Computer Science and Information Systems 21(2):473–490 https://doi.org/10.2298/CSIS220826008L

Navigation Control of an Autonomous Ackerman Robot in Unknown Environments by Using a Lidar-Sensing-Based Fuzzy Controller

Cheng-Jian Lin[1], Jyun-Yu Jhang[2,⋆] and Chen-Chia Chuang[1]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan
cjlin@ncut.edu.tw
3a617090@gmail.com

[2] Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung 404, Taiwan
jyjhang@nutc.edu.tw

Abstract. In this paper, a real-time navigation control system based on lidar sensing is proposed for use in unknown environments. The proposed system comprises a behavioral controller for controlling an autonomous Ackerman robot for obstacle avoidance in the absence of global map information when moving toward a goal. The adopted obstacle avoidance method is selected by a wall-following fuzzy controller. The input parameter of this controller is the distance between the robot and the wall, which is determined by the lidar sensor, and the output parameter of the controller is the steering angle of the robot for it to reach the destination without collision. To prevent the robot from entering an endless loop, an endless loop escape mechanism is added to the proposed system. The simulation and experimental results of this study indicate that the proposed navigation control system can effectively assist an Ackerman robot to complete the navigation task successfully in unknown environments.

Keywords: Ackerman robot, fuzzy logic controller, lidar, navigation system, unknown environment.

1. Introduction

Autonomous mobile robots is key in the trend toward automation, due to labor shortages, in factories. However, autonomy is difficult to achieve in these robots because of unknown environments and uncertain dynamic obstacles[12], as evident in applications such as self-driving cars [25] and large object manipulation [28],[17]. The navigation control of autonomous mobile robots involves the two steps of goal finding and obstacle avoidance, where are performed using a robust controller [26]. For unknown environments, autonomous robots must perceive environmental information and control the angle and speed of robot movement to reach the destination and automatically avoid obstacles [2].

Many methods have been proposed to solve problems related to robot navigation control; these methods include artificial potential field [11], vector field histogram [5], behavior-based [21], and fuzzy logic [7] methods. Behavior-based methods are widely

⋆ Corresponding author

used in the navigation of autonomous mobile robots [10], and these methods can handle various situations without a global map. In behavior-based methods, autonomous mobile robots engage in wall-following behavior to explore an unknown environment. These robots can move by following the contour and distance information of an object to avoid obstacles and move toward the destination [24],[15]. To control a robot efficiently and stably, fuzzy logic control (FLC) has been incorporated into robot navigation controllers.

Fuzzy theory is used to express the knowledge and experience of experts in the form of language rules to construct a knowledge base and handle uncertain situations [1]. Fuzzy control systems have been used in many domains, including control engineering, signal processing, information processing, and machine intelligence technology [13],[14],[3], [22],[27]. In addition, FLC has proven to be a successful control method for many complex nonlinear systems and has replaced traditional control methods [9]. Mamdani and Assilian [18],[19] designed a fuzzy controller system for controlling a small steam engine. Their experimental results indicate that a fuzzy controller system can achieve better control performance than can a classical controller.

Autonomous mobile robots mostly rely on sensors to measure their relative distance from objects in the environment [6] for perceiving an unknown environment, analyzing and processing environmental information, and making relevant movement decisions. The sensors commonly used in autonomous mobile robots include infrared cameras, sonar, radar, and ultrasonic sensors. However, in a real environment, noise affects the signal captured by a sensor and might lead to wrong decisions. In contrast to the aforementioned sensors, lidar sensors can measure the distance between objects with high precision, identify the shapes of objects, and construct a three-dimensional geographic information model of the surrounding area without being affected by the weather. In the present study, a lidar sensor was adopted to obtain accurate environmental information.

The mobile chassis of autonomous mobile robots are mostly designed with a two-wheel differential structure or omnidirectional wheel structure. The radius and speed of a two-wheel differential structure during steering are determined by the speeds of the two wheels, which can enable circular objects, such as wheels, to be turned on the spot. This structure has relatively strong flexibility but low control precision. The omnidirectional wheel structure can realize omnidirectional walking without changing the body posture [20]. This structure results in very smooth movement but cannot be used in uneven environments. Compared with the aforementioned structures, the Ackerman chassis architecture has higher control precision and smoother movement. Moreover, this architecture allows the robot to move freely in different types of terrain. When the Ackerman architecture is turning, each wheel rotates around the same center; thus, this architecture is not prone to slippage and tire position misalignment [4].

In this paper, a navigation control method is proposed for an autonomous Ackerman robots in unknown environments. The proposed system comprises a behavior controller for controlling an Ackerman robot to achieve obstacle avoidance when heading toward the destination in the absence of global map information. To achieve obstacle avoidance, a wall-following fuzzy controller (WFFC) is used. Furthermore, an escape mechanism is used to prevent the robot from entering an endless loop. Experimental results indicate that the proposed navigation method can complete the navigation task in simulated and real environments. The remainder of this paper is structured as follows. Section 2 illustrates related work. Section 3 introduces the proposed navigation method. Section 4 presents the experimental results obtained in a simulated environment and real environment. Finally, section 5 concludes this study.

2. Related Work

In recent years, the development of autonomous Ackermann robot controller has gained significant attention. This section provides an overview of the related work and advancements in this field.

• Lidar-based Perception and Mapping: Lidar sensors are widely used in autonomous robotics for environment perception and mapping. Researchers have explored the integration of Lidar sensors with Ackermann steering robots to enable accurate and real-time perception of the surroundings. Through Lidar data, the generated 3D environment map was then used for localization and obstacle detection, facilitating autonomous navigation of the Ackermann robot [23].

• Fuzzy Logic Control for Autonomous Navigation: Fuzzy logic controllers have been applied to achieve autonomous navigation in various robotic systems. When combined with Lidar sensing, these controllers can effectively handle uncertainties and variations in the environment. The controller utilized fuzzy rules to interpret Lidar data and generate steering and speed commands, enabling safe and efficient navigation in dynamic environments [16].

• Obstacle Avoidance and Collision Detection: Autonomous Ackermann robots require practical obstacle avoidance and collision detection capabilities to ensure safe navigation. Researchers have developed a fuzzy logic-based collision avoidance system for Ackermann steering robots. By analyzing Lidar data, their system made real-time decisions to avoid obstacles and maintain a safe distance during navigation [8].

In summary, the development of autonomous Ackermann robot controller has seen significant progress. Researchers have focused on perception, control, obstacle avoidance, path planning, and real-world applications. The integration of Lidar sensors with fuzzy control systems provides a powerful approach to achieving autonomous navigation in diverse environments.

3. Navigation Control System for the Autonomous Ackerman Robot

This section introduces the proposed navigation control system for an autonomous Ackerman robot. The proposed navigation system includes a behavior controller that makes an Ackerman robot move toward the destination while avoiding obstacles. The flowchart of this navigation system is shown in Fig. 1. When the behavior controller does not detect any obstacle, it instructs the robot to move toward the destination. By contrast, if this controller detects an obstacle, it switches to the wall-following mode for the robot to avoid the obstacle. However, in the wall-following mode, the robot might encounter an endless terrain loop, which makes the robot unable to successfully complete the navigation task. Therefore, an endless loop escape mechanism is designed to assist the robot to escape an endless loop terrain. Navigation control is completed when the autonomous Ackerman robot reaches its destination.

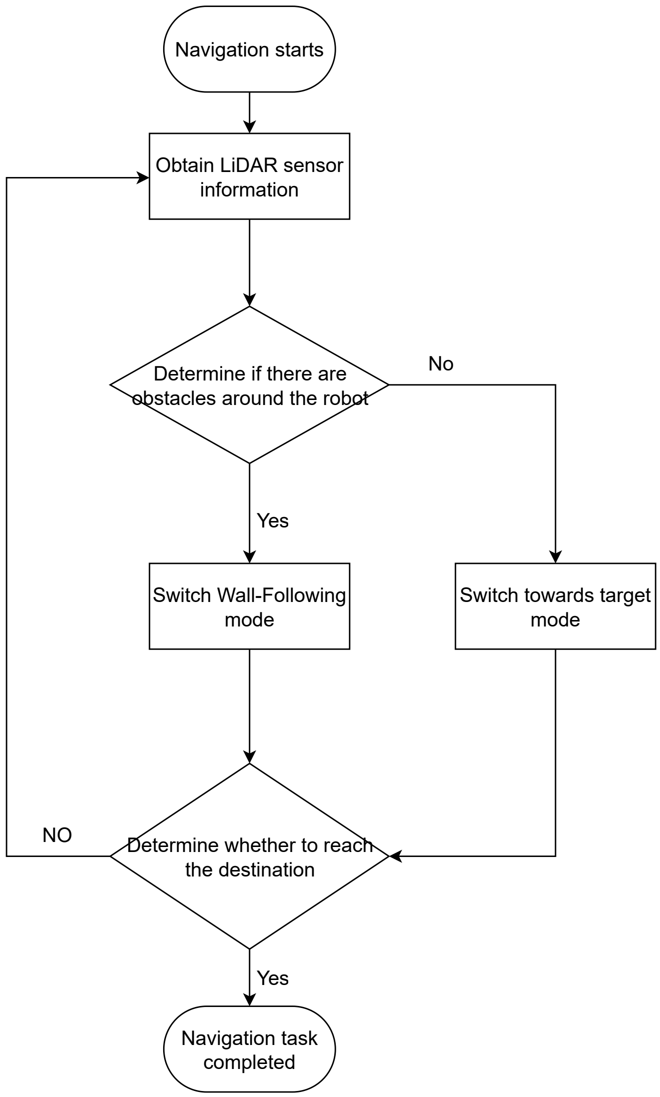


Fig. 1. Flowchart of the proposed navigation system

3.1. Autonomous Ackerman Robot

We independently developed the autonomous Ackerman robot used in this study. The adopted robot uses a Velodyne Puck (VLP-16) lidar sensor to scan for surrounding obstacles and an edge-embedded device [NVIDIA Jetson AGX Xavier (AGX)] to conduct real-time data processing. The sensing range of VLP-16 is 0.5 to 5 m, and its horizontal angular measurement range is 360°. AGX uses Ubuntu 16.04 and the Robot Operating System (ROS) to drive the robot's motor system. Through control commands, the movement speed and turning angle of the robot are controlled. In addition, the robot chassis has the Ackerman architecture for it to move smoothly when handling heavy objects and uneven terrain. The designed autonomous Ackerman robot is shown in Fig. 2.

Fig. 2. Designed autonomous Ackerman robot

3.2. Behavior Controller

The behavioral controller plays a decision-making role in the proposed navigation system. This controller switches between the toward-goal mode and wall-following mode depending on the robot position and environment. To detect the position of an obstacle, the sensing area of the lidar sensor is divided into three areas, denoted A1, A2 and A3 (Fig. 3). A1 is the front-right area of the robot, A2 is the front-left area of the robot, and A3 is the rear of the robot. When an obstacle is detected in A1 or A2, the behavior controller switches from the toward-goal mode to the wall-following mode. The behavior controller remains in the toward-goal mode as long as the lidar sensor does not detect any obstacles in A1 and A2.

A. Toward-Goal Mode

When no obstacle is detected in front of the autonomous Ackerman robot, the robot moves toward the goal. As displayed in Fig. 4., the autonomous Ackerman robot calculates the steering angle according to its current position and the goal position, then turns toward the goal position, and then moves straight toward the goal. The designed Ackerman architecture has a turning angle between 45° and −45°; thus, the maximum angle of left and right turns is 45°.

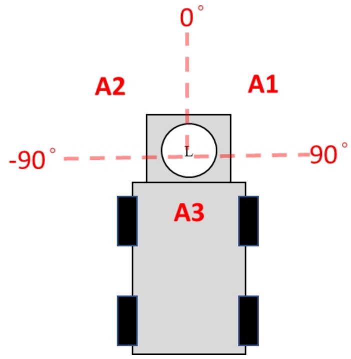


Fig. 3. Three obstacle detection areas for the autonomous Ackerman robot

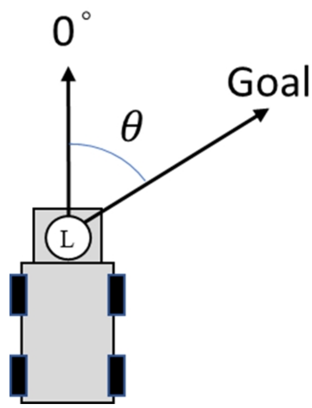


Fig. 4. Angle between the autonomous Ackerman robot and the goal

B. Wall-Following Mode

If an obstacle is detected in front of the autonomous Ackerman robot, the behavior controller switches to the wall-following mode to instruct the robot to move along the object until the object has been passed. To achieve this behavior, a fuzzy controller with a wall-following function, namely a WFFC, is designed. Fig. 5 displays the system flow of the wall-following mode. First, the lidar sensor detects the distance to obstacles around the robot. Subsequently, the distance information is used as the input of the controller to obtain the steering angle of the robot as the output. The proposed WFFC contains four parts: a fuzzifier, fuzzy rule base, fuzzy inference engine, and defuzzifer. The basic structure of the proposed WFFC is displayed in Fig. 6.

The various parts of the WFFC are detailed in the following text.

● Fuzzifier

A fuzzifier maps a crisp value to a fuzzy number (i.e., a real number between 0 and 1). This process is called fuzzification, and fuzzy logic better accords with human cognition

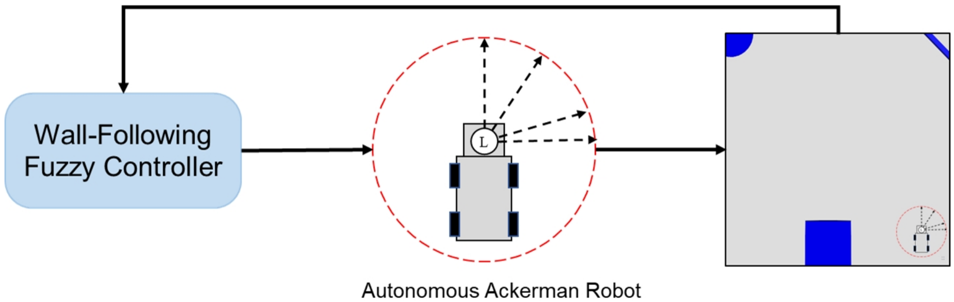


Fig. 5. System flow of the wall-following mode

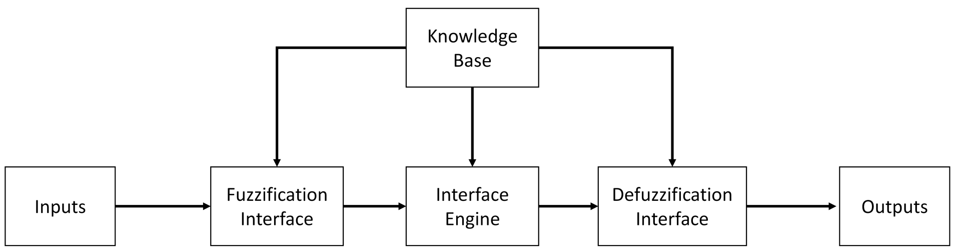


Fig. 6. Architecture of the proposed WFFC

relative to classical (bivalent) logic. Membership functions are used to evaluate the degree of each input of a system. Triangular or trapezoidal membership functions are most commonly used membership functions in fuzzy systems (Figs. 7 and 8). These membership functions are constructed using straight lines. Please review whether the edits convey your intended meaning accurately.Compared with Gaussian membership functions, linear membership functions are simpler and thus enable the design of simpler and more computationally lightweight robot controllers. Therefore, triangular and trapezoidal membership functions were used to design the robot controller in this study. The membership function of fuzzy set A can be defined as $\mu_A(x)$, where $\mu_A(x)$ denotes the degree of input x from fuzzy set A. A triangular membership function contains three parameters, namely a_1, a_2, and b_1, which denote the positions of the left boundary, Please specify which vertex of the triangle is being referred to here.triangle vertex, right boundary, respectively. The definition of a triangular membership function is provided in Equation 1. Trapezoidal membership functions contain four parameters, namely a_1, a_2, b_1, and b_2, which represent the positions of the left boundary, right boundary, left triangle vertex, and right triangle vertex, respectively. The definition of trapezoidal membership is provided in Equation 2.

$$\mu_A(x) = \begin{cases} 0 & x \leq a_1 \\ \frac{x-a_1}{a_2-a_1} & a_1 < x < a_2 \\ \frac{b_1-x}{b_1-a_2} & a_2 < x < b_1 \\ 0 & b_1 \leq x \end{cases} \tag{1}$$

$$\mu_A(x) = \begin{cases} 0 & x \leq a_1 \\ \frac{x-a_1}{a_2-a_1} & a_1 < x < a_2 \\ 1 & a_2 < x < b_1 \\ \frac{b_1-x}{b_1-a_2} & b_1 < x < b_2 \\ 0 & b_2 \leq x \end{cases} \tag{2}$$

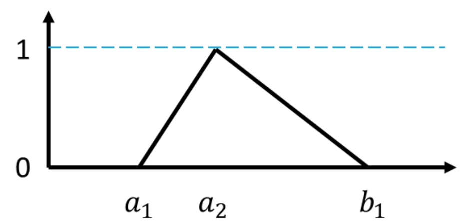


Fig. 7. Triangular membership function

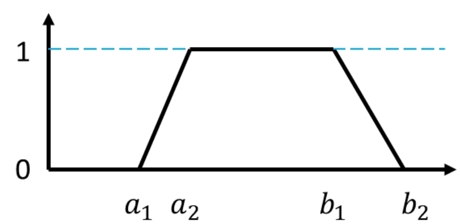


Fig. 8. Trapezoidal membership function

The scanning angle range of a lidar sensor is 360°, and A1 covers an angular range of 90°. Therefore, if lidar information is acquired for each degree in A1, 90 input data are obtained. To reduce the quantity of input data, only the lidar sensing data at 0°, 30°, 75°, and 90° are used as input (denoted as L1, L2, L3, and L4, respectively). Fig. 9 shows the four directions sensed by the lidar in A1.

Three membership functions can be used to define the distance of objects sensed by the lidar sensor as near, normal, or far. The detection angle of L1 is close to that of L2, and the detection angle of L3 is close to that of L4. Therefore, the same membership function is used for the forward (L1) and obliquely forward (L2) directions, and the same membership function is used for the forward-right (L3) and right (L4) directions. The membership functions of lidar for forward and rightward sensing are displayed in Figs. 10 and 11, respectively. The Ackerman architecture requires a large radius of gyration when turning. Therefore, in the membership function for forward sensing (Fig. 10), a sensing distance of greater than 4.25 m indicates that the obstacle is located far away from the

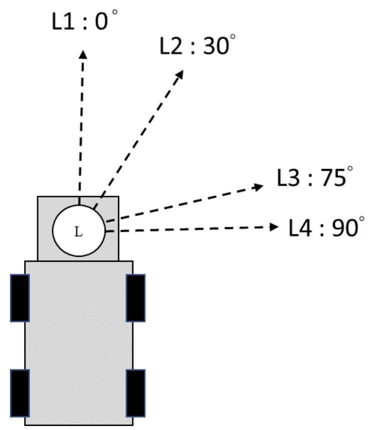


Fig. 9. Four directions sensed by the lidar sensor

robot. By contrast, a sensing distance of less than 3 m indicates that the object is close to the robot. In the membership function for rightward sensing (Fig. 11), a sensing distance of less than 2 m indicates that the robot is close to the obstacle. Moreover, a sensing distance of greater than 2 m indicates that the robot is located far from the obstacle.

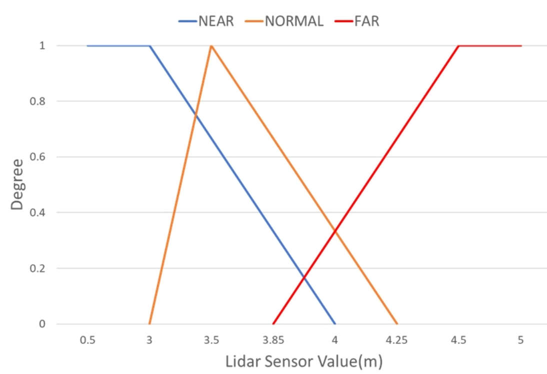


Fig. 10. Membership functions for forward sensing by the lidar sensor (L1 and L2)

• Fuzzy Rule Base and Fuzzy Inference Engine

If–then rule statements are adopted to construct a fuzzy rule base. A fuzzy rule can be defined as follows:

R_i: If x_1 is A_1 and x_2 is $A_2 \ldots$ and x_n is A_n, then y is b_i,

where x and y are linguistic variables. Because the robot might encounter different terrains during its movement, appropriate fuzzy rules must be designed for different conditions. Fig. 12 illustrates the obstacles located ahead of the robot at an acute angle, a right

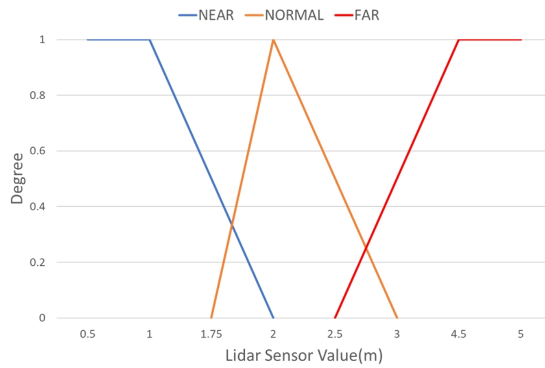


Fig. 11. Membership functions for rightward sensing by the lidar sensor (L3 and L4)

angle, and an obtuse angle. According to the distances sensed by L1 and L2, the state of the obstacle can be obtained to determine the turning range. Fig. 13 depicts the state of the robot and obstacles. In Fig. 13, three robot–obstacle conditions are observed: robot is parallel to the obstacle, robot is located close to the obstacle, and robot is located far away from the obstacle. On the basis of the values sensed by L3 and L4, the angle of the robot body can be adjusted.

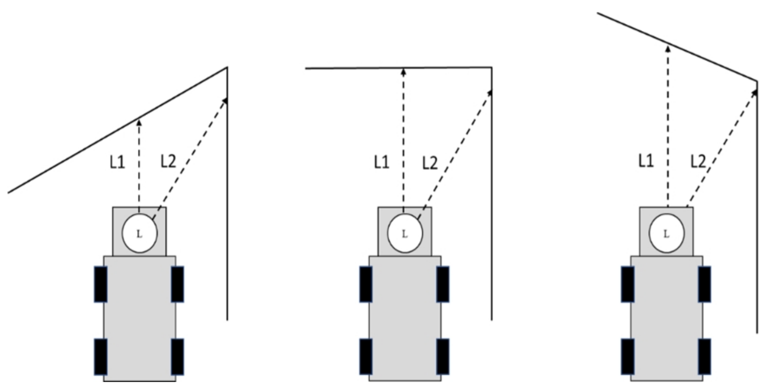


Fig. 12. Obstacles located ahead of the robot at three angles

On the basis of the aforementioned conditions, 21 fuzzy rules were designed (Table 1). The inputs of the proposed wall-following controller are four distance variables, namely those sensed by L1–L4, and the output is the Ackerman turning angle (θ), which is between 45° and $-45°$. This angle is mapped to a value between 1 and -1 (ω) by using Equation 3. In this study, the AND operation is used for fuzzy rule computation.

$$\omega = \frac{\theta}{45^\circ} \quad (3)$$

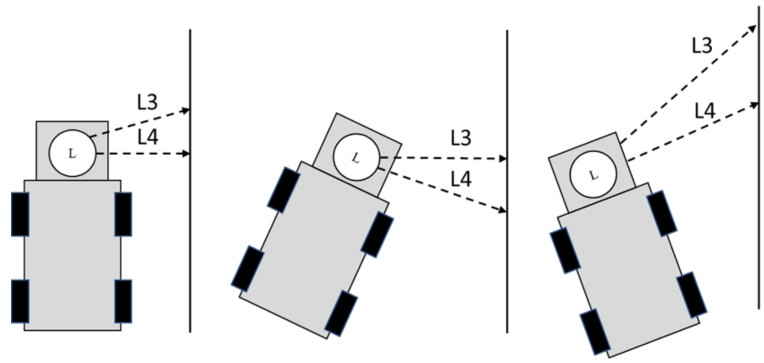


Fig. 13. State of the robot and obstacles

Table 1. Twenty-one fuzzy rules of the proposed wall-following controller

Intput				Output
L1	L2	L3	L4	ω
ANY	ANY	ANY	NEAR	0.7
ANY	ANY	ANY	NORMAL	0
ANY	ANY	ANY	FAR	-0.7
NEAR	NEAR	ANY	ANY	0.7
NEAR	NORMAL	ANY	ANY	0.7
NEAR	FAR	ANY	ANY	0.7
NORMAL	NEAR	ANY	ANY	0.7
NORMAL	NORMAL	ANY	ANY	0.6
NORMAL	FAR	ANY	ANY	0.6
FAR	NEAR	ANY	ANY	0.5
FAR	NEAR	ANY	ANY	0
FAR	FAR	ANY	ANY	-0.3
ANY	ANY	NEAR	NEAR	0.5
ANY	ANY	NEAR	NORMAL	0.3
ANY	ANY	NEAR	FAR	0.3
ANY	ANY	NORMAL	NEAR	0.1
ANY	ANY	NORMAL	NORMAL	0
ANY	ANY	NORMAL	FAR	-0.1
ANY	ANY	FAR	NEAR	0.1
ANY	ANY	FAR	NORMAL	-0.2
ANY	ANY	FAR	FAR	-0.6

• Defuzzifier

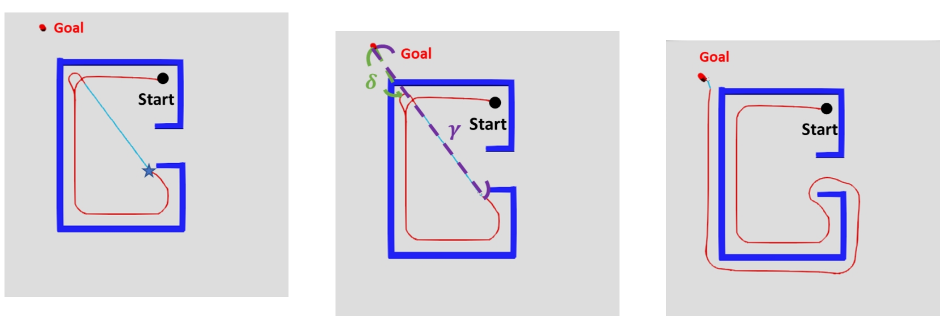
The output of a defuzzifier is a crisp value. The centroid defuzzification process can be expressed as follows:

$$y = \frac{\sum_{i=1}^{21} \mu_A(x_i)\omega_i}{\sum_{i=1}^{21} \mu_A(x_i)} \tag{4}$$

where y represents the output of the wall-following controller, $\mu_A(x_i)$ is the firing strength of the ith rule, and ω is the fuzzy value of the robot turning angle (between -1 and 1).

3.3. Endless Loop Escape Mechanism

To prevent the autonomous Ackerman robot from falling into an endless loop, an endless loop escape mechanism is proposed (Fig. 14). In Fig. 14(a), the robot faces no obstacle when it moves toward the star point along the wall. At this time, the behavior controller switches to the toward-goal mode and cause the robot to fall into an endless loop. In the proposed endless loop escape mechanism, the shortest distance (δ) between the robot and the goal is recorded to determine which behavioral mode should be employed. When the robot moves to the star point, the current distance from the robot to the goal (γ) is greater than δ, and the behavior controller executes the wall-following mode. Until δ is greater than γ, the behavior controller functions in the toward-goal mode to escape the endless loop, as displayed in Fig 14(c).



(a) Trapped in an endless loop **(b)** Escape mechanism **(c)** Escaping the endless loop

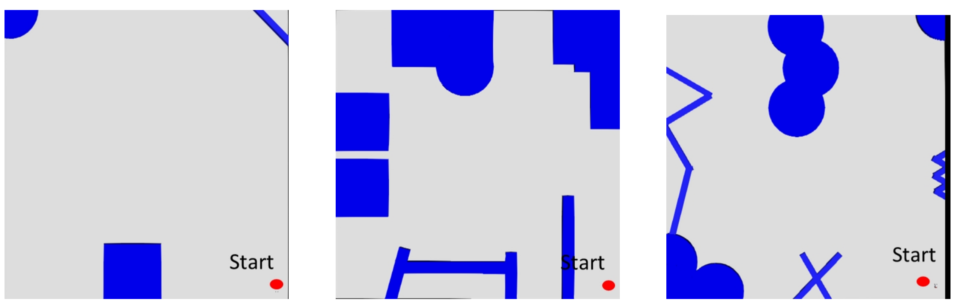
Fig. 14. Proposed endless loop escape mechanism

4. Simulation and Experimental Results

To verify the performance of the proposed navigation method, an open-source robot simulator (Webots) was used to construct a testing environment. Webots is a development environment software for the modeling, programming, and simulation of mobile robots, and this software program can run on Linux, Windows, and macOS. The proposed robot controller can be programmed in C, C++, Python, Java, MATLAB, or the ROS by using a simple application programming interface that covers all basic robot control techniques. Simulation experiments were conducted in six environments to test the effectiveness of the proposed method, and the size of each simulated environment was 40 m 40 m. Finally, the proposed method was used to complete a navigation task with an autonomous Ackerman robot in a real environment.

4.1. Simulation Results Obtained for the WFFC

We designed three environments with small numbers of square and circular obstacles to test the performance of the proposed WFFC (Fig. 15). The first environment consisted of simple circular objects, hypotenuses, and square obstacles. The second environment was mainly composed of square obstacles to test the robot's obstacle avoidance performance at right angles. Finally, the third environment was composed of circular obstacles and special concave corners to test whether the robot could effectively avoid concave corners.



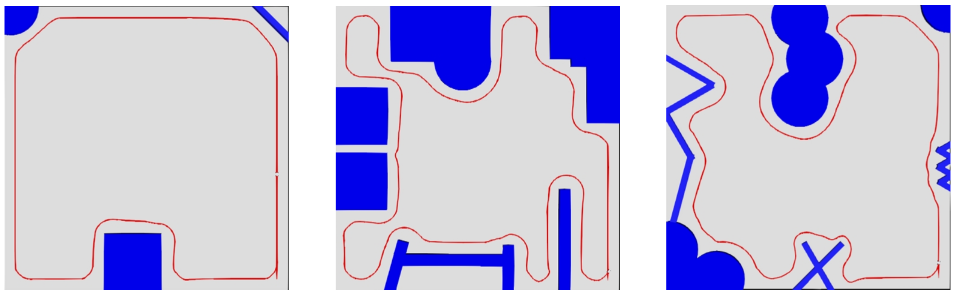
(a) Few obstacles **(b)** Square obstacles **(c)** Circular obstacles

Fig. 15. Three testing environments for the proposed WFFC

Fig. 16 illustrates the paths of the autonomous Ackerman robot in the aforementioned three testing environments when using the proposed WFFC. The robot successfully circumnavigated the three environments without collision.

4.2. Simulation Results for Navigation Control

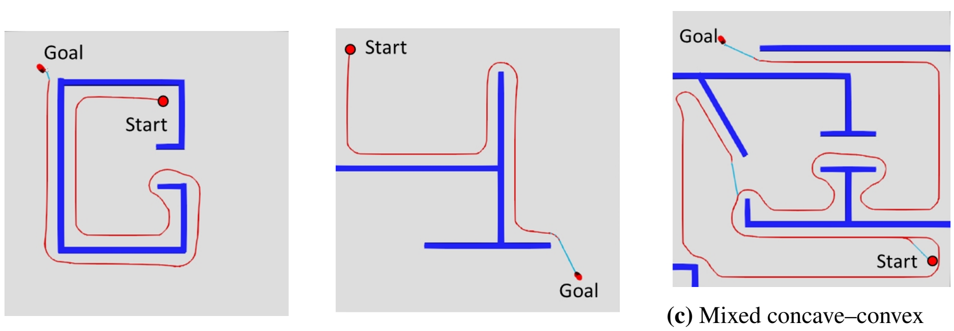
Three testing environments, namely a concave environment, convex environment, and mixed concave–convex environment, were designed to evaluate the navigation control performance achieved with the proposed method. These environments are displayed in Fig. 17. The red movement path in Fig. 17 represents the behavior controller executing the wall-following mode, whereas the blue movement path denotes this controller executing the toward-goal mode. As displayed in Fig. 17(b), when the navigation started, the behavior controller executed the toward-goal mode because no obstacles were detected



(a) Environment 1 (b) Environment 2 (c) Environment 3

Fig. 16. Movement paths of the autonomous Ackerman robot in the three testing environments designed for the proposed WFFC

ahead of the robot (blue path). When an obstacle was detected, the controller switched to the wall-following mode and moved the robot along the red path. Fig. 17(c) indicates that the proposed endless loop escape mechanism effectively assisted the robot to escape an endless loop terrain and complete the navigation task in an unknown environment.



(a) Concave environment **(b)** Convex environment **(c)** Mixed concave–convex environment

Fig. 17. Navigation paths of the autonomous Ackerman robot in three testing environments

4.3. Experimental Results in a Real Environment

The proposed navigation control method was implemented for an autonomous Ackerman robot in a real environment. Fig. 18 shows the floor plan of the real testing environment. Several square and circular obstacles were included in the real environment to verify the effectiveness of the proposed navigation control method. As displayed in Fig 19, the autonomous Ackerman robot sensed obstacles in detection area A1. Therefore, the behavior controller executed the wall-following mode. Fig. 20 displays the computation time for each time step. As displayed in Fig. 20, the computation time for each step was between

0.275 and 0.295 s. Thus, the proposed navigation control method can realize real-time computation. The results also indicate that the proposed method can be effectively applied in unknown environments without the need for complex global map construction and model training.

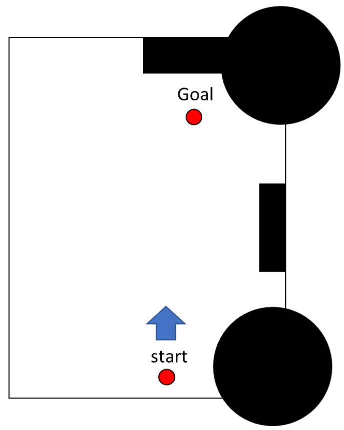


Fig. 18. Floor plan of the real testing environment

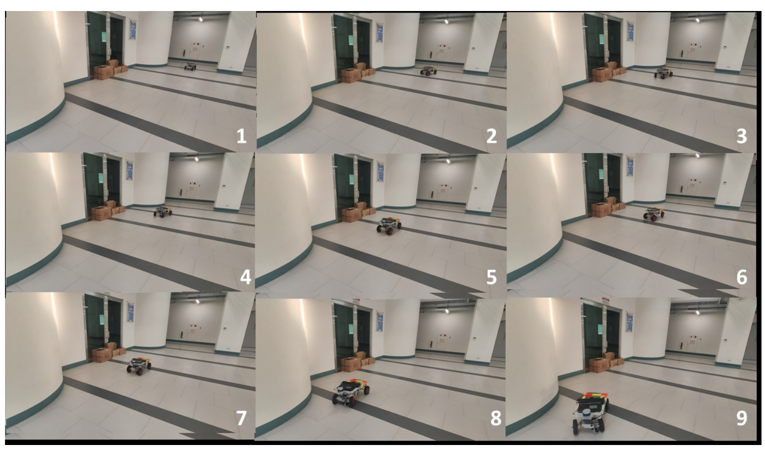


Fig. 19. Navigation control in the real environment

5. Conclusion

In this paper, an effective navigation control method is proposed for autonomous Ackerman robots moving in unknown environments. The proposed method can accomplish

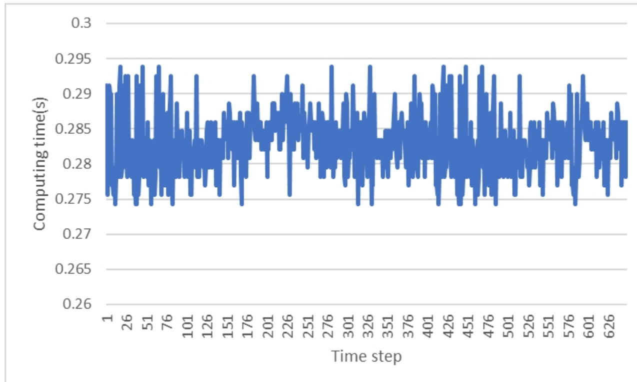


Fig. 20. Computation time for each time step

the navigation task without the construction of a global map or the training of a complex model. The designed behavior controller enables an autonomous Ackerman robot to undertake obstacle avoidance and complete the navigation task automatically according to the current environment state. Furthermore, the computation time per time step of the proposed method is less than 0.3 s, which indicates that the proposed method has real-time computation capability. Simulation and experimental results indicated that the proposed navigation control method can enable an autonomous Ackerman robot to complete the navigation task effectively without collision in an unknown environment. In a future study, we will consider applying the developed autonomous Ackerman robot to practical applications.

Acknowledgments. The authors would like to thank the National Science and Technology Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSTC 111-2222-E-025-001.

References

1. Abdelazim, T., Malik, O.: An adaptive power system stabilizer using on-line self-learning fuzzy systems. In: 2003 IEEE Power Engineering Society General Meeting (IEEE Cat. No.03CH37491). vol. 3, pp. 1715–1720 Vol. 3 (2003)
2. Bao, Q.y., Li, S.m., Shang, W.y., An, M.j.: A fuzzy behavior-based architecture for mobile robot navigation in unknown environments. In: 2009 International Conference on Artificial Intelligence and Computational Intelligence. vol. 2, pp. 257–261 (2009)
3. Batur, C., Kasparian, V.: Model based fuzzy control. Mathematical and Computer Modelling 15(12), 3–14 (1991)
4. Carpio, R.F., Potena, C., Maiolini, J., Ulivi, G., Rosselló, N.B., Garone, E., Gasparri, A.: A navigation architecture for ackermann vehicles in precision farming. IEEE Robotics and Automation Letters 5(2), 1103–1110 (2020)
5. Chen, W., Wang, N., Liu, X., Yang, C.: Vfh based local path planning for mobile robot. In: 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI). pp. 18–23 (2019)

6. Discant, A., Rogozan, A., Rusu, C., Bensrhair, A.: Sensors for obstacle detection - a survey. In: 2007 30th International Spring Seminar on Electronics Technology (ISSE). pp. 100–105 (2007)
7. Doitsidis, L., Valavanis, K., Tsourveloudis, N.: Fuzzy logic based autonomous skid steering vehicle navigation. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292). vol. 2, pp. 2171–2177 vol.2 (2002)
8. Elsayed, H., Abdullah, B.A., Aly, G.: Fuzzy logic based collision avoidance system for autonomous navigation vehicle pp. 469–474 (2018)
9. Feng, G.: A survey on analysis and design of model-based fuzzy control systems. IEEE Transactions on Fuzzy Systems 14(5), 676–697 (2006)
10. Ganapathy, V., Yun, S.C., Ng, J.: Fuzzy and neural controllers for acute obstacle avoidance in mobile robot navigation. In: 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. pp. 1236–1241 (2009)
11. Hwang, Y., Ahuja, N.: Path planning using a potential field representation. In: Proceedings. 1988 IEEE International Conference on Robotics and Automation. pp. 648–649 vol.1 (1988)
12. Kan, X., Teng, H., Karydis, K.: Online exploration and coverage planning in unknown obstacle-cluttered environments. IEEE Robotics and Automation Letters 5(4), 5969–5976 (2020)
13. Lee, C.: Fuzzy logic in control systems: fuzzy logic controller. i. IEEE Transactions on Systems, Man, and Cybernetics 20(2), 404–418 (1990)
14. Lee, C.: Fuzzy logic in control systems: fuzzy logic controller. ii. IEEE Transactions on Systems, Man, and Cybernetics 20(2), 419–435 (1990)
15. Li, Y.J., Chou, W.C., Chen, C.Y., Shih, B.Y., Chen, L.T., Chung, P.Y.: The development on obstacle avoidance design for a humanoid robot based on four ultrasonic sensors for the learning behavior and performance. In: 2010 IEEE International Conference on Industrial Engineering and Engineering Management. pp. 376–379 (2010)
16. Lin, C.J., Chang, M.Y., Tang, K.H., Huang, C.K.: Navigation control of ackermann steering robot using fuzzy logic controller. Sensors and Materials 35, 781 (2023)
17. Liu, O., Yuan, S., Li, Z.: A survey on sensor technologies for unmanned ground vehicles. In: 2020 3rd International Conference on Unmanned Systems (ICUS). pp. 638–645 (2020)
18. Mamdani, E.: Application of fuzzy algorithms for control of simple dynamic plant. Proceedings of the Institution of Electrical Engineers 121, 1585–1588(3) (December 1974)
19. Mamdani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7(1), 1–13 (1975)
20. Morales, S., Magallanes, J., Delgado, C., Canahuire, R.: Lqr trajectory tracking control of an omnidirectional wheeled mobile robot. In: 2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA). pp. 1–5 (2018)
21. Motlagh, O.R.E., Hong, T.S., Ismail, N.: Development of a new minimum avoidance system for a behavior-based mobile robot. Fuzzy Sets and Systems 160(13), 1929–1946 (2009), theme: Information Processing and Applications
22. Pedrycz, W.: Fuzzy Control and Fuzzy Systems (2nd, Extended Ed.). Research Studies Press Ltd., GBR (1993)
23. Pendleton, S.D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y.H., Rus, D., Ang, M.H.: Perception, planning, control, and coordination for autonomous vehicles. Machines 5(1) (2017)
24. Peng, J., Yumei, H.: Behavior-based avoiding barriers system of mobile robot. In: 2009 WRI World Congress on Software Engineering. vol. 3, pp. 106–112 (2009)
25. Said, H.B., Marie, R., Stéphant, J., Labbani-Igbida, O.: Skeleton-based visual servoing in unknown environments. IEEE/ASME Transactions on Mechatronics 23(6), 2750–2761 (2018)
26. Shayestegan, M., Din, S.: Fuzzy logic controller for robot navigation in an unknown environment. In: 2013 IEEE International Conference on Control System, Computing and Engineering. pp. 69–73 (2013)
27. Sugeno, M.: On stability of fuzzy systems expressed by fuzzy rules with singleton consequents. IEEE Transactions on Fuzzy Systems 7(2), 201–224 (1999)

28. Tsuru, M., Escande, A., Tanguy, A., Chappellet, K., Harad, K.: Online object searching by a humanoid robot in an unknown environment. IEEE Robotics and Automation Letters 6(2), 2862–2869 (2021)

Cheng-Jian Lin received the B.S. degree in electrical engineering from the Ta Tung Institute of Technology, Taipei, Taiwan, in 1986, and the M.S. and Ph.D. degrees in electrical and control engineering from National Chiao-Tung University, Taiwan, in 1991 and 1996, respectively. Currently, he is a Chair Professor of the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung, Taiwan, and the Dean of the Intelligence College, National Taichung University of Science and Technology, Taichung. His current research interests include machine learning, pattern recognition, intelligent control, image processing, intelligent manufacturing, and evolutionary robots.

Jyun-Yu Jhang received the B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, in 2015, and the Ph.D. degree in electrical and control engineering from National Chiao-Tung University, Taiwan, in 2021. He is currently an Assistant Professor with the Computer Science and Information Engineering Department, National Taichung University of Science and Technology, Taichung. His current research interests include fuzzy logic theory, type-2 neural fuzzy systems, evolutionary computation, machine learning, and computer vision and application.

Chen-Chia Chuang received the B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, in 2023. His current research interests are machine learning, pattern recognition, and image processing.

Received: August 26, 2023; Accepted: July 06, 2023.