IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 August 2023, accepted 10 September 2023, date of publication 15 September 2023, date of current version 6 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3315741

RESEARCH ARTICLE

Type 2 Fuzzy Neural Controller for Navigation Control of an Ackermann Steering Vehicle

CHENG-JIAN LIN[1,2], **(Senior Member, IEEE), BING-HONG CHEN**[2], **AND JYUN-YU JHANG**[3]

[1]Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan
[2]Ph.D. Program, Prospective Technology of Electrical Engineering and Computer Science, National Chin-Yi University of Technology, Taichung 411, Taiwan
[3]Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung 404, Taiwan

Corresponding author: Cheng-Jian Lin (cjlin@ncut.edu.tw)

This work was supported by the National Science and Technology Council of the Republic of China under Grant MOST 110-2221-E-167-031-MY2 and Grant NSTC 111-2222-E-025-001-.

ABSTRACT In this study, we proposed a type 2 fuzzy neural controller (FNC) based on Bayesian dynamic group particle swarm optimization (BDGPSO) for the navigation control of Ackermann steering vehicles. The type 2 FNC has a five-layer network architecture. The advantages of the proposed BDGPSO algorithm are that it prevents PSO from falling into a local optimum, and it uses the Bayesian algorithm to determine the optimal inertia weight and learning factor combination. To evaluate the proposed type 2 FNC on the basis of the distance information returned by lidar, we used a new fitness function from the reinforcement-learning strategy. In an unknown testing environment, the fitness value and time required for the proposed type 2 FNC with angular and linear velocity outputs were 0.973510 and 25.555 s, respectively. These values indicated that the proposed approach outperforms other wall-following control methods. In addition, the proposed controller could successfully control the navigation of an Ackerman steering vehicle in an unknown actual environment.

INDEX TERMS Ackermann steering vehicle, type 2 fuzzy neural controller, particle swarm optimization, lidar, navigation control.

I. INTRODUCTION

Controlling the navigation of vehicles and robots is a crucial topic. Researchers have used various sensors to realize the navigation control of vehicles or robots used in product handling as well as autonomous vehicles. This is why high-precision and high-reliability sensors have become a highly critical component in the navigation control of vehicles and robots. Liu et al. [1] discussed the advantages and disadvantages of various sensors, including infrared, camera, radar, and lidar sensors. Infrared sensors collect data from their environment by detecting the infrared radiation signal of an object, but they cannot generate depth information, and their resolution is relatively low in application [2]. Camera sensors acquire color and texture data from their environment, but they cannot obtain depth data and are sensitive to light and weather conditions. Radar sensors have a large detection range and are small but are too expensive. Lidar sensors [3] have thus often been used to collect environmental data. This is because lidar offers a long detection distance, a wide viewing angle range, high data accuracy, target depth information, and insensitivity to light.

Common navigation control methods for vehicles or robots include supervised learning and reinforcement learning. In supervised learning, the navigation system achieves navigation and obstacle avoidance control by collecting a large amount of training data. Wellhausen et al. [4] projected the footholds of a robot's trajectory into a camera. In this scenario, the robot's motion control system used these footholds as markers to collect relevant data for training a neural network. Once the foothold of the terrain was predicted, the robot could navigate autonomously. According to Kahn et al. [5], navigation is a geometric problem. The objective of a robot is to perceive the geometric shape of its environment and plan a collision-free path toward the target. Therefore, a Berkeley autonomously driving ground robot [5] capable of collecting

The associate editor coordinating the review of this manuscript and approving it for publication was M. Venkateshkumar.

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

autonomous environmental markers was proposed. When entering a new environment, this robot can use different terrains for training and also collect additional data to improve itself. However, navigation control with supervised learning has certain disadvantages, such as a time-consuming data collection process and the need to recollect data for training when the environment changes.

Yudha et al. [6] compared two methods, a fuzzy logic controller and a neural network, in the navigation trajectory control of a navigation system with reinforcement learning. Wang et al. [7] proposed a combination of a fuzzy neural network (FNN) and an enhanced Kalman filter for dynamic adaptive network parameter adjustment. They reported that integrating the Kalman filter considerably improved the generalizability and convergence speed of the FNN. Faisal et al. [8] combined a fuzzy logic controller and a particle swarm optimization (PSO) algorithm to determine the behavioral decisions of a robot. Using the data collected by the robot's ultrasonic sensor array, they confirmed that the adaptation function of the current vehicle movement avoided obstacles and led to following of a navigation path. Sahloul et al. [9] used two algorithms, PSO and the genetic algorithm (GA), to achieve navigation control by adjusting the membership function and fuzzy rules of a fuzzy logic controller. The experimental results presented in [9] indicated that the performance of the GA is higher than that of PSO, whereas the convergence speed of PSO is higher than that of the GA. Recent evidence also indicates that the performance of type 2 fuzzy controllers is higher than that of type 1 fuzzy controllers [10], [11]. Therefore, many researchers have implemented type 2 fuzzy controllers in navigation control. For example, Saidi et al. [10] proposed robust waypoint navigation using a type 2 fuzzy controller for navigation control. They reported that, compared with type 1 fuzzy controllers, type 2 fuzzy controllers are more robust and have higher convergence speed. Sharma and Bajaj [11] analyzed the performance of type 1 and type 2 FNNs. They reported that type 2 FNNs can address incomplete vehicle information and are better suited for navigation control. Lin et al. [12] used a type 2 FNN to address the problem of environmental disturbances. Son et al. [13] used the back propagation algorithm to tune the parameters of a type 2 FNN. Although the back propagation algorithm has high convergence speed, it easily falls into a regional optimum. Therefore, some researchers have used evolutionary algorithms to find the global optimum. For example, Baklouti and Alimi [14] used PSO to tune the parameters of a type 2 FNN.

Compared to other traditional controllers, type 2 FNC offers several advantages including: 1. robustness to uncertainties: the type 2 fuzzy logic allows for handling uncertainties and variations which provides a higher level of robustness and tolerance to uncertainties in the input data and system dynamics. 2. increased modeling flexibility: type 2 fuzzy sets offer enhanced modeling capabilities by capturing higher degrees of uncertainty and ambiguity that allows for a more accurate representation of complex and uncertain systems. 3. handling nonlinearity and complex relationships: type 2 FNC is well-suited for dealing with nonlinear and complex relationships between inputs and outputs. They can effectively model and control systems with nonlinear dynamics and interactions. While type 2 FNC has several advantages, there are also some limitations and potential disadvantages to consider: 1. type 2 FNC involves more complex computations compared to traditional fuzzy systems. This can result in increased computational requirements. 2. The design type 2 fuzzy rules heavily rely on expert knowledge and empirical approaches, making it difficult to design an effective rule base.

The PSO algorithm is widely used to tune the parameters of type 1 and type 2 FNNs. However, the network output is determined by the parameter settings (i.e., inertia weights and learning factors) of the PSO algorithm. Therefore, determining the optimal inertia weight and learning factor is essential. Chen et al. [15] proposed an adaptive inertia weight method for determining the optimal inertia weight and learning factor. However, when applied to different network architectures, their method failed to identify the optimal solution. Liu et al. [16] proposed a method of linearly descending inertial weights. Although this method performed more highly than the adaptive inertia weight method, it still fell into a local optimum. Ren and Liu [17] proposed a dynamic adjustment of inertia weights and learning factors. They set the inertia weight and learning factor to be automatically adjusted within an interval and determined the optimal combination.

In this study, we proposed a type 2 fuzzy neural controller (FNC) based on Bayesian dynamic group PSO (BDGPSO) for the navigation control of Ackermann steering vehicles. The type 2 FNC has a five-layer network architecture. The advantages of the proposed BDGPSO algorithm are that it prevents PSO from falling into a local optimum and it uses the Bayesian algorithm to determine the optimal inertia weight and learning factor combination. We also adopt a reinforcement-learning strategy and define a new fitness function. This fitness function is a controller evaluation method and is based on the distance information returned by lidar. Finally, the angular velocity (AV) and linear velocity (LV) of the type 2 FNC output are used to complete the navigation control of an Ackerman steering vehicle in an unknown environment.

The remainder of this paper is organized as follows. In Section II, the architecture of the type 2 FNC and proposed BDGPSO algorithm are presented. In Section III, the simulation experiment results of wall-following control and navigation control of an Ackerman steering vehicle with lidar sensing are presented and compared with those obtained using other methods. In Section IV, the experimental results obtained for the Ackerman steering vehicle in an actual environment are presented. Finally, in Section V, the study conclusions and future research recommendations are presented.

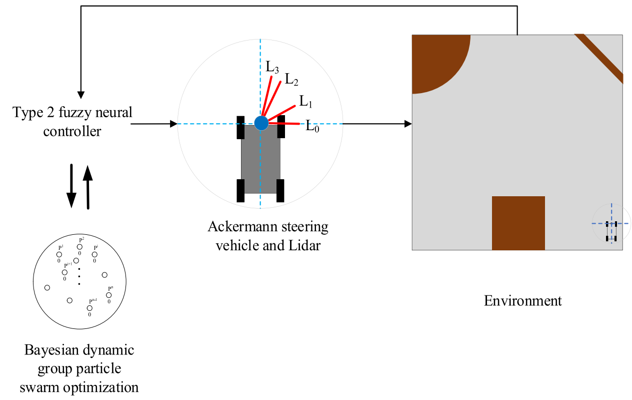
Type 2 fuzzy neural controller
L3
L2
L1
L0
Ackermann steering vehicle and Lidar
Environment
Bayesian dynamic group particle swarm optimization

FIGURE 1. Architecture of the navigation control system.

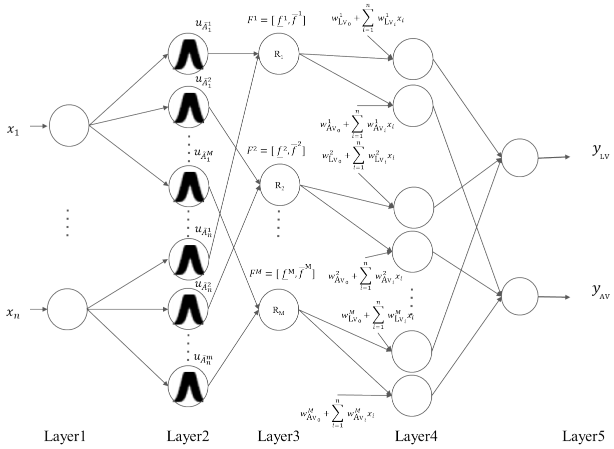
x_1
x_n
$u_{\tilde{A}_1^1}$
$u_{\tilde{A}_1^2}$
$u_{\tilde{A}_1^M}$
$u_{\tilde{A}_n^1}$
$u_{\tilde{A}_n^2}$
$u_{\tilde{A}_n^m}$
$F^1 = [\underline{f}^1, \overline{f}^1]$
$F^2 = [\underline{f}^2, \overline{f}^2]$
$F^M = [\underline{f}^M, \overline{f}^M]$
R1
R2
RM
$w_{LV_0}^1 + \sum_{i=1}^n w_{LV_i}^1 x_i$
$w_{AV_0}^1 + \sum_{i=1}^n w_{AV_i}^1 x_i$
$w_{LV_0}^2 + \sum_{i=1}^n w_{LV_i}^2 x_i$
$w_{AV_0}^2 + \sum_{i=1}^n w_{AV_i}^2 x_i$
$w_{LV_0}^M + \sum_{i=1}^n w_{LV_i}^M x_i$
$w_{AV_0}^M + \sum_{i=1}^n w_{AV_i}^M x_i$
y_{LV}
y_{AV}
Layer1
Layer2
Layer3
Layer4
Layer5

FIGURE 2. Type 2 FNC architecture.

II. MATERIALS AND METHODS

In this study, a type 2 FNC based on BDGPSO was developed for the navigation control of Ackermann steering vehicles. Fig. 1 depicts the architecture of the navigation control system. The distances (L_0, L_1, L_2, and L_3) of the returned obstacle or wall are obtained from lidar and are the input of the type-2 FNC; the two outputs are AV and LV. The proposed BDGPSO algorithm is then used to determine the optimal tunable parameters of the type 2 FNC.

A. TYPE 2 FNC

As shown in Fig. 2, the type 2 FNC has a five-layer architecture. The first to fifth layers are the input, fuzzified, rule, order reduction process, and output layers, respectively. In the following equation showing the fuzzy rules, If-Then is used as a representation:

$$\begin{aligned} &R_j: \text{ IF } x_1 \text{ is } \tilde{A}_1^j \text{ and } x_2 is \tilde{A}_2^j \ldots \text{ and } x_n is \tilde{A}_n^j \\ &\text{THEN } y_{LV} \text{ is } w_{LV_0}^j + \sum\nolimits_{i=1}^n w_{LV_i}^j x_i, \\ &y_{AV} \text{ is } w_{AV_0}^j + \sum\nolimits_{i=1}^n w_{AV_i}^j x_i \end{aligned}$$

where $x_1, x_2, \ldots, x_n$ are the inputs; $\tilde{A}_1^j,, \tilde{A}_2^j, \ldots, \tilde{A}_n^j$ are type 2 fuzzy sets; and y_{AV} and y_{LV} are the AV and LV of the consequent output of the TSK-type fuzzy rules.

The layers of the type 2 FNC architecture perform the following functions:

Layer 1: This layer transmits the lidar sensing distance value to the following layer without any calculation.

Layer 2: The output of Layer 1 is fuzzified and used as the input of this layer. As shown in Fig. 3, each node in this layer is defined as a type 2 fuzzy set. Type 2 fuzzy sets use a Gaussian primary membership function with standard deviation σ and mean value $[m_1, m_2]$. The Gaussian primary membership function $u_{\tilde{A}}$ is as follows:

$$u_{\tilde{A}_i^j} = \exp\left(-\frac{1}{2}\frac{\left[x_i - m_i^j\right]^2}{\left(\sigma_i^j\right)^2}\right) \equiv N\left(m_i^j, \sigma_i^j; x_i\right), \quad m_i^j \in \left[m_{i1}^j, m_{i2}^j\right]. \tag{1}$$

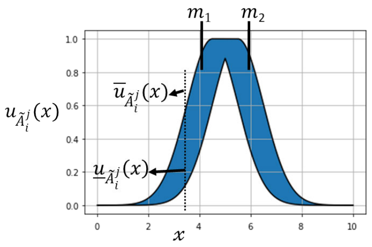

FIGURE 3. Schematic of a type 2 fuzzy membership function.

The membership degree of $u_{\tilde{A}}$ is called the uncertainty footprint and can be expressed as the membership functions $\bar{u}_{\tilde{A}}$ and $\underline{u}_{\tilde{A}}$ of the upper and lower bounds. These membership functions are defined as follows:

$$\bar{u}_{\tilde{A}_i^j} = \begin{cases} N\left(m_{i1}^j, \sigma_i^j; x_i\right), & x_i < m_{i1}^j \\ 1, & m_{i1}^j \le x_i \le m_{i2}^j \\ N\left(m_{i2}^j, \sigma_i^j; x_i\right), & x_i > m_{i2}^j \end{cases} \tag{2}$$

$$\underline{u}_{\tilde{A}_i^j} = \begin{cases} N\left(m_{i2}^j, \sigma_i^j; x_i\right), & x_i \le \frac{m_{i1}^j + m_{i2}^j}{2} \\ N\left(m_{i1}^j, \sigma_i^j; x_i\right), & x_i > \frac{m_{i1}^j + m_{i2}^j}{2} \end{cases} \tag{3}$$

The output nodes formed by the two boundaries are represented as the set $[\bar{u}_{\tilde{A}_i^j}, \underline{u}_{\tilde{A}_i^j}]$.

Layer 3: This layer is called the rule layer. Each node in this layer represents a fuzzy rule. These fuzzy rules often employ product operations to achieve the firing strength. The firing strength F^j of each node is as follows:

$$F^j = [\underline{f}^j, \bar{f}^j] \tag{4}$$

$$\bar{f}^j = \prod_i^n \bar{u}_{\tilde{A}_i^j} \text{ and } \underline{f}^j = \prod_i^n \underline{u}_{\tilde{A}_i^j} \tag{5}$$

Layer 4: In this layer, an order reduction operation is used to realize the complex operation of the type 2 fuzzy set and obtain the upper and lower bounds y_{i_upper} and y_{i_lower} of the output. To reduce the operation complexity in the reduction process, the center-of-sets reduction method is used as follows:

$$y_{i\_upper} = \frac{\sum_{j=1}^{M} \bar{f}^{j} \left(w_0^j + \sum_{i=1}^{n} w_i^j x_i \right)}{\sum_{j=1}^{M} \bar{f}^{j}} \tag{6}$$

$$y_{i\_lower} = \frac{\sum_{j=1}^{M} \underline{f}^{j} \left(w_0^j + \sum_{i=1}^{n} w_i^j x_i \right)}{\sum_{j=1}^{M} \underline{f}^{j}} \tag{7}$$

Layer 5: In this layer, the average method is used to calculate the output of Layer 4 and obtain a crisp output value $\mathrm{y_i}$. Here, the output contains two outputs of AV (y_{AV}) and LV (y_{LV}):

$$y_i = \frac{y_{i\_upper} + y_{i\_lower}}{2} \tag{8}$$

B. PROPOSED BDGPSO ALGORITHM

Bayesian parameter optimization was proposed by Pelikan [18]. Under known sample conditions, Bayesian optimization determines the optimal solution of a function by forming the posterior probability of the output of the objective function f(x). This optimal solution is the hyperparameter of the algorithm, which in this study consists of an inertia weight and two learning factors. Here, f(x) is represented as follows:

$$x^* = \mathrm{argmax} f(x) \tag{9}$$

The PSO algorithm uses the information shared by the members of a group to realize the sharing mechanism, and it then determines the optimal solution. We propose a BDGPSO algorithm to address the shortcomings of the traditional PSO algorithm. The steps used are as follows.

Step 1: Coding

The parameters of a type 2 FNC include the mean value $[m_{i1}^j, m_{i2}^j]$, standard deviation σ_i^j, and weights w_0^j and w_i^j of the posterior identification part. Each particle is composed of these parameters.

Step 2: Calculating the fitness value

The fitness values of all particles are calculated and sorted from high to low, and the group numbers in all particles are initialized to 0.

Step 3: Determining the similarity threshold of the fitness value

The particle with the highest fitness value currently sorted is determined, and this particle is set as the leader of the new group. This leader is called g. The initial value of g is then set to 1. With the group leader at the center, the average fitness difference and average distance difference between the corresponding pairs are calculated as follows:

$$F^g = \sum_{i=1}^{n} \left| Fit\left(L^g\right) - Fit\left(P^i\right) \right| \tag{10}$$

$$D^g = \sum_{i=1}^{n} \sum_{j=1}^{D} \sqrt{\left(L_j^g - P_j^i\right)^2} \tag{11}$$

$$AverageFitness\left(AF^g\right) = \frac{F^g}{GNC} \tag{12}$$

$$AverageDistance\left(AD^g\right) = \frac{D^g}{GNC} \tag{13}$$

where AF^g represents the fitness threshold of the gth group, L_j^g represents the jth position of the gth group, $Fit\ (Leader^g)$ represents the fitness value of the leader of the gth group, AD^g represents the distance threshold of the gth group, D represents the encoding dimension, n is the total number of particles, and GNC represents the number of particles with the zeroth group in the current solution space.

Step 4: Grouping

Ungrouped particles are grouped in order by using the following two formulas. The fitness difference between the current particle and leader particle is calculated as follows:

$$Fit^i = \left| Fit\left(L^g\right) - Fit\left(P^i\right) \right| \tag{14}$$

The distance difference between the current particle and leader particle is calculated as follows:

$$Dis^i = \sum_{j=1}^{D} \sqrt{\left(L_j^g - P_j^i\right)^2} \tag{15}$$

If the conditions $Fit^i < AF^g$ and $Dis^i < AD^g$ are met, the current particle and group leader have similar characteristics. Therefore, the current particle and group leader are assigned to the same group, and the group number of the current particle is updated to g. If the conditions $Fit^i < AF^g$ and $Dis^i < AD^g$ are not met, the current particle does not belong to this group. Therefore, the group number of the current particle is not updated.

Step 5: Determining whether the current particle will end the grouping step

If ungrouped particles exist among all particles, the process returns to Step 2, and the particle with the highest fitness value among the ungrouped particles is set as a new group. Steps 3 and 4 are then repeated. If all particles are grouped, the grouping step is ended.

Step 6: Updating the particle positions

The proposed BDGPSO algorithm does not refer to its own optimal position but rather to the leader particle L_g in its own group. The following are the updating formulas for position and velocity:

$$\begin{aligned} v_i(t+1) = \omega V_i(n) + C_1 r_1 \times \left(P_{L_g} - X_i(t)\right) \\ + C_2 r_2 \left(P_G - X_i(t)\right) \end{aligned} \tag{16}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{17}$$

Step 7: Mutual local search algorithm based on simulated annealing

The traditional PSO algorithm converges too quickly and easily falls into local optima. To solve this problem, we use a mutual local search algorithm [19] with the following formula:

$$P_j(t+1) = P_j(t) + r_2(-1,1) \times (P_G - Mutual_{vector} \times BF_2) \quad (18)$$

$$P_k(t+1) = P_k(t) + r_1(-1,1) \times (P_G - Mutual_{vector} \times BF_1) \quad (19)$$

$$Mutual_{vector} = \frac{P_j + P_k}{2} \quad (20)$$

where P_j is the jth particle in the swarm, P_G is the particle in the current best position, BF_1 and BF_2 are the benefit factors and are set to 1, P_k is the kth particle randomly selected in the swarm, and $r_1(-1,1)$ and $r_2(-1,1)$ are random real values within $[-1, 1]$. To find a more-optimal solution, simulated annealing methods start at high temperature. With the continuous decrease in temperature parameters, the cooling process is accomplished.

The difference between the current fitness value $Fit\left(P^i\right)$ and the two newly generated particles $Fit\left(P_j\right)$ and $Fit(P_k)$ is then estimated. The following formula is used to determine whether the current local optimum should be escaped:

$$e^{((\mathrm{Fit}(P_j)-\mathrm{Fit}(P^i))/\mathrm{T})} > \mathrm{r} \quad (21)$$

$$e^{((\mathrm{Fit}(P_k)-\mathrm{Fit}(P^i))/\mathrm{T})} > \mathrm{r} \quad (22)$$

where T is the temperature coefficient and r is a random number between -1 and 1. If either Eq. (21) or Eq. (22) is satisfied, the current solution is generated using the mutual local search algorithm displayed in Eq. (18). Otherwise, the current solution is randomly regenerated using Eq. (16).

Step 8: Determining whether the conditions for terminating the learning step have been met.

Steps 2–7 are repeated until the termination learning condition is met.

C. FITNESS FUNCTION

In this study, the fitness function is designed to evaluate the performance of the proposed controller. The designed fitness function Φ consists of four sub-fitness parts, namely $\varphi_1, \varphi_2, \varphi_3$, and φ_4. The fitness function φ_1 is primarily used to evaluate the actual movement distance of the Ackerman autonomous vehicle. When the movement distance D_m approaches the preset value D_p (330 m), it indicates that the Ackerman autonomous vehicle has completed an entire loop of map traversal in terms of the actual distance covered. The definition is as follows:

$$\varphi_1 = 1 - \frac{D_m}{D_p} \quad (23)$$

The sub-fitness φ_2 is used to estimate the average distance between the Ackerman autonomous vehicle and obstacles during movement. Herein, L_0 is employed to sense obstacles, and the distance from obstacles is set to 1.5 meters. The definition is as follows:

$$\varphi_2 = \frac{\sum_{t=1}^{T} |L_0(t) - 1.5|}{T} \quad (24)$$

where T present the total timestep. The sub-fitness φ_3 is to calculate the angle (θ) between the Ackermann autonomous vehicle and obstacles. θ represents the angle between L_0 and L_1 in the LiDAR sensor, and if the angle is 90 degrees, it indicates that the vehicle and the obstacle are parallel. The definition is as follows:

$$\varphi_3 = \frac{\sum_{t=1}^{T} |\theta(t) - 90|}{T} \quad (25)$$

The sub-fitness φ_4 is adopted to evaluate the obstacle avoidance distance of the Ackermann autonomous vehicle. Due to the larger turning radius of the Ackermann architecture, an appropriate obstacle avoidance distance is required when the vehicle detects obstacles. Sensing of the obstacle avoidance distance is achieved through the L_3 sensor in the LiDAR sensor, with the maximum detectable distance of the LiDAR set to 8 meters to ensure the vehicle has sufficient distance for obstacle avoidance. The definition is as follows:

$$\varphi_4 = \frac{\sum_{t=1}^{T} 8 - L_0(t)}{\sum_{t=1}^{T} 8} \quad (26)$$

Finally, the fitness function can be obtained by combining the four sub-fitness as follows:

$$\Phi = \frac{1}{1 + (\varphi_1 + \varphi_2 + \varphi_3 + \varphi_4)} \quad (27)$$

D. STABILITY ANALYSIS

To ensure the convergence and stability of the proposed Type 2 FNC, we employ the Lyapunov function approach to describe the stability of the dynamic system. The convergence analysis conducted in this section aims to ensure the convergence of the output error. It is important to note that while the guaranteed convergence of the output error is achieved, it does not imply that the parameters converge to optimal values. Detailed convergence derivation and stability analysis are presented in Appendices A and B.

E. ACKERMANN STEERING VEHICLE SYSTEM

The Ackerman steering geometry employed in this study was a self-developed vehicle (Fig. 4). The Ackerman steering geometry was characterized by two critical reference values. The first value was the LV, which is measured in meters per second and represents how fast the vehicle moves in a particular direction. The second value was the AV (Ω), which is measured in radians per second and represents the radians turned by the vehicle within unit time.

During the training process, to avoid Ackermann driverless vehicles colliding, moving away from walls, and violating the rules of motion. Herein, several constraints are designed to terminate the training, including: 1. The distance detected by any lidar signal is less than 0.5 meters, meaning the vehicle collides with a wall or an obstacle. 2. If the horizontal sensing

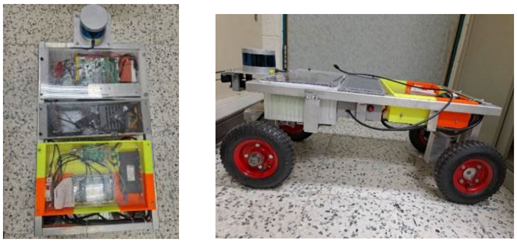

FIGURE 4. Self-developed Ackerman steering vehicle system.

distance of the LiDAR is greater than 5 meters, the vehicle is far from the wall. 3. The LV and AV of the vehicle exceed the set range (5 m/s and 0.7 rad/s) 4. The moving distance of the vehicle is greater than the length of the training field, meaning that the vehicle has completed a circle around the field.

III. EXPERIMENTAL RESULTS

To verify the practicability of the type 2 FNC based on BDGPSO, the fitness value and distance of the training environment of obstacle avoidance control and wall-following control were analyzed (Fig. 5). The designed type 2 FNC was also trained with different algorithms to illustrate the performance.

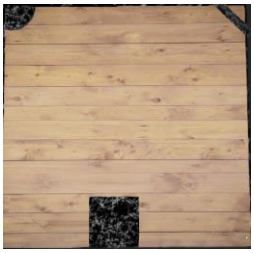

FIGURE 5. Training environment.

A. OPTIMIZING THE PSO PARAMETER RESULTS BY USING THE BAYESIAN METHOD

During the Bayesian optimization experiment on the PSO parameters, the range setting of the various parameters was based on [17]. The range of PSO parameters is set as follows: for the inertia weight ω, [0.3, 0.9]; for the cognitive coefficient c_1, [1.5, 2.5]; and for the social coefficient c_2, [1.5, 2.5]. In this process, ω, c_1, and c_2 will be automatically adjusted by Bayesian optimization. After 10 experiments, different optimized parameter combinations can be obtained. As shown in Table 1, the experimental results revealed the performance when using various inertia weights and learning factors and the Bayesian method. The parameters $\omega = 0.3728$, $c_1 = 1.636$, and $c_2 = 1.56$ were discovered to result in the highest performance.

TABLE 1. Performance for various inertia weights and learning factors when using the bayesian method.

c_1	c_2	ω	Fitness value L_3
1.746	2.36	0.6233	0.9212
2.053	2.342	0.3745	0.835905
1.779	2.086	0.8818	0.826441
2.061	1.519	0.7804	0.867234
1.733	2.307	0.5327	0.976887
2.364	2.247	0.6337	0.937347
1.636	1.56	0.3728	0.983756
1.545	1.607	0.4354	0.78073
2.213	2.06	0.3075	0.825758

B. EXPERIMENTAL RESULTS OF A TYPE 2 FNC IN THE TRAINING ENVIRONMENT

To clearly demonstrate the output results of the proposed type 2 FNC, we illustrated angle changes with various colors. For example, we used red to indicate a vehicle moving in a straight line, blue to indicate a vehicle turning left, and green to indicate a vehicle turning right. The color definitions are as follows:

$$\begin{cases} red, & if -0.1 \leq y_{AV} \leq 0.1 \\ blue, & if -0.1 < y_{AV} \leq -0.5 \\ green, & if 0.1 < y_{AV} \leq 0.5 \end{cases} \tag{28}$$

In terms of computational complexity, both models use 4 inputs and 6 fuzzy rules, with a total parameter count of 233 and 281, respectively. The key factor influencing training time is the complexity of different algorithms. During training, each algorithm undergoes 1000 iterations. To ensure fair training conditions, the initial parameters of the models are fixed. Therefore, each algorithm is trained under the same conditions. In the training environment, two experiments were conducted: an experiment with type 2 FNC output with AV only and an experiment with type 2 FNC output with both AV and LV. Table 2 lists the experimental results of wall-following control when a type 2 FNC was used with BDGPSO. From Table 2, it can be observed that the DE algorithm has the fastest training time in type 2 FNC output with AV output, followed by QPSO and BDGPSO. In type 2 FNC output with AV and LV outputs, BDGPSO and QPSO exhibit similar training times, but BDGPSO achieves a better fitness value. The fitness values of the type 2 FNC output with AV output only and with both AV and LV outputs were 0.980572 and 0.985946, respectively, and were higher than those for other methods. On the other hand, the fitness value of the GAPSO method is the lowest among all methods. This means that the GAPSO method is more likely to fall into the local optimal solution than other methods.

Figs. 6 and 7 depict the movement trajectories in wall-following control when using the type 2 FNC with AV output only and with both AV and LV outputs, respectively, for various algorithms. The moving trajectory of the proposed type 2 FNC with BDGPSO was more stable than those of other algorithms. In addition, in the trajectory involving avoidance of square obstacles, the radian curve of the proposed type 2 FNC with BDGPSO was smoother than those for other methods.

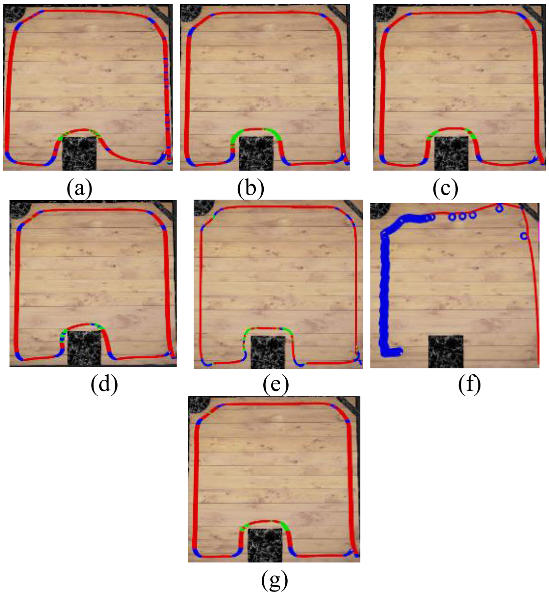

FIGURE 6. Movement trajectories of a type 2 FNC with AV output when using the (a) DE, (b) DGPSO, (c) QPSO, (d) ABC, (e) ESe-PSO, (f) GAPSO, and (g) BDGPSO algorithms.

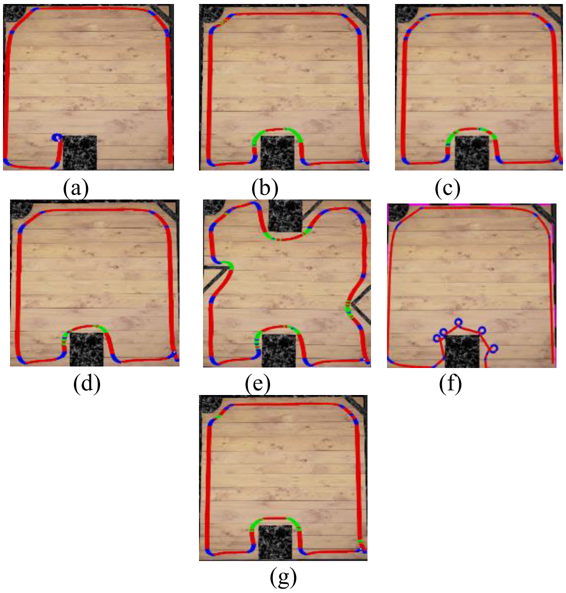

FIGURE 7. Movement trajectories with a type 2 FNC with both AV and LV outputs when using the (a) DE, (b) DGPSO, (c) QPSO, (d) ABC, (e) ESe-PSO, (f) GAPSO, and (g) BDGPSO algorithms.

TABLE 2. Performance of various algorithms in the training environment.

Environment / Evaluation / Methods	Training environment					
	Output with AV (Fig. 6)			Output with AV and LV (Fig. 7)		
	F	D (m)	T (hr)	F	D (m)	T (hr)
DGPSO[20]	0.975673	332.026	104	0.982351	332.0214	132
QPSO[23]	0.972339	332.0215	83	0.985743	332.0211	107
ABC[21]	0.974572	332.0214	92	0.978166	332.0212	114
DE[22]	0.930307	332.0211	77	0.883606	219.1	113
ESe-PSO[24]	0.9774	332.22	90	0.9662	332.2	117
GAPSO[25]	0.6902	215.46	100	0.8797	276.23	123
BDGPSO	0.980572	332.0214	84	0.985946	332.0212	108

*The optimal trajectory in the training environment is of length 332 m. F = Fitness value, D = Distance, T = Training time.

TABLE 3. Performance comparison of different algorithms in testing environment.

Environment / Evaluation / Methods	Testing environment					
	Output with AV (Fig. 8)			Output with AV and LV (Fig. 9)		
	F	D (m)	T (sec)	F	D (m)	T (sec)
DGPSO[20]	0.958333	388.827	12.07	0.96473	388.8271	40.97
QPSO[23]	0.9549	388.826	11.64	0.9595	388.824	25.274
ABC[21]	0.9524	388.824	11.91	0.9543	388.8278	32.53
DE[22]	0.9053	388.825	13.38	fail	fail	fail
ESe-PSO[24]	0.9543	388.822	12.15	0.9558	388.825	19.79
GAPSO[25]	fail	fail	fail	fail	fail	fail
BDGPSO	0.9648	388.824	10.94	0.97351	388.821	25.555

*The optimal trajectory in the training environment is of length 388.8 m. F =

C. EXPERIMENTAL RESULTS OF A TYPE 2 FNC IN THE TESTING ENVIRONMENT

In the testing environment, two experiments were conducted. Table 3 lists the experimental results of wall-following control when a type 2 FNC with BDGPSO was used. The fitness values of the type 2 FNC with AV output only and with both AV and LV outputs were 0.964876 and 0.973510, respectively, and were higher than those for other methods. In addition, the durations required to move in a circle around the testing environment using the type 2 FNC output with AV output only and with both AV and LV outputs were 10.944 and 25.555 s, respectively, and were shorter than those of other methods. Figs. 8 and 9 depict the movement trajectories of wall-following control using the type 2 FNC with AV output only and with both AV and LV outputs in various algorithms. As illustrated in Fig. 9, the type 2 FNC with both AV and LV outputs using the DE algorithm failed in the testing environment. According to Figs. 8 and 9, the moving trajectory of the proposed type 2 FNC with BDGPSO was smoother than those of other algorithms.

D. NAVIGATION CONTROL RESULTS OF A TYPE 2 FNC

Navigation control is divided into two parts: control for moving along a wall (obstacle avoidance) and control for moving toward a target. As shown in Fig. 10, the environment

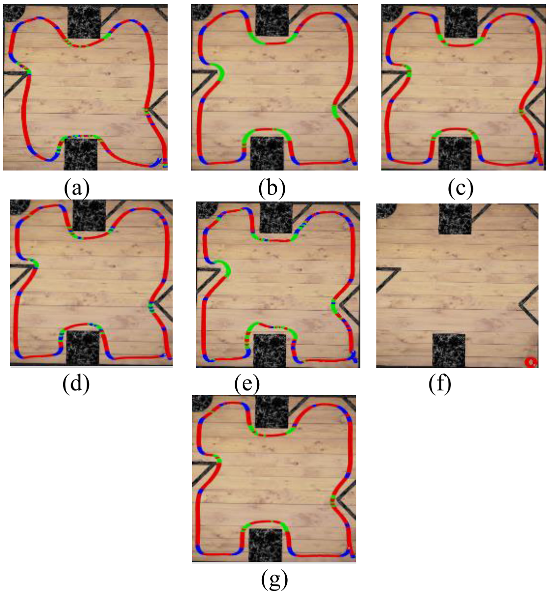
(a) (b) (c)
(d) (e) (f)
(g)

FIGURE 8. Movement trajectories of a type 2 FNC with AV output only when using the (a) DE, (b) DGPSO, (c) QPSO, (d) ABC, (e) ESe-PSO, (f) GAPSO, and (g) BDGPSO algorithms.

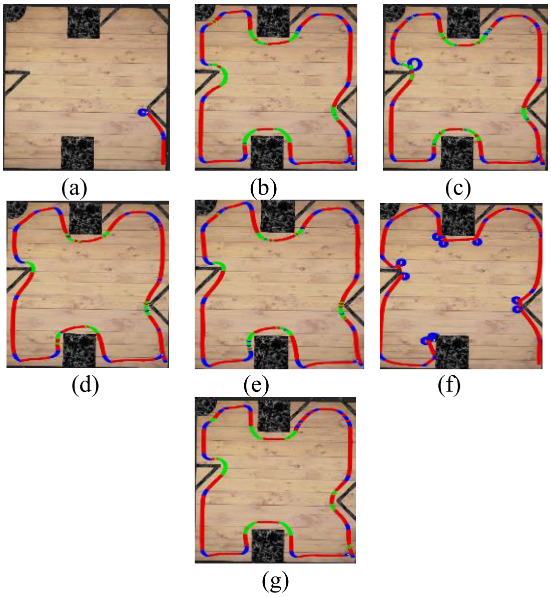
(a) (b) (c)
(d) (e) (f)
(g)

FIGURE 9. Moving trajectories using the type-2 FNC with AV and LV outputs in various algorithms. (a) DE, (b) DGPSO, (c) QPSO, (d) ABC, (e) ESe-PSO, (f) GAPSO, and (g) BDGPSO algorithms.

of a vehicle can be divided into four areas: A_1, A_2, A_3, and A_4. If the target direction falls within A_{1-3} and the area contains obstacles, the navigation control switches to control for moving along a wall (obstacle avoidance). If the target direction falls within A_{1-3} and the area contains no obstacles, the navigation control switches to control for moving toward a target. If the target direction falls within A_4, the navigation control switches to control for moving along a wall (obstacle avoidance) until the target direction falls within another area.

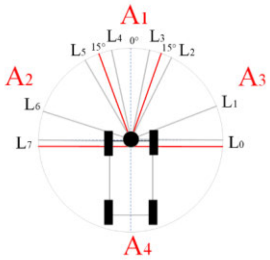

FIGURE 10. Four areas in the environment of a vehicle.

To verify the effectiveness of this navigation control method on different terrains, the proposed type 2 FNC with AV output only and with both AV and LV outputs was used to complete navigation control. Table 4 lists the navigation control results of the proposed type 2 FNC with AV output only and with both AV and LV outputs and compares the erformance of various algorithms. The vehicle movement distance and required time of the proposed type 2 FNC with AV output only were 152.03 m and 144 s, respectively, whereas those of the proposed type 2 FNC with both AV and LV outputs were 147.09 m and 720 s, respectively. The experimental results indicated that the vehicle navigation movement distance and required time of the proposed method were superior to those of other algorithms.

TABLE 4. Performance results for various algorithms.

Environment / Evaluation / Methods	Output with AV (Fig. 11)		Output with AV and LV (Fig. 12)	
	D (m)	T (sec)	D (m)	T (sec)
DGPSO[20]	157.93	152	163.2902	1499
QPSO[23]	155.12	160	158.8112	1459
ABC[21]	159.39	153	159.392	753
DE[22]	166.59	159	fail	fail
ESe-PSO[24]	217.86	162	199.599	1048
GAPSO[25]	fail	fail	fail	fail
BDGPSO	152.03	144	147.0904	720

*D = Distance ,T = Time of a circle around the environment.

IV. SELF-DEVELOPED ACKERMANN STEERING VEHICLE TESTED IN AN ACTUAL UNKNOWN ENVIRONMENT

Fig. 13 presents a schematic of an actual unknown environment; the blue squares indicate iron frames with built-in glass in the wall. The vehicle starts at Zone A, travels 1 m along the wall, and completes a full circle to Zone D. For actual environment wall-following control, two experiments were conducted: an experiment with type 2 FNC output with AV only and an experiment with type 2 FNC output with both AV and LV.

The first experiment was conducted using the proposed type 2 FNC with AV output only. The LV was fixed at 1 m/s, and the duration required to move in a circle around the environment was 39 s. The actual movement of the vehicle along Zones A, B, C, and D is depicted in Fig. 14, and the distance

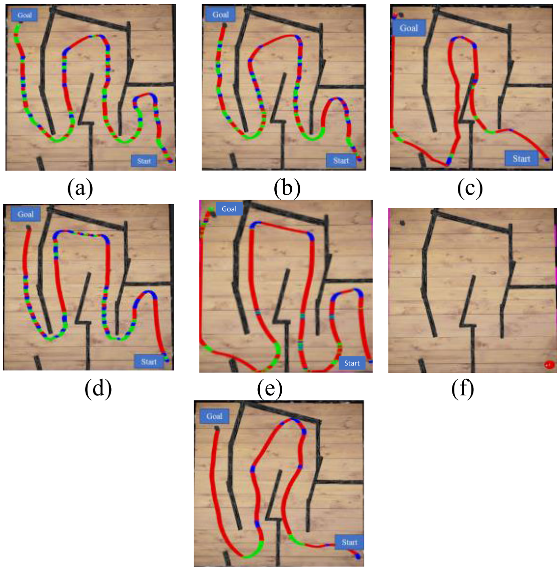

FIGURE 11. Navigation control movement trajectories of a type 2 FNC with AV output only when using the (a) DE, (b) DGPSO, (c) QPSO, (d) ABC, (e) ESe-PSO, (f) GAPSO, and (g) BDGPSO algorithms.

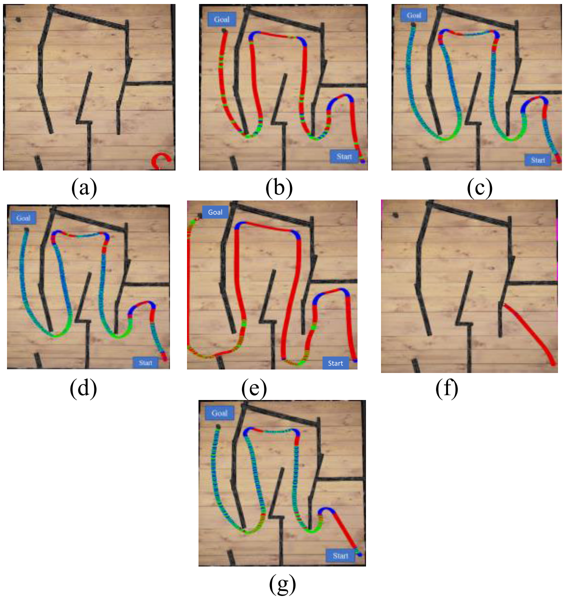

FIGURE 12. Navigation control movement trajectories of a type 2 FNC with both AV and LV outputs when using the (a) DE, (b) DGPSO, (c) QPSO, (d) ABC, (e) ESe-PSO, (f) GAPSO, and (g) BDGPSO algorithms.

between the vehicle and the wall in the wall-following control scenario is depicted in Fig. 15(a). In Zones A and C, the vehicle moved stably along the wall. Zone B is depicted in Fig. 13 as a concave terrain. Because the Ackerman steering vehicle had a large radius of gyration, the distance obtained was relatively large. As shown in Fig. 13, Zone D had iron frames with built-in glass in the wall. This glass was penetrated by lidar, resulting in long-distance return values. However, the iron frames returned the lidar-sensed distance signal. Therefore, the vehicle did not attempt to move through the glass and continued to move along the wall, eventually reaching the end point.

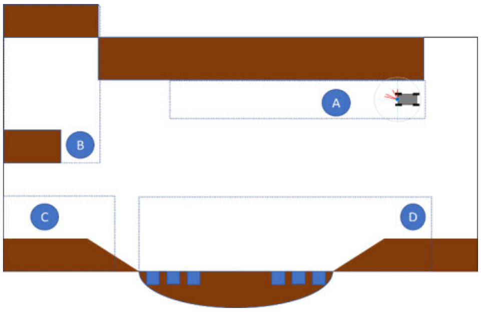

FIGURE 13. Schematic of an actual unknown environment.

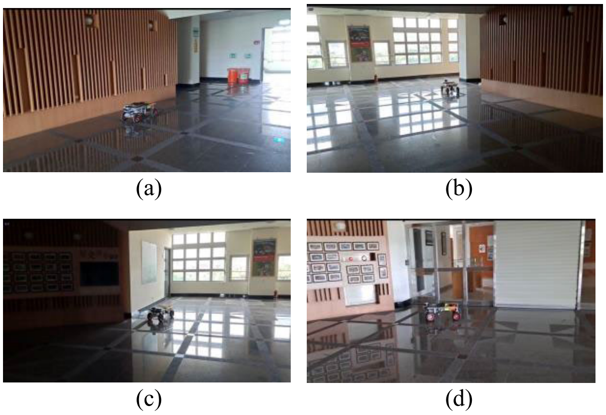

FIGURE 14. Actual vehicle movement in Zones (a) A, (b) B, (c) C, and (d) D.

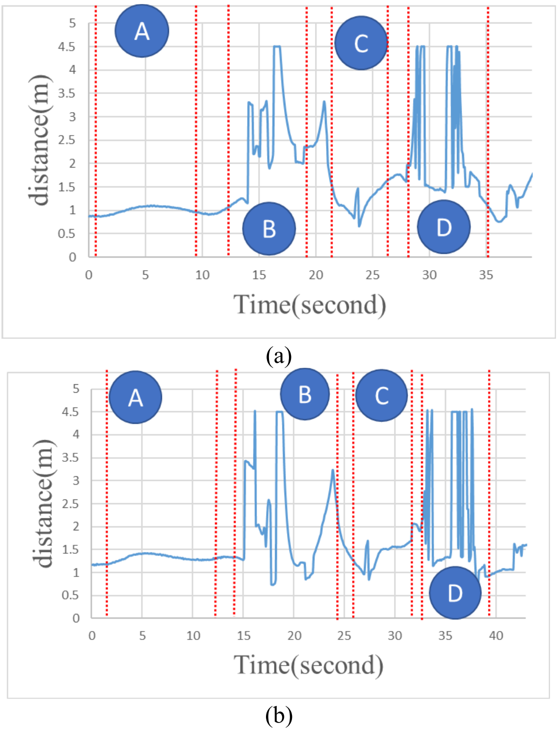

FIGURE 15. Distance between the vehicle and wall in the wall-following control scenario: (a) AV output only and (b) AV and LV outputs.

The second experiment was conducted using the proposed type 2 FNC with both AV and LV outputs. The duration required to move in a circle around the environment was 43 s, and the actual movement was similar to that displayed in Fig. 14. The distance between the vehicle and the wall in the wall-following control scenario is depicted in Fig. 15(b). At Zone B, the LV output of the controller slowed the vehicle, resulting in an increase in the time needed to reach Zone B. In conclusion, both type 2 FNCs, that with AV output only and that with both AV and LV outputs, can move in a complete circle around unknown environments and can be used in both simulated and actual environments.

V. CONCLUSION

In this study, we proposed a type 2 FNC based on BDGPSO for navigation control of an Ackermann steering vehicle. The following are the main contributions of this study:

1. A type 2 FNC was developed that has a five-layer network architecture consisting of an input layer, a fuzzified layer, a rule layer, an order reduction process layer, and an output layer.
2. The proposed BDGPSO algorithm prevents PSO from falling into local optima and uses the Bayesian algorithm to determine the optimal inertia weight and learning factor combination.
3. The new fitness function of the reinforcement-learning strategy based on the distance information returned by lidar can be used to evaluate type 2 FNCs.
4. According to the experimental results for wall-following control, the fitness values of the type 2 FNC with AV output only and with both AV and LV outputs were 0.964876 and 0.973510, respectively, and were higher than those of other methods. In addition, the durations required to move in a circle around the testing environment when using the type 2 FNC with AV output only and with both AV and LV outputs were 10.944 and 25.555 s, respectively, and were shorter than those of other methods.
5. Type 2 FNCs with AV output only and with both AV and LV outputs can be used to successfully complete navigation control of Ackerman steering vehicles in unknown environments.

In this study, the lidar sensing return distance was used to determine how a vehicle could be navigated and obstacles avoided. However, during the process of navigation control, we were unaware of the obstacles and whether they should be avoided. Therefore, for navigation control, we aim in future research to introduce imaging technology and use heterogeneous sensors (e.g., lidar and cameras) to monitor the surrounding environment in real time. In future work, the proposed method can be applied to the following applications such as: 1. Automation guided vehicle: Ackermann steering geometry keeps the vehicle stable during driving and is suitable for transporting goods in warehouses and factories. 2. agricultural machinery vehicle: unmanned vehicles make precise turns and maneuver through fields, ensuring efficient crop cultivation, harvesting, and spraying operations. 3. unmanned exploration vehicle: Ackermann steering vehicles can be employed as unmanned exploration vehicles in environments that are hazardous or inaccessible to humans, including exploring remote areas, rugged terrains, underwater environments, and extraterrestrial surfaces. By leveraging the precise control and maneuverability of Ackermann steering, these vehicles can navigate challenging landscapes, collect data, and conduct scientific research.

APPENDIX

A. PROOF OF THE UNIVERSAL APPROXIMATOR THEOREM

To demonstrate the universal approximator theorem in the type 2 FNC, we utilize the Stone-Weierstrass theorem [26] specifically for the multi-input single-output (MISO) function. The type 2 FNC is expressed as:

$$y_{(x)} = \frac{y_{upper} + y_{lower}}{2} \tag{A.1}$$

$$y_{upper} = \frac{\sum_{j=1}^{M} \bar{f}^j \left(w_0^j + \sum_{i=1}^{n} w_i^j x_i \right)}{\sum_{j=1}^{M} \bar{f}^j},$$
$$y_{lower} = \frac{\sum_{j=1}^{M} \underline{f}^j \left(w_0^j + \sum_{i=1}^{n} w_i^j x_i \right)}{\sum_{j=1}^{M} \underline{f}^j}) \tag{A.2}$$

Stone-Weierstrass Theorem: Let A denote a collection of continuous real-valued function defined on a compact set U. If the following conditions are satisfied, then A's uniform closure consists of all real continuous functions on U.

Condition 1: U is an algebra such that if $c \in \mathbb{R}$ and f_1, $f_2 \in A$ then $cf_1 \in A$, $f_1 + f_2 \in A$, and $f_1 \cdot f_2 \in A$.

Condition 2: A separates points on U, which implies that for any two distinct points in U, there exists a function in A such that the function values at those points are distinct as well. More formally, for any two points $x, y \in U$ where $x \neq y$, there exists a function $f_1 \in A$ such that $f_1(x) \neq f_2(y)$.

Condition 3: A vanishes at no point of U, meaning that for every point in U, there exists a function in A such that the function value at that point is nonzero. More formally, for every point $x \in U$, there exists a function $f_1 \in A$ such that $f_1(x) \neq 0$.

Lemma A1: U is a compact set. Let Y denote the set of function $y{:}\mathfrak{R}^N \rightarrow \mathfrak{R}$ defined in Equation (A.1); then Y → U.

Proof of Lemma A1: The membership function is defined as:

$$0 < u_{\tilde{A}_i^j} = \exp\left(-\frac{1}{2} \frac{\left[x_i - m_i^j \right]^2}{\left(\sigma_i^j \right)^2} \right) \leq 1 \tag{A.3}$$

The membership function $u_{\tilde{A}_i^j}$ is closed and bounded for all $x \in \mathfrak{R}^N$. That is, $\mathrm{Y} \subset U$.

Proof of Stone-Weierstrass Theorem: Let $f_1, f_2 \in Y$, the function can be written as:

$$f_1(x) = \frac{1}{2}\left(\frac{\sum_{j1=1}^{M1}\overline{f1}^j\left(w1_0^j+\sum_{i=1}^{n}w1_i^j x1_i\right)}{\sum_{j=1}^{M}\overline{f1}^j} + \frac{\sum_{j1=1}^{M1}\underline{f1}^j\left(w1_0^j+\sum_{i=1}^{n}w1_i^j x1_i\right)}{\sum_{j=1}^{M}\underline{f1}^j}\right)$$
$$= \frac{1}{2}\left(\frac{\sum_{j1=1}^{M1}\overline{f1}^j T1_j}{\sum_{j=1}^{M}\overline{f1}^j} + \frac{\sum_{j1=1}^{M1}\underline{f1}^j T1_j}{\sum_{j=1}^{M}\underline{f1}^j}\right) \tag{A.4}$$

$$f_2(x) = \frac{1}{2}\left(\frac{\sum_{j2=1}^{M2}\overline{f2}^j\left(w2_0^j+\sum_{i=1}^{n}w2_i^j x2_i\right)}{\sum_{j=1}^{M}\overline{f2}^j} + \frac{\sum_{j2=1}^{M2}\underline{f2}^j\left(w2_0^j+\sum_{i=1}^{n}w2_i^j x2_i\right)}{\sum_{j=1}^{M}\underline{f2}^j}\right)$$
$$= \frac{1}{2}\left(\frac{\sum_{j2=1}^{M2}\overline{f2}^j T2_j}{\sum_{j=1}^{M}\overline{f2}^j} + \frac{\sum_{j2=1}^{M2}\underline{f2}^j T2_j}{\sum_{j=1}^{M}\underline{f2}^j}\right) \tag{A.5}$$

Therefore,

$$f_1 + f_2(x) = \frac{1}{2}\left(\frac{\sum_{j1=1}^{M1}\sum_{j2=1}^{M2}\left(T1_j+T2_j\right)\cdot\left(\overline{f1}^j\overline{f2}^j\right)}{\sum_{j1=1}^{M1}\sum_{j2=1}^{M2}\left(\overline{f1}^j\overline{f2}^j\right)} + \frac{\sum_{j1=1}^{M1}\sum_{j2=1}^{M2}\left(T1_j+T2_j\right)\cdot\left(\underline{f1}^j\underline{f2}^j\right)}{\sum_{j1=1}^{M1}\sum_{j2=1}^{M2}\left(\underline{f1}^j\underline{f2}^j\right)}\right) \tag{A.6}$$

The Gaussian form of $\overline{f}$ and $\underline{f}$ can be easily verified through algebraic operations. As a result, Equation (A.6) has the same form as Equation (A.1), which implies that $f_1 + f_2 \in Y$. Similarly, we have

$$f_1 \cdot f_2(x) = \frac{1}{2}\left(\frac{\sum_{j1=1}^{M1}\sum_{j2=1}^{M2}\left(T1_j\cdot T2_j\right)\cdot\left(\overline{f1}^j\overline{f2}^j\right)}{\sum_{j1=1}^{M1}\sum_{j2=1}^{M2}\left(\overline{f1}^j\overline{f2}^j\right)} + \frac{\sum_{j1=1}^{M1}\sum_{j2=1}^{M2}\left(T1_j\cdot T2_j\right)\cdot\left(\underline{f1}^j\underline{f2}^j\right)}{\sum_{j1=1}^{M1}\sum_{j2=1}^{M2}\left(\underline{f1}^j\underline{f2}^j\right)}\right) \tag{A.7}$$

which is worth noting that Equation (A.7) has the same form as Equation (A.1); hence, $f_1 \cdot f_2 \in Y$. Finally, for arbitrary $c \in R$

$$c \cdot f_1(x) = \frac{1}{2}\left(\frac{\sum_{j1=1}^{M1}\left(c\cdot T1_j\right)\cdot\overline{f1}^j}{\sum_{j1=1}^{M1}\overline{f1}^j} + \frac{\sum_{j1=1}^{M1}\left(c\cdot T1_j\right)\cdot\underline{f1}^j}{\sum_{j1=1}^{M1}\underline{f1}^j}\right) \tag{A.8}$$

This further confirms that Equation (A.8) has the form of Equation (A.1), which implies that $cf_1 \in Y$. Consequently, Y is an algebra.

To demonstrate that Y separates points on U, we construct a function f that satisfies the necessary conditions. For arbitrarily given $\alpha, \beta \in U$ with $\alpha \neq \beta$, the $f \in Y$ is specified such that $f(\alpha) \neq f(\beta)$. In this demonstration, we select two fuzzy rules $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_i)$ and $\beta = (\beta_1, \beta_2, \cdots, \beta_i)$ for the fuzzy rule base based on Equation 1. The Gaussian membership functions is present as:

$$u_{\tilde{A}_1} = \exp\left(-\frac{1}{2}\frac{[x_i-\alpha_i]^2}{(\sigma_i)^2}\right) \tag{A.9}$$
$$u_{\tilde{A}_2} = \exp\left(-\frac{1}{2}\frac{[x_i-\beta_i]^2}{(\sigma_i)^2}\right) \tag{A.10}$$

Then, f can be expressed as

$$f = \frac{T1\cdot\prod_i^n \exp\left(-\frac{1}{2}\frac{[x_i-\alpha_i]^2}{(\sigma_{i1})^2}\right)+T2\cdot\prod_i^n \exp\left(-\frac{1}{2}\frac{[x_i-\beta_i]^2}{(\sigma_{i2})^2}\right)}{\prod_i^n \exp\left(-\frac{1}{2}\frac{[x_i-\alpha_i]^2}{(\sigma_{i1})^2}\right)+\prod_i^n \exp\left(-\frac{1}{2}\frac{[x_i-\beta_i]^2}{(\sigma_{i2})^2}\right)} \tag{A.11}$$

$$f(\alpha) = \frac{T1+T2\cdot\prod_i^n \exp\left(-\frac{1}{2}\frac{[\alpha_i-\beta_i]^2}{(\sigma_{i2})^2}\right)}{1+\prod_i^n \exp\left(-\frac{1}{2}\frac{[\alpha_i-\beta_i]^2}{(\sigma_{i2})^2}\right)} \tag{A.12}$$
$$f(\beta) = \frac{T2+T1\cdot\prod_i^n \exp\left(-\frac{1}{2}\frac{[\beta_i-\alpha_i]^2}{(\sigma_{i1})^2}\right)}{1+\prod_i^n \exp\left(-\frac{1}{2}\frac{[\beta_i-\alpha_i]^2}{(\sigma_{i1})^2}\right)} \tag{A.13}$$

As $\alpha \neq \beta$, we have $f(\alpha) \neq f(\beta)$. Consequently, we can conclude that Y separates points on U. Lastly, we can demonstrate that Y vanishes at no point of U. According to Equation (A.1), both $\overline{f}^j$ and $\underline{f}^j$ are constant and nonzero. If $\overline{f}^j$ and $\underline{f}^j$ are greater than zero, then y $\in Y$ can be chosen as the required function f.

B. SYSTEM STABILITY ANALYSIS

Assuming the type 2 FNC as an identifier, the objective function is defined as:

$$e(k) = y_d(k) - y_O(k) \tag{B.1}$$
$$V(k) = \frac{1}{2}e^2(k) \tag{B.2}$$

The change of Lyapunov function is obtained as

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}\left(e^2(k+1) - e^2(k)\right)$$
$$= \frac{1}{2}\left(e(k+1) - e(k)\right)\left(e(k+1) + e(k)\right)$$
$$= \frac{1}{2}\Delta e(k)\left(\Delta e(k) + 2e(k)\right) \tag{B.3}$$

The network parameters that need to be updated in Type 2 FNC include the mean value $M \in \left[m_{i1}^j, m_{i2}^j\right]$ and deviation σ of the Interval type-2 Gaussian membership and the consequent TSK-type parameter $T = w_0^j + \sum_{i=1}^{n} w_i^j x_i$, which are defined as follows:

$$M=\begin{bmatrix} m_{11}^{1} & \cdots & m_{i1}^{1} & m_{12}^{1} & \cdots & m_{i2}^{1} \\ & \cdots & & m_{12}^{j} & \cdots & m_{i2}^{j} \end{bmatrix} \tag{B.4}$$

$$\sigma=\left[\begin{array}{ccc} \sigma_{1}^{1} & \cdots & \sigma_{i}^{1} \\ \hline \cdots & \cdots & \sigma_{i}^{j} \end{array}\right] \tag{B.5}$$

$$T=\left[\begin{array}{ccc} w_{0}^{1} & \cdots & w_{i}^{1} \\ \hline \cdots & \cdots & w_{i}^{j} \end{array}\right] \tag{B.6}$$

Here we use Equation 16 to update the parameters of Type 2 FNC. Since the best particle experience is equivalent to the global best leader, we can proceed to examine the particle dynamics that are associated with the best leader particle, as follows:

$$v_i\left(k\right)=\omega v\left(k-1\right)+\delta\times\left(x_{L_g}-x\left(k\right)\right) \tag{B.7}$$

$$x_i\left(k\right)=x_i\left(k-1\right)+v_i\left(k\right) \tag{B.8}$$

The $\delta = C_1 r_1 + C_2 r_2$ The combined stochastic parameter is no longer uniformly distributed but satisfy

$$0<\delta<C_1+C_2 \tag{B.9}$$

Here, we use (B.7) and (B.8) for type 2 FNC, the change of error can be approximated by

$$\Delta e\left(k\right)=\Delta M\left(k\right)+\Delta\sigma\left(k\right)+\Delta T(k) \tag{B.10}$$

Before delving into the stability analysis of the Lyapunov function, let us first examine the updating rule of parameters that are utilized in the BDGPSO algorithm:

$$\begin{aligned} M\left(k+1\right)=M\left(k\right)+\omega^{M}\left(k\right)\left(M\left(k\right)\right. \\ \left. -M\left(k-1\right)\right)+\delta^{M}(k)(M_{L_g}-M\left(k\right)) \end{aligned} \tag{B.11}$$

$$\begin{aligned} \sigma\left(k+1\right)=\sigma\left(k\right)+\omega^{\sigma}\left(k\right)\left(\sigma\left(k\right)\right. \\ \left. -\sigma\left(k-1\right)\right)+\delta^{\sigma}(k)(\sigma_{L_g}-\sigma\left(k\right)) \end{aligned} \tag{B.12}$$

$$\begin{aligned} T\left(k+1\right)=T\left(k\right)+\omega^{T}\left(k\right)\left(T\left(k\right)\right. \\ \left. -T\left(k-1\right)\right)+\delta^{T}(k)(T_{L_g}-T\left(k\right)) \end{aligned} \tag{B.13}$$

From (B.11) to (B.13) we can write:

$$\begin{aligned} \Delta M\left(k\right)&=M\left(k+1\right)-M\left(k\right) \\ &=\omega^{M}\left(k\right)\left(M\left(k\right)-M\left(k-1\right)\right)+\delta^{M}(k)(M_{L_g}-M\left(k\right)) \end{aligned} \tag{B.14}$$

$$\begin{aligned} \Delta\sigma\left(k\right)&=\sigma\left(k+1\right)-\sigma\left(k\right) \\ &=\omega^{\sigma}\left(k\right)\left(\sigma\left(k\right)-\sigma\left(k-1\right)\right)+\delta^{\sigma}(k)(\sigma_{L_g}-\sigma\left(k\right)) \end{aligned} \tag{B.15}$$

$$\begin{aligned} \Delta T\left(k\right)&=T\left(k+1\right)-T\left(k\right) \\ &=\omega^{T}\left(k\right)\left(T\left(k\right)-T\left(k-1\right)\right)+\delta^{T}(k)(T_{L_g}-T\left(k\right)) \end{aligned} \tag{B.16}$$

From (B.3) and (B.10) the following equation is obtained:

$$\begin{aligned} \Delta V\left(k\right)=\frac{1}{2}(\Delta M\left(k\right)+\Delta\sigma\left(k\right)+\Delta T(k))(\Delta M\left(k\right) \\ +\Delta\sigma\left(k\right)+\Delta T(k)+2e\left(k\right)) \end{aligned} \tag{B.17}$$

To guarantee the stability, the change of Lyapunov function must be less than zero so we have:

$$\begin{aligned} \Delta V\left(k\right)<0\Longrightarrow\frac{1}{2}\left(\Delta M\left(k\right)+\Delta\sigma\left(k\right)+\Delta T\left(k\right)\right)\left(\Delta M\left(k\right)\right. \\ \left.+\Delta\sigma\left(k\right)+\Delta T\left(k\right)+2e\left(k\right)\right)<0 \end{aligned} \tag{B.18}$$

After performing several mathematical operations on the aforementioned equation, we derived the stability condition.

CONFLICTS OF INTEREST

The authors have no conflict of interest to declare.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable.

REFERENCES

[1] O. Liu, S. Yuan, and Z. Li, "A survey on sensor technologies for unmanned ground vehicles," in *Proc. 3rd Int. Conf. Unmanned Syst. (ICUS)*, Nov. 2020, pp. 638–645, doi: 10.1109/ICUS50048.2020.9274845.

[2] J. Lin, Y. Tan, and J. Tian, "Inter-frame correlation based on moving vehicle target detection in infrared image sequences," in *Proc. Int. Conf. Comput. Technol., Electron. Commun. (ICCTEC)*, Dec. 2017, pp. 1187–1191, doi: 10.1109/ICCTEC.2017.00258.

[3] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, "Deep learning for LiDAR point clouds in autonomous driving: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3412–3432, Aug. 2021, doi: 10.1109/TNNLS.2020.3015992.

[4] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should I walk? Predicting terrain properties from images via self-supervised learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1509–1516, Apr. 2019, doi: 10.1109/LRA.2019.2895390.

[5] G. Kahn, P. Abbeel, and S. Levine, "BADGR: An autonomous self-supervised learning-based navigation system," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1312–1319, Apr. 2021, doi: 10.1109/LRA.2021.3057023.

[6] H. M. Yudha, T. Dewi, N. Hasana, P. Risma, Y. Oktarini, and S. Kartini, "Performance comparison of fuzzy logic and neural network design for mobile robot navigation," in *Proc. Int. Conf. Electr. Eng. Comput. Sci. (ICECOS)*, Oct. 2019, pp. 79–84, doi: 10.1109/ICECOS47637.2019.8984577.

[7] J. Wang, Z. Ma, and X. Chen, "Generalized dynamic fuzzy NN model based on multiple fading factors SCKF and its application in integrated navigation," *IEEE Sensors J.*, vol. 21, no. 3, pp. 3680–3693, Feb. 2021, doi: 10.1109/JSEN.2020.3022934.

[8] M. Faisal, M. Algabri, B. M. Abdelkader, H. Dhahri, and M. M. Al Rahhal, "Human expertise in mobile robot navigation," *IEEE Access*, vol. 6, pp. 1694–1705, 2018, doi: 10.1109/ACCESS.2017.2780082.

[9] S. Sahloul, D. Benhalima, and C. Rekik, "Comparative study of hybrid fuzzy logic methods for mobile robot navigation in unknown environments," in *Proc. 19th Int. Conf. Sci. Techn. Autom. Control Comput. Eng. (STA)*, Mar. 2019, pp. 170–175, doi: 10.1109/STA.2019.8717260.

[10] Y. Saidi, M. Tadjine, and A. Nemra, "Robust waypoints navigation using fuzzy type 2 controller," in *Proc. Int. Conf. Adv. Electr. Eng. (ICAEE)*, Nov. 2019, pp. 1–6, doi: 10.1109/ICAEE47123.2019.9015101.

[11] P. Sharma and P. Bajaj, "Performance analysis of vehicle classification system using type-1 fuzzy, adaptive neuro-fuzzy and Type-2 fuzzy inference system," in *Proc. 2nd Int. Conf. Emerg. Trends Eng. Technol.*, 2009, pp. 581–584, doi: 10.1109/ICETET.2009.171.

[12] Y.-Y. Lin, J.-Y. Chang, N. R. Pal, and C.-T. Lin, "A mutually recurrent interval type-2 neural fuzzy system (MRIT2NFS) with self-evolving structure and parameters," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 3, pp. 492–509, Jun. 2013, doi: 10.1109/TFUZZ.2013.2255613.

[13] B. Son, J.-S. Kim, J.-W. Kim, Y.-J. Kim, and S.-Y. Jung, "Adaptive particle swarm optimization based on kernel support vector machine for optimal design of synchronous reluctance motor," *IEEE Trans. Magn.*, vol. 55, no. 6, pp. 1–5, Jun. 2019, doi: 10.1109/TMAG.2019.2902935.

[14] N. Baklouti and A. M. Alimi, "Interval type-2 beta fuzzy neural network for wheeled mobile robots obstacles avoidance," in *Proc. Int. Conf. Adv. Syst. Electric Technol. (IC_ASET)*, Jan. 2017, pp. 481–486, doi: 10.1109/ASET.2017.7983740.

[15] J. Chen, F. Ye, and T. Jiang, "Numerical analyses of three inertia-weight-improvement-based particle swarm optimization algorithms," in *Proc. 2nd IEEE Int. Conf. Comput. Intell. Appl. (ICCIA)*, Sep. 2017, pp. 150–154, doi: 10.1109/CIAPP.2017.8167198.

[16] J. Liu, S. Anavatti, and M. G. Hussein Abbass, "Comprehensive learning particle swarm optimisation with limited local search for UAV path planning," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 2287–2294, doi: 10.1109/SSCI44817.2019.9002992.

[17] Y. Ren and S. Liu, "Modified particle swarm optimization algorithm for engineering structural optimization problem," in *Proc. 13th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2017, pp. 504–507, doi: 10.1109/CIS.2017.00117.

[18] M. Schwaab, E. C. Biscaia Jr., J. L. Monteiro, and J. C. Pinto, "Nonlinear parameter estimation through particle swarm optimization," *Chem. Eng. Sci.*, vol. 63, no. 6, pp. 1542–1552, Mar. 2008.

[19] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: A new metaheuristic optimization algorithm," *Comput. Struct.*, vol. 139, pp. 98–112, Jul. 2014.

[20] C.-J. Lin, H.-Y. Lin and C.-Y. Yu, "Using a type-2 neural fuzzy controller for navigation control of evolutionary robots," in *Proc. Int. Symp. Comput., Consum. Control (ISC)*, 2018, pp. 306–309, doi: 10.1109/IS3C.2018.00084.

[21] X. Zhang, X. Zhang, S. L. Ho, and W. N. Fu, "A modification of artificial bee colony algorithm applied to loudspeaker design problem," *IEEE Trans. Magn.*, vol. 50, no. 2, pp. 737–740, Feb. 2014.

[22] Y.-L. Li, Z.-H. Zhan, Y.-J. Gong, W.-N. Chen, J. Zhang, and Y. Li, "Differential evolution with an evolution path: A DEEP evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1798–1810, Sep. 2015.

[23] S. L. Ho, S. Yang, G. Ni, and J. Huang, "A quantum-based particle swarm optimization algorithm applied to inverse problems," *IEEE Trans. Magn.*, vol. 49, no. 5, pp. 2069–2072, May 2013.

[24] M. A. Kunna, T. A. A. Kadir, M. A. Remli, N. M. Ali, K. Moorthy, and N. Muhammad, "An enhanced segment particle swarm optimization algorithm for kinetic parameters estimation of the main metabolic model of Escherichia coli," *Processes*, vol. 8, no. 8, p. 963, Aug. 2020.

[25] B. M. Sahoo, H. M. Pandey, and T. Amgoth, "GAPSO-H: A hybrid approach towards optimizing the cluster based routing in wireless sensor network," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100772.

[26] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York, NY, USA: McGraw-Hill, 1976.

CHENG-JIAN LIN (Senior Member, IEEE) 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. He is currently a Chair Professor with 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 robot.

BING-HONG CHEN received the B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung City, Taiwan, in 2022. He is currently pursuing the Ph.D. degree with the Ph.D. Program, Prospective Technology of Electrical Engineering and Computer Science, National Chin-Yi University of Technology. His current research interests include type-2 neural fuzzy systems, evolutionary computation, machine learning, and computer vision and application.

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.