dual-mode structure enables both efficient following in open areas and safe driving in complex environments. The detailed operating principles of each mode are explained in subsequent chapters.

2.1 Follower Recognition Algorithm

In this system, only data from the 180° region in front of the robot is used, and the effective distance range is set from 0.1 m to 10.0 m. The raw data received in polar coordinates is transformed into Cartesian coordinates based on the robot coordinate system through equations (1) and (2).

$$x = r \times \cos\left(\theta - \frac{\pi}{2}\right) \tag{1}$$

$$y = r \times \sin\left(\theta - \frac{\pi}{2}\right) \tag{2}$$

Here, the offset of $\pi/2$ radians is for compensating the installation orientation of the LiDAR sensor, aligning the coordinate system so that the robot's forward direction coincides with the positive x-axis direction.

The coordinate-transformed points are grouped into individual objects through a distance-based clustering algorithm. In this study, a simple yet effective method is applied where adjacent points within a distance of 0.1 m are classified as the same object. For each cluster, the minimum/maximum coordinate values and the number of points are calculated to determine the size and position of the object.

The following conditions are set for follower recognition. First, the diagonal length of the cluster must be between 0.15 m and 0.8 m. This value is set based on the leg width of an adult, filtering out objects that are too small such as noise or too large such as walls and furniture. Second, during initial follower recognition, only objects located within 1.0 m directly in front of the robot and within ±0.3 m to the left and right are considered as candidates. This allows the user to intentionally stand in front of the robot to initiate following.

Once an object is recognized as a follower, it is continuously tracked in subsequent frames. The position is updated only when the distance between the follower position from the previous frame and the object detected in the current frame is within 0.8 m, determining it as the same follower. This continuity condition prevents the follower from suddenly switching to another object. Additionally, it enables stable following even in situations with temporary LiDAR noise or occlusion.

3. Human Following Control Algorithm

This chapter describes the control algorithm for the Direct Following mode, in which the robot directly follows the follower based on their position in an environment without obstacles. This mode is applied in situations where there are no obstacles on the path, such as open areas or wide corridors. It determines the forward velocity and rotational velocity based on the distance and direction to the follower.

3.1 Direct Following

This section presents a distance maintenance control algorithm based on the follower's position information. The proposed controller converges the relative distance between the follower and the robot to the target distance while preventing abrupt velocity changes to achieve stable following behavior.

The follower's position (x, y) obtained from the follower recognition module is converted to polar coordinates to calculate the relative distance r and direction angle θ as shown in equations (3) and (4).

$$r = \sqrt{x^2 + y^2} \tag{3}$$

$$\theta = \text{atan2}(y, x) \tag{4}$$

The forward velocity command v_{cmd} is generated proportionally to the error between the current distance r and the target safe distance d_{safe} as shown in equation (5).

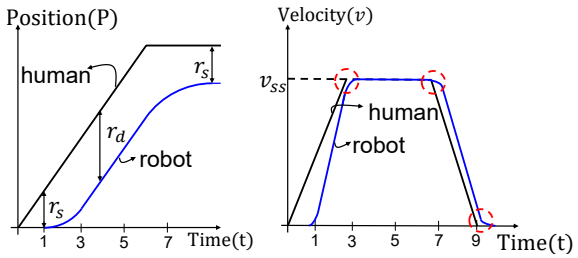

Fig. 3 Position and velocity graph based on follower distance

$$v_{cmd} = K_p \cdot (r - d_{safe}) \tag{5}$$

Here, K_p is the proportional gain, which is set to $K_p = 2.0$ in this system, and $d_{safe} = 0.85$ m is set to maintain an appropriate distance from the follower.

Abrupt velocity changes can compromise the driving stability of the robot and cause discomfort to passengers. To prevent this, a Rate Limit Filter is applied to the velocity command to limit acceleration as shown in equation (6).

$$v_{fwd}(t) = v_{fwd}(t-1) + \text{clip}(v_{cmd} - v_{fwd}(t-1), -\Delta v_{max}, \Delta v_{max}) \tag{6}$$

Here, $\Delta v_{max} = 0.005$ m/s^2 is set to implement smooth acceleration and deceleration. Fig. 3 shows the following behavior of the proposed controller. In the Position graph, the robot starts following with a delay relative to the follower's movement and converges to the target distance with an S-shaped smooth trajectory. In the Velocity graph, due to the effect of the Rate Limit Filter, the robot's velocity does not change abruptly but gradually reaches the steady-state velocity v_{ss}, and when the follower stops, it also decelerates smoothly.

Finally, the left and right wheel velocities are calculated using the forward velocity v_{fwd} and the angular velocity ω generated from the direction angle θ as shown in equations (7) and (8).

$$v_L = v_{fwd} + \omega \tag{7}$$

$$v_R = v_{fwd} - \omega \tag{8}$$

Here, $\omega = K_\omega \cdot \theta$, and K_ω is the angular velocity gain. Through this, the robot performs following motion that adjusts its direction toward the follower while maintaining the target distance.

Various emergency stop conditions are set for the safe operation of the autonomous following robot. These conditions are designed considering the physical structure of the robot and the limitations of sensors. In case of dangerous situations, safety is ensured through immediate stopping and transition to manual mode.

The first emergency stop condition is follower recognition failure. If the follower is not detected for 3 consecutive frames or more, following is stopped and the robot halts. This is to prepare for situations where the follower suddenly disappears from view or LiDAR sensor occlusion occurs.

The second condition is follower distance deviation. If the

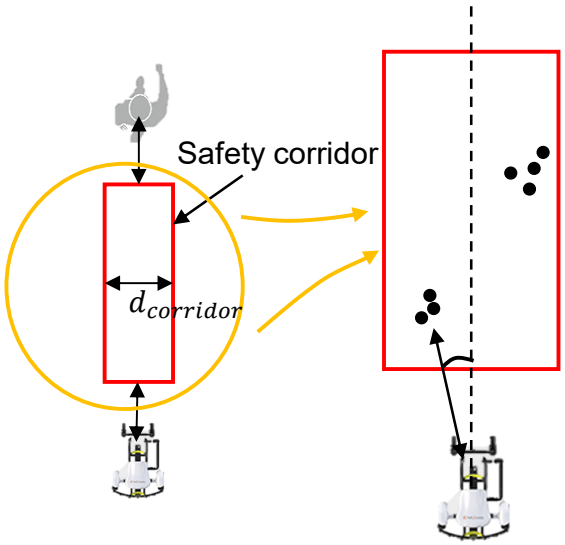

Fig. 4 Obstacle detection in the Safety Corridor region

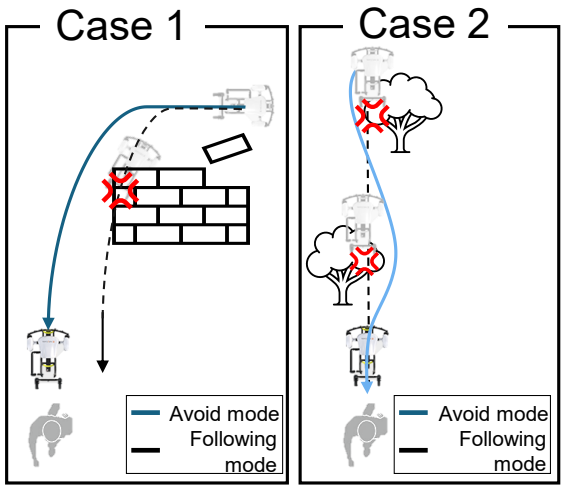

Fig. 5 Avoidance driving by scenario

state where the distance to the follower exceeds 2.5 m continues for 10 frames or more, it is determined that the follower has been lost and the robot stops. This applies to cases where the follower moves faster than the robot's maximum following speed or makes a sudden direction change.

The third condition is obstacle detection near the follower. If another object of a certain size (0.2 m) or larger is continuously detected for 6 frames or more within 0.6 m around the object recognized as the follower, it is determined that the follower and obstacle are too close or an error has occurred in follower recognition, and the robot stops.

The fourth condition is LiDAR communication abnormality. If LiDAR data is not received for 400 ms or more, it is determined as a sensor abnormality and the robot stops immediately. These multiple safety conditions work complementarily to ensure safe stopping of the robot in various exceptional situations.

3.2 Obstacle Avoidance Control Based on User Position

The Direct Following mode operates effectively in open areas or straight sections, but has limitations in environments such as corner sections with walls or narrow corridors. When the follower passes near an obstacle, the robot attempting to follow in a straight path risks colliding with that obstacle. To solve this problem, this study developed an avoidance path following algorithm introducing the Safety Corridor concept.

The core design principles of the proposed avoidance algorithm are computational load minimization and real-time processing. Instead of complex path planning algorithms, a virtual safety corridor was set between the robot and the follower. By considering only obstacles within this corridor as avoidance targets, the computational burden was greatly reduced. Additionally, a strategy of immediately returning toward the follower after minimal avoidance action was adopted to minimize degradation of following performance.

The Safety Corridor is a virtual corridor-shaped region connecting the robot and the follower, and only obstacles existing within this region are considered as avoidance targets. For the generation of the Corridor, the coordinate system is first rotation-transformed toward the follower direction as shown in equations (9), (10), and (11).

$$\theta_{target} = \text{atan2}(y_{follower}, x_{follower}) \tag{9}$$

$$x' = x \times \cos(-\theta_{target}) + y \times \sin(-\theta_{target}) \tag{10}$$

$$y' = x \times \sin(-\theta_{target}) + y \times \cos(-\theta_{target}) \tag{11}$$

In the rotated coordinate system, the Corridor is defined as follows. In the x-axis direction, it is the section from the robot position 0 to the follower position. The total width of the Corridor in the y-axis direction is set to 1.5 m. This is the physical width of the robot (approximately 0.6 m) plus appropriate clearance space. Fig. 4 helps understanding with the corridor image.

For all points detected by LiDAR, the same coordinate transformation is applied and then it is determined whether the rotated coordinates are included in the Corridor region. Points located inside the Corridor while excluding the follower region (0.6 m around the follower) are classified as obstacle candidates. These candidate points are again grouped into individual obstacle objects through distance-based clustering with a reference distance of 0.1 m. Only clusters with diagonal length of 0.2 m or more are recognized as actual obstacles.

The sensitivity of obstacle detection is adjusted through a consecutive frame counter. To prevent false detection due to single frame noise, the avoidance mode is activated only when obstacles are detected for 3 consecutive frames in the same area. Conversely, the avoidance state is maintained for 3 frames even after the obstacle disappears to ensure stable mode transition.

When an obstacle is detected within the Safety Corridor, the system transitions to Avoidance Following mode. The basic principle of avoidance path generation is to offset a certain angle in the opposite direction of the obstacle from the reference direction toward the follower.

The avoidance direction is determined according to the position of the obstacle. If the average y-coordinate of the obstacle cluster within the Corridor is positive, the obstacle is on the left side of the following path, so it avoids to the right.

If negative, it avoids to the left. The avoidance angle is set to a default value of 30°. This was determined through experiments as the optimal value for effectively bypassing obstacles while not losing the follower, as shown in equation (12).

$$\theta_{avoid} = \theta_{target} \pm 30^{\circ} \tag{12}$$

The forward velocity in avoidance mode is determined proportionally to the distance to the follower, the same as in Direct Following mode, but the rotation command is generated based on the modified avoidance angle. This allows the robot to maintain a constant distance from the follower while moving along a curved path that bypasses the obstacle.

When the avoidance action is completed and obstacles are no longer detected within the Corridor, the system automatically returns to Direct Following mode. To prevent sudden direction changes during mode transition, the avoidance angle decreases gradually, thereby generating a smooth trajectory.

4. Scenario-Based Avoidance Following Verification

Experiments were conducted in two representative scenarios presented in Fig. 5 to verify the performance of the avoidance path following algorithm.

Case 1: Corner Corridor Avoidance Following

The first scenario is a situation where the follower passes by a wall corner while turning the corner. The follower passed through a 90° corner while grazing the obstacle. If the robot follows the follower in a straight line, it would collide with the wall.

As a result of the experiment, as shown in the graph of Fig. 6, at approximately 1 m before the follower entered the corner, the wall was detected within the Safety Corridor and the avoidance mode was activated. The robot moved along a path offset by approximately 30° from the follower direction, securing a safe distance of approximately 0.5 m from the wall while passing the corner. After passing the corner, as the obstacle within the Corridor disappeared, it automatically returned to Direct Following mode. Throughout the entire process, the distance to the follower was maintained within the range of 0.8 m to 1.3 m.

Case 2: S-Curve Avoidance Following

The second scenario is a situation where the follower goes straight but there is an obstacle (tree) on the path that is difficult for the robot to pass through. The follower went straight at a position approximately 1.0 m away from the obstacle. However, considering the robot's width, there is a risk of collision with the obstacle on the same path.

In this scenario, the robot generated an S-shaped avoidance trajectory as shown in the graph of Fig. 7. When the obstacle is detected within the Corridor, the avoidance mode is activated to avoid in the opposite direction of the obstacle. After passing the obstacle, it returns in the opposite direction to approach the follower. As a result of the experiment, the robot successfully avoided while maintaining a minimum distance of approximately 0.4 m from the obstacle, and returned to normal following state after completing the avoidance action.

The Angle graph in Fig. 8 shows the change in the robot's target direction angle over time. In Direct Following mode,

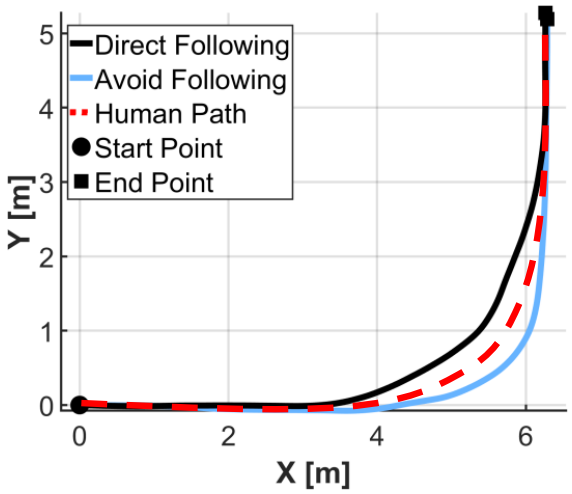

Fig. 6 Case 1: Corner corridor avoidance

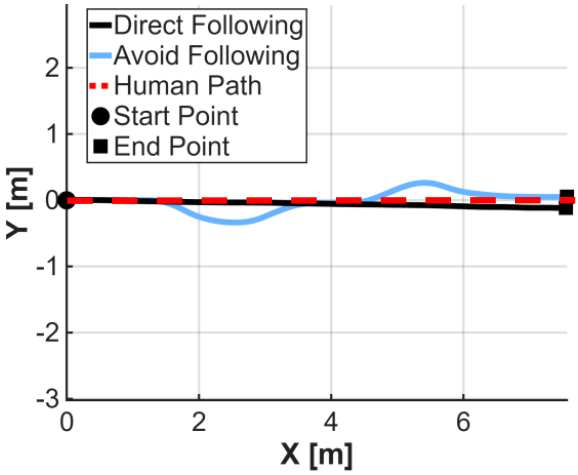

Fig. 7 Case 2: S-curve avoidance

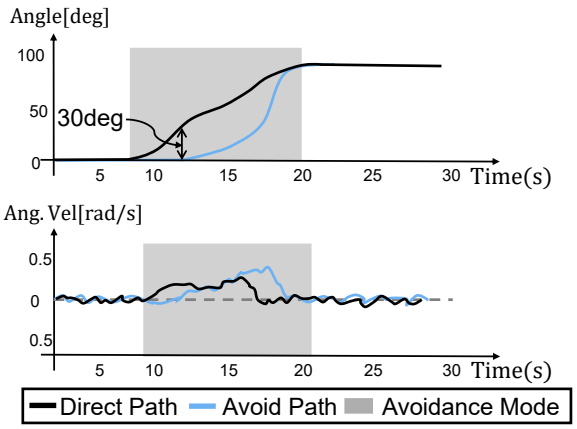

Fig. 8 Angular response in avoidance mode

the robot follows the follower's direction angle θ directly, showing gradual angle changes. In contrast, in Avoidance Following mode, the avoidance angle θ_{avoid} is applied in the section where obstacles are detected within the Safety Corridor, resulting in abrupt angle changes.

Upon entering the avoidance mode, an offset of approximately 30° is instantaneously applied, which represents the

robot's directional change to bypass the obstacle. After the avoidance mode ends, the robot gradually returns to the original following angle, achieving a smooth path transition.

The Angular Velocity graph in Fig. 8 compares the angular velocity characteristics in the two modes. In Direct Following mode, the robot maintains a relatively constant angular velocity while following the follower. In contrast, in Avoidance Following mode, significantly larger angular velocity fluctuations are observed within the avoidance section.

These angular velocity fluctuations are attributed to two factors. First, the application of the avoidance angle causes an abrupt directional change due to the offset. Second, continuous directional adjustments occur during the process of bypassing the obstacle.

However, after the avoidance section, both modes converge to similar angular velocity patterns, confirming that the proposed avoidance algorithm returns stably to Direct Following mode.

5. Conclusion

This paper proposed a driving control algorithm for an autonomous following robot using a 2D LiDAR sensor. The proposed system implemented a dual-mode structure of Direct Following and Avoidance Following to achieve stable following in various environments.

The follower recognition algorithm achieved reliable follower detection through distance-based clustering and multi-stage filtering. By combining object size conditions and continuity conditions, noise and false detection were effectively filtered, and the recognized follower was prevented from switching to another object.

In Direct Following mode, a proportional control method was applied that determines forward velocity and rotational velocity based on the distance and direction to the follower. Acceleration and deceleration control through the Rate Limit Filter and the safe distance maintenance strategy of 0.85 m ensured safe stopping even in situations where the follower suddenly stops.

The Safety Corridor-based avoidance algorithm enabled effective obstacle avoidance without complex path planning. A virtual corridor of 1.5 m width was set, and only obstacles within this area were considered as avoidance targets. This minimized the computational burden, and by applying a 30° avoidance angle, the connection with the follower was maintained while bypassing obstacles.

Experimental results showed that the proposed algorithm demonstrated stable following performance not only in straight following in open areas but also in corner sections and environments with obstacles. The distance to the follower was maintained near the target value of 0.85 m, and even during avoidance actions, it was managed within 1.3 m, so situations where following was interrupted did not occur.

For future research, improvement of follower recognition accuracy through sensor fusion with 3D LiDAR or cameras can be considered. Additionally, the possibility of GPS-based following recovery function in outdoor environments and application to assistive robots for the disabled is also worth exploring.

Acknowledgement

This research was supported by the Bisa Research Grant of Keimyung University in 2025 (Project No. 20240608)

References

[1] A. Eirale, M. Martini, and M. Chiaberge, "Human following and guidance by autonomous mobile robots: A comprehensive review," IEEE Access, 2025.

[2] W. Chi, J. Wang, and M. Q-H. Meng, "A gait recognition method for human following in service robots," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 9, pp. 1429–1440, 2017.

[3] M. Gupta, S. Kumar, L. Behera, and V. K. Subramanian, "A novel vision-based tracking algorithm for a human-following mobile robot," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 7, pp. 1415–1427, 2016.

[4] T.-H. Tsai and C.-H. Yao, "A robust tracking algorithm for a human-following mobile robot," IET Image Processing, vol. 15, no. 3, pp. 786–796, 2021.

[5] T. Yoshimi, M. Nishiyama, T. Sonoura, H. Nakamoto, S. Tokura, H. Sato, F. Ozaki, N. Matsuhira, and H. Mizoguchi, "Development of a person following robot with vision based target detection," in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5286–5291, 2006.

[6] A. Wang, Y. Makino, and H. Shinoda, "Machine learning-based human-following system: Following the predicted position of a walking human," in Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 3131–3136, 2021.

[7] Y. Sung and W. Chung, "Hierarchical sample-based joint probabilistic data association filter for following human legs using a mobile robot in a cluttered environment," IEEE Transactions on Human-Machine Systems, vol. 46, no. 3, pp. 340–349, 2015.

[8] K. Morioka, J.-H. Lee, and H. Hashimoto, "Human-following mobile robot in a distributed intelligent sensor network," IEEE Transactions on Industrial Electronics, vol. 51, no. 1, pp. 229–237, 2004.

[9] J. H. Choi, J. H. Park, S.-Y. Park, and U. H. Baek, "An autonomous human following caddie robot with high-level driving functions," Electronics, vol. 9, no. 9, p. 1516, 2020.

[10] T. Inoue and Y. Okazaki, "Human-following control of omnidirectional autonomous mobile robot by integrating sensor information," in Proc. 23rd International Conference on Control, Automation and Systems (ICCAS), pp. 1–6, 2023.

[11] Z. Zhou, Y. Fu, and W. Wu, "Human-following task without a prior map," Industrial Robot: the international journal of robotics research and application, vol. 51, no. 6, pp. 960–972, 2024.

[12] P. Nikdel, R. Shrestha, and R. Vaughan, "The hands-free push-cart: Autonomous following in front by predicting user trajectory around obstacles," in Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 4853–4859, 2018.

[13] H.-M. Gross, H.-J. Boehme, C. Schroeter, S. Mueller, A. Koenig, E. Einhorn, C. Martin, M. Merten, and A. Bley, "TOOMAS: interactive shopping guide robots in everyday use-final implementation and experiences from long-term field trials," in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2005–2012, 2009.

[14] J. E. Sierra-García, M. Santos, Á. Verdejo, and B. Curto, "Development and experimental validation of control algorithm for person-following autonomous robots," Electronics, vol. 12, no. 9, p. 2077, 2023.