

Research Article

Motion Control of a Mobile Robot Using the Hedge–Algebras-Based Controller

Sy-Tai Nguyen ^{1,2} Thi-Thoa Mac ^{1,3} and Hai-Le Bui ^{1,3}

¹School of Mechanical Engineering, Hanoi University of Science and Technology, No 1 Dai Co Viet Street, Hanoi, Vietnam

²Viet Nam National Institute of Occupational Safety and Health, Hanoi, Vietnam

³Intelligent Mechatronics and AI Research Group, School of Mechanical Engineering, Hanoi University of Science and Technology, No 1 Dai Co Viet Street, Hanoi, Vietnam

Correspondence should be addressed to Hai-Le Bui; le.buihai@hust.edu.vn

Received 1 February 2023; Revised 5 June 2023; Accepted 25 October 2023; Published 15 November 2023

Academic Editor: Constantin Florin Caruntu

Copyright © 2023 Sy-Tai Nguyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Hedge–algebras (HA) theory provides a useful mathematical tool for modeling the linguistic values of a linguistic variable. These values are quantified by real numbers between 0 and 1. Therefore, the HA-based controller (HAC) has many advantages over the traditional fuzzy set theory-based controller (FC) in setup steps, control efficiency, computation time, and optimization. This study aims to control the avoidance of obstacles in the workspace and move to the destination of an autonomous robot using HAC, in which the HAC is optimized using the balancing composite motion optimization (BCMO) to return the optimal path. In which the investigated model is inherited from a reference. The HAC is established and optimized to minimize the traveling distance of the mobile robot and help it to avoid obstacles simultaneously. Simulations include one and two obstacle environments. Design variables, when optimizing, include the fuzzy parameters of linguistic variables and the reference range of state variables. This work is the first study in motion control of mobile robots based on the HA theory. The simulation data show that the proposed control rule base suits the mobile robot models. Therefore, the control efficiency of HAC is higher than that of a FC both in terms of the traveling distance of the robot and computation time (CPU time). Also, the establishment steps of the HAC controller show that HAC is more explicit, easier to optimize, and simpler to operate than FC. Research results in the present work also indicate that HAC can be developed and applied in motion control problems for different robot models with the advantages of a smaller traveling distance and faster computation time.

1. Introduction

Mobile robot systems have increasing applications in industry in general and in the field of automation in particular. Studies on these systems are attracting considerable attention from scientists to different types of mobile robots, such as walking robots, wheeled robots, or mobile robots equipped with manipulators [1–4].

Motion control of mobile robot systems is often based on artificial intelligence (AI) algorithms in which controllers based on the fuzzy set theory (FC) of Zahde play an essential role in this research direction. In a study by Zhao et al. [5], the research model was a three-wheel robot with nine sensors that use two FCs to control obstacle avoidance motion based on the distance from around the robot to the obstacles. The

fuzzy controller in a study by Yen and Cheng [6] was optimized using the ant colony algorithm to find the shortest path and avoid obstacles for a mobile robot. In a study by Sun et al. [7], the motion of a three-dimensional autonomous underwater vehicle was controlled by an optimized FC using particle swarm optimization. A Mamdani fuzzy control was used in a study by Chiu and Lin [8] to control the motion of an omnidirectional spherical mobile robot. An adaptive fuzzy neural controller was developed in a study by Kong et al. [9] to control a multirobot system with robots' unknown dynamics and environment. Investigations in a study by Wu et al. [10] focus on developing a backstepping algorithm based on the fuzzy sliding technique to control a mobile robot with uncertainty and noise. Aouf et al. [11] have developed the optimal fuzzy controller for navigating a mobile robot in strange

environments. In a study by Guo et al. [12], a path planning algorithm was optimized based on fuzzy control to navigate for a mobile robot, where the robot environment was unknown. The fuzzy controller was investigated in a study by Hacene and Mendil [13] for a three-wheeled robot in an environment of static and dynamic targets and obstacles. The robot performs the tasks of approaching the target, avoiding obstacles, and following the wall. The problem of controlling the speed of a mobile robot using a fuzzy controller was presented in a study by Hartono and Nizar [14], with the state variables being speed and the change in the robot's speed. A type 2 fuzzy controller was introduced in a study by Humaidi et al. [15] for the trajectory tracking of a 3-Revolute-Revolute-Revolute planar parallel robot. The PD fuzzy controller was proposed in a study by Jabeur and Seddik [16] to control the motion of a two-wheeled robot and avoid damaging its motor. The adaptive type 2 fuzzy controller proposed in a study by Ha et al. [17] was for motion control of a wheeled mobile robot with external disturbances, including wheel slip. The fuzzy controller in a study by Kumar et al. [18] was optimized to control multiple mobile robots in static and dynamic environments. The above publications show the popularity and efficiency of fuzzy control for mobile robot models with their different environmental conditions.

The hedge-algebras (HA) theory has been developed since 1990 with different approaches from fuzzy set theory in modeling linguistic values of linguistic variables [19]. In HA, linguistic values can be represented as real numbers with magnitudes consistent with linguistic values' natural semantics [20]. This representation allows simplifying the computational steps in HA. Therefore, HA has been effectively applied to problems in information technology and process control fields, especially in the control of mechanical models, for example, underactuated systems [21], earthquake-resistant building models [22, 23], car suspension systems [24], and industrial robot models [25, 26]. Research results in the above publications show that HA-based controller (HAC) has many advantages compared to FC, such as a more explicit setup, more straightforward optimization, more efficient operation, and significantly faster computation time.

Balancing composite motion optimization (BCMO) is a recently published swarm-based optimization method. The advantages of BCMO are no need for algorithm parameters, fast convergence, and high optimal performance [27].

With the potential of HA in the field of control of mechanical models proven through the above publication, in this paper, HAC is applied for the first time to the motion control problem of mobile robots in which a simple model of an obstacle avoidance mobile robot is referenced. The proposed control rule system of HAC is suitable for mobile robot behavior. The HAC is optimized using BCMO to help the robot find the shortest path while avoiding obstacles. The main contributions of the present study include: (1) a novel development in motion control of an autonomous robot using the HAC, in which the HAC is optimized using the BCMO to return the optimal path; (2) the controller is designed to achieve two objectives simultaneously: minimizing the traveling distance of the mobile robot and enabling obstacle

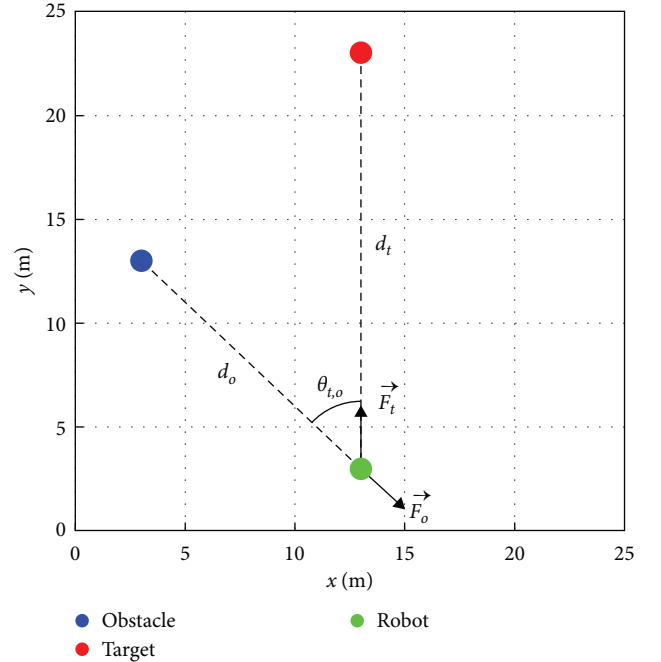


FIGURE 1: Investigated model.

avoidance. The controller has been established and optimized to balance these two goals effectively; (3) the comparison of the proposed method with FC in terms of traveling distance and computation time (CPU time) has shown the outstanding efficiency of HAC; (4) in some scenarios, when using FC, the robot encounters a stuck phenomenon when approaching an obstacle, and HA still controls the robot reach the target successfully.

The remaining part of this paper has the following structure: Section 2 introduces the robot model. Section 3 presents the proposed controller based on HA and the BCMO. The numerical simulations are presented and discussed in Section 4. Finally, a conclusion to sum up the paper and future works is given in Section 5.

2. Investigated Model

In this study, the mobile robot model [28] is used (see Figure 1). The model includes a square moving environment with a size of each side of 25 m. The obstacle, target, and differential robot are circular with a radius of 0.5 m. The robot's movement is made by two unit force vectors \vec{F}_t (directed from the robot's current position to the target's position) and \vec{F}_o (directed from the robot's current position to the obstacle's position), as shown in Figure 1.

Combining the above forces through a weight w will produce a general control force \vec{F} , as shown in Equation (1), to bring the robot to the target position and help it to avoid obstacles simultaneously.

$$\vec{F} = w\vec{F}_o + (1 - w)\vec{F}_t, 0 \leq w \leq 1. \quad (1)$$

The sharp changes in the direction of the mobile robot are limited to $[-\pi/4, \pi/4]$ for each control loop. Therefore, the robot's direction $\theta_r(k+1)$ in the $(k+1)$ th control loop is

calculated through $\theta_r(k)$ in the k th control loop as follows:

$$\theta_r(k+1) = \theta_r(k) + \Delta\theta, \quad (2)$$

$$\Delta\theta = \begin{cases} \frac{\pi}{4} & \text{if } \theta - \theta_r(k) > \frac{\pi}{4} \\ \theta - \theta_r(k) & \text{if } -\frac{\pi}{4} \leq \theta - \theta_r(k) \leq \frac{\pi}{4} \\ -\frac{\pi}{4} & \text{if } \theta - \theta_r(k) < -\frac{\pi}{4} \end{cases}, \quad (3)$$

where θ is the angle between the control force \vec{F} and the x -axis and $\Delta\theta$ is the robot's direction variation between the k th and $(k+1)$ th iterations. Let s be the robot's speed and Δt be the sample period. The change in position of the robot (x_r, y_r) in the $(k+1)$ th control loop compared to the k th loop is given as follows:

$$\begin{aligned} x_r(k+1) &= x_r(k) + s\Delta t \cos \theta_r \\ y_r(k+1) &= y_r(k) + s\Delta t \sin \theta_r \end{aligned} \quad (4)$$

Let $\alpha = d_o/d_t$ be the ratio between the distance from the robot to the obstacle (d_o) and the distance from the robot to the target (d_t) and $\theta_{t,o}$ is the angle between the directions from the robot to the target and to the obstacle (see Figure 1). In order for the robot to move to the target and avoid obstacles, the weight w must follow the following observation: weight w has a high value when α and $\theta_{t,o}$ are low to avoid obstacles, and it has a low value for the remaining cases, so that the robot quickly moves to the target. The weight w will be determined through the proposed rule base of the HAC controller, as shown in Section 3.

It is assumed that the robot has a constant velocity, no dynamic constraints, and knows its current position and the position of the obstacle to simplify the investigated problem.

3. Control Design

In HA, a HA structure of a linguistic variable X usually consists of primary terms, hedges, and constants. For example, $C = (\text{low}, \text{high}) = (c^-, c^+)$, denoted by (L, H), are negative and positive primary terms; $H = (\text{little}, \text{very}) = (h^-, h^+)$, represented by (l, v), are negative and positive hedges; and (absolutely low, neutral, absolutely high) = (0, W, 1) are constants. Linguistic values will be formed based on a combination of primary terms and hedges following the natural semantic order of these linguistic values, for example, absolutely L, vvL, vL, lvL, L, llL, lL, vlL, W, vH, lH, llH, H, lvH, vH, vvH, and absolutely H [29].

The notations $fm(c^-)$ and $v(h^-)$ are fuzziness measures of the negative primary term and the negative hedge, respectively, and their values range from 0 to 1. The theorems in HA allow determining fuzzy measures φ (between 0 and 1) of all linguistic values of a linguistic variable based on the value of $fm(c^-)$ and $v(h^-)$ [29]. The basic formulas in HA theory are expressed in Appendix A. In the case of $fm(c^-)$ and $v(h^-)$ changing between 0.2 and 0.8, the variation range

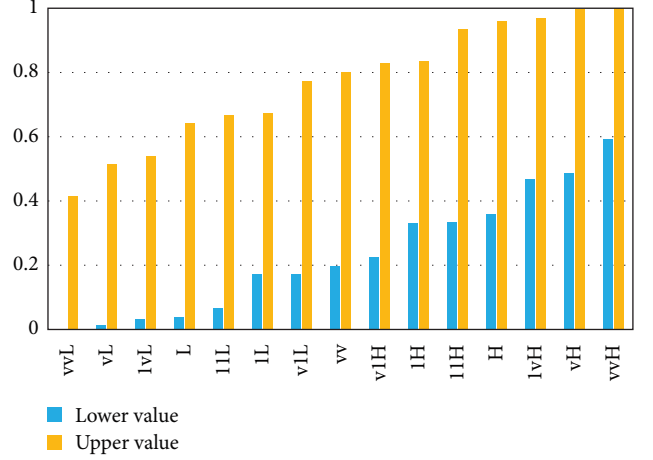


FIGURE 2: Fuzziness measure values φ of typical linguistic values.

of fuzziness measure values of several typical linguistic values is shown in Figure 2.

As shown in Figure 2, the values have quite wide ranges when $fm(c^-)$ and $v(h^-)$ change. Furthermore, the semantic order of linguistic values is always strictly guaranteed. These ranges are the basis for performing the HAC optimization problem, as will be presented below.

The block diagram of the mobile robot control process is shown in Figure 3. Because the robot knows its current position and the position of the obstacle, the position and posture of the robot and the position of the obstacle are determined at any given time. These parameters allow easy determining α and $\theta_{t,o}$. The parameters α and $\theta_{t,o}$ are the inputs of the HAC controller, as shown in Figure 4. The weight w from the output of the HAC determines the general control force \vec{F} through Equation (1). Under the action of this force, the robot will change direction (see Equations (2) and (3)) and position (see Equation (4)). This process is repeated until the robot meets the target or collides with the obstacles.

The control flowchart of HAC is shown in Figure 4(a) in which the normalization and denormalization blocks have the same role as the fuzzification and defuzzification steps of a traditional fuzzy controller.

It is assumed that the variables α and $\theta_{t,o}$ have linguistics values L, W, and H and the variable w has linguistics values vvL, vL, L, W, lH, H, and vH. Hence, the variation ranges of the fuzziness measure S of the variables α , $\theta_{t,o}$, and w are $[S(L), S(H)]$, $[S(L), S(H)]$, and $[S(vvL), S(vH)]$, respectively.

The normalization step to convert the real values of the variables α and $\theta_{t,o}$ to their fuzziness measure values S is shown in Figure 4(b)–4(c) in which α_s and θ_s are representations of α and $\theta_{t,o}$ in the fuzziness measure domain, and a and b represent reference intervals in the real domain of the variables α and $\theta_{t,o}$. The HA's rule base to determine the weight w is proposed, as shown in Figure 4(d). This rule base satisfies the analysis of the behavior of the weight w , as described in Section 2. The geometrical representation of this rule base is shown in Figure 4(e). From the data, as shown in Figure 4(e), the value of w_s (the notation of w in the fuzziness measure domain) can be easily determined

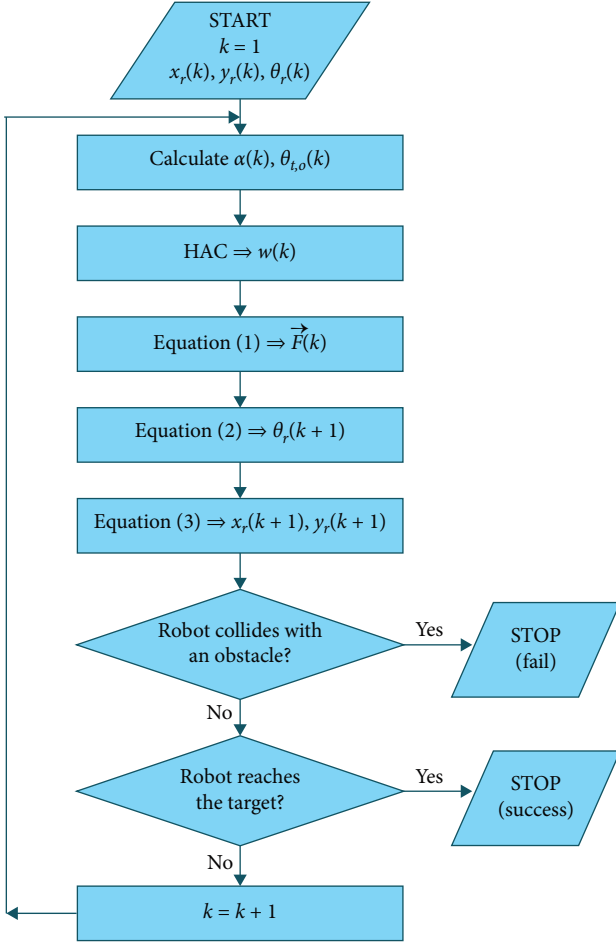


FIGURE 3: The block diagram of the mobile robot control process.

through a simple interpolation (using Matlab's `interp2` command). This interpolation scheme can be considered as the inference step of HAC. Finally, the denormalization step, as shown in Figure 4(f), converts the value of w from the fuzziness measure domain to the real one.

Hence, it can be seen that the operating principle of HAC is similar to a traditional fuzzy controller (FC). Function blocks of HAC only use linear interpolation. In addition, when optimizing fuzzy parameters of HAC, only two design variables ($fm(c^-)$ and $v(h^-)$) are needed for each linguistic variable. The constraint on the semantic order of the linguistic values is ignored because this order is always guaranteed (see Figure 2). Therefore, it can be concluded that HAC is more explicit, easier to optimize, and more straightforward to operate than FC.

The HAC controller will be optimized to improve the mobile robot's performance. The optimization problem is expressed as follows:

- (1) Objective function [28]:

$$\text{Traveling distance(robot)} = > \min. \quad (5)$$

- (2) Design variables: $fm(c^-)$ and $v(h^-)$ of α , $\theta_{t,o}$, and w and a and b . Hence, the total number of design variables is 8.

- (3) Constraints [28]:

$$\text{Collision between robot and obstacle} = 0. \quad (6)$$

$$\text{Contact between robot and target} = 1. \quad (7)$$

- (4) Training data [28]: The initial position and direction of the robot are [13 2] and $\pi/2$, the position of the target is [13 22], and the positions of the obstacle are [3 12] and [13 18].

- (5) Optimization tool: BCMO algorithm is used with population size (denoted by NP) and number of generations (represented by maxGen) equal to 200. The optimal algorithm flowchart is shown in Figure 5.

Hence, it can be seen that the controller's parameters include the following:

- (1) As analyzed in Section 2, the number of input state variables is 2 (α and $\theta_{t,o}$), and the number of output control variables (u) is 1.
- (2) In HA-based controllers, the number of linguistic values of input state variables is usually odd (3, 5, or 7), representing the symmetry between the linguistic values on both sides of the neutral element (W). Therefore, the minimum number of linguistic values of two state variables to be selected is 3. For the control variable, the number and arrangement of linguistic values in the rule base are determined by the trial and error method in which a low value of w is prioritized, and, therefore, the robot quickly moves to the target.
- (3) In addition, the reference ranges of variables are also essential parameters and significantly affect the controller's performance. The reference range of the control variable is normalized in the range [0, 1] [28]. The reference ranges of state variables are determined from the optimization problem.
- (4) Finally, the fuzzy parameters of the variables ($fm(c^-)$ and $v(h^-)$) are also optimized to improve the control efficiency of the HAC.

4. Numerical Simulation

4.1. Comparative Study with One Obstacle. This section presents the numerical simulation results in which the robot's speed s is 0.5 m/s and the sample period Δt is 0.1 s. Simulation results between HAC and FC [28] will be compared for some typical scenarios. The parameters of the controller FC [28] are presented in Appendix B.

The optimal values of the design variables of HAC are shown in Table 1. From these values, the fuzziness measure of the linguistic values of the variables is calculated and shown in Table 2. The values, as shown in Table 2, are brought into the rule base, as shown in Figure 4(d), to obtain the optimal HA's inference schema, as shown in Figure 4(e). The simulated scenarios are given in Table 3.

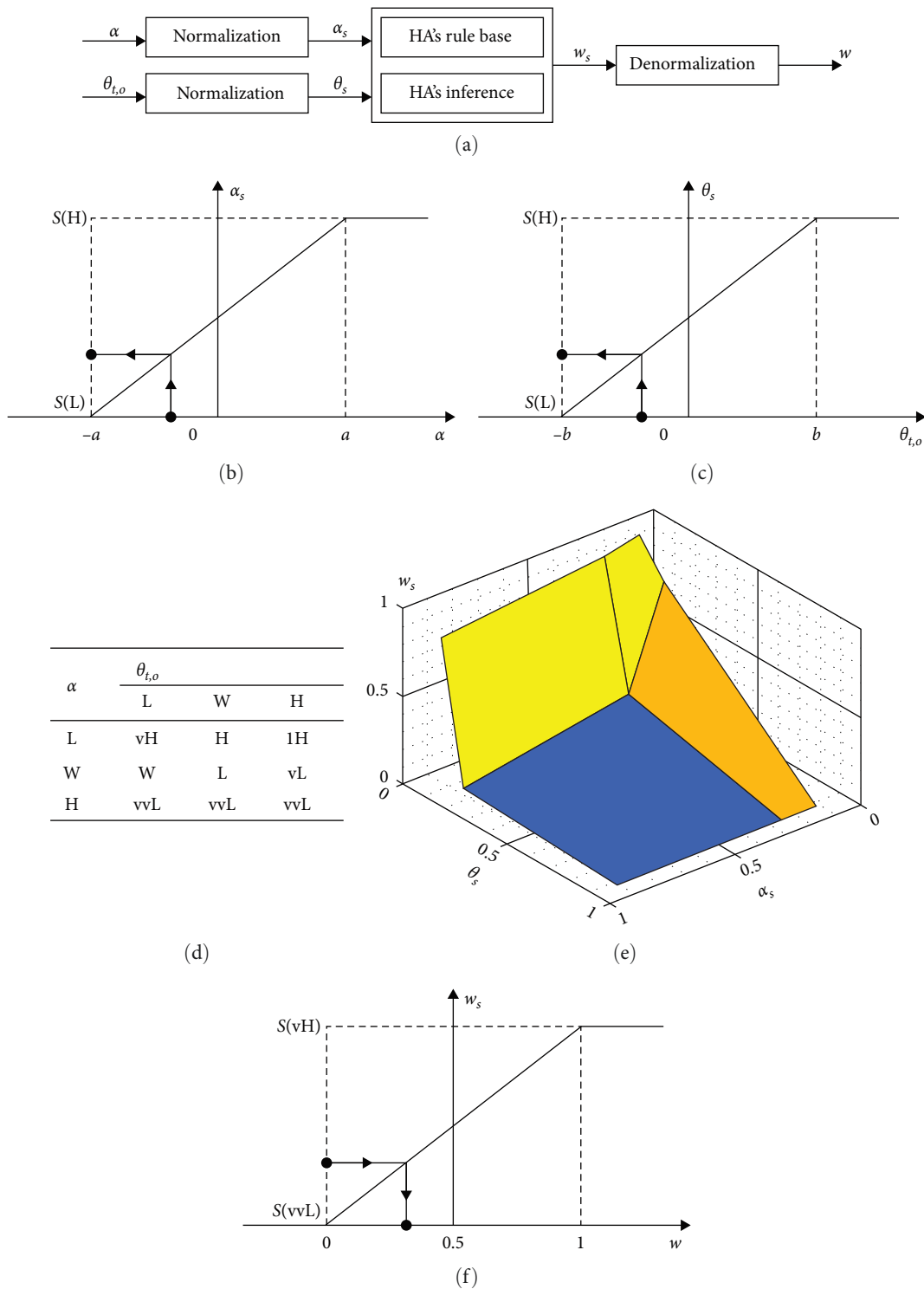


FIGURE 4: Hedge-algebras-based controller. (a) Control flowchart of HAC, (b and c) normalization step of the input variables, (d) HA's rule base, (e) rule base's geometrical representation, and (f) de-normalization step of the output variable.

Simulation results for the above cases in this comparative study are shown in Figures 6–9. Table 4 shows the traveling distance of the robot when using the HAC and FC controllers.

As shown in Figures 6–9, the proposed HAC is implemented and compared to FC through several test cases, assuming that the robot is performing a respective mission

in a 25×25 square meter working space. In each scenario, the obstacles are generated with different positions, and the robot has a similar start (magenta circle) and destination point.

The results, as shown in Figures 6–9, also show that the control goal is always achieved. That is, the robot reaches the

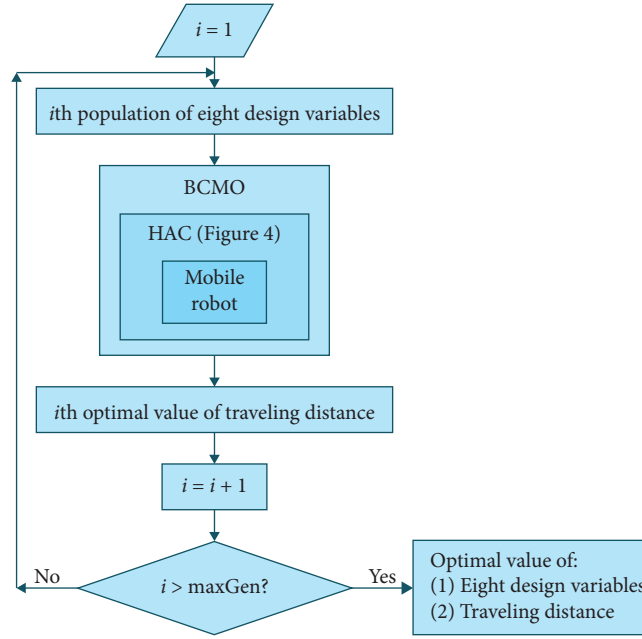


FIGURE 5: The optimal algorithm block diagram.

TABLE 1: Optimal values of design variables.

$fm(c^-)$ of α	$v(h^-)$ of α	$fm(c^-)$ of $\theta_{t,o}$	$v(h^-)$ of $\theta_{t,o}$	$fm(c^-)$ of w	$v(h^-)$ of w	a	b
0.2	0.570	0.266	0.534	0.8	0.708	1.996	2.259

TABLE 2: Fuzziness measure of linguistic values.

α			$\theta_{t,o}$			w						
L	W	H	L	W	H	vvL	vL	L	W	LH	H	vH
0.086	0.2	0.936	0.127	0.266	0.921	0.02	0.068	0.234	0.8	0.841	0.941	0.983

TABLE 3: The simulated scenarios in the comparative study.

Scenario	Obstacle position (m)	Robot's initial position (m)	Robot's initial direction (rad)	Target position (m)
1	[11 12]	[13 2]	$\pi/2$	[13 22]
2	[13 18]	[13 2]	$\pi/2$	[13 22]
3	[10 10]	[5 20]	$-\pi/2$	[15 4]
4	[8 8]	[19 19]	$-\pi$	[5 5]

target and avoids obstacles. The movement trajectory of the robot when using HAC and FC is quite similar. However, the robot's trajectory in scenarios 2 and 4 in the case of using FC encounters a stuck phenomenon when the robot approaches an obstacle. In general, the trajectories of the robot in the cases of using HAC are smoother than in the cases of using FC. The traveling distance of the robot when using HAC is slightly smaller than when using FC (more than a 3% decrease on the average of all four scenarios). In addition, the controllers' computation time (CPU time) is also measured and listed, as shown in Table 5. It can be seen from Table 5 that the CPU time of HAC is significantly reduced

compared to FC (about 44%). The CPU time proves that HAC is improved considerably, resulting in a reduced computational factor. In various real-time applications, the computation of the robot controller is restricted, so a control method with a lower computational cost, like HAC, is a better option. Therefore, this approach has the potential for many real-life robotic applications, such as autonomous cars, military robots, agricultural robots, exploration robots, and rescue robots in real-time control applications [30] in which CPU time results are measured on the computer with the configuration: CPU E5-1650 v3 @ 3.50 GHz, 32.0 GB RAM, using Matlab software and Windows 10 operating system.

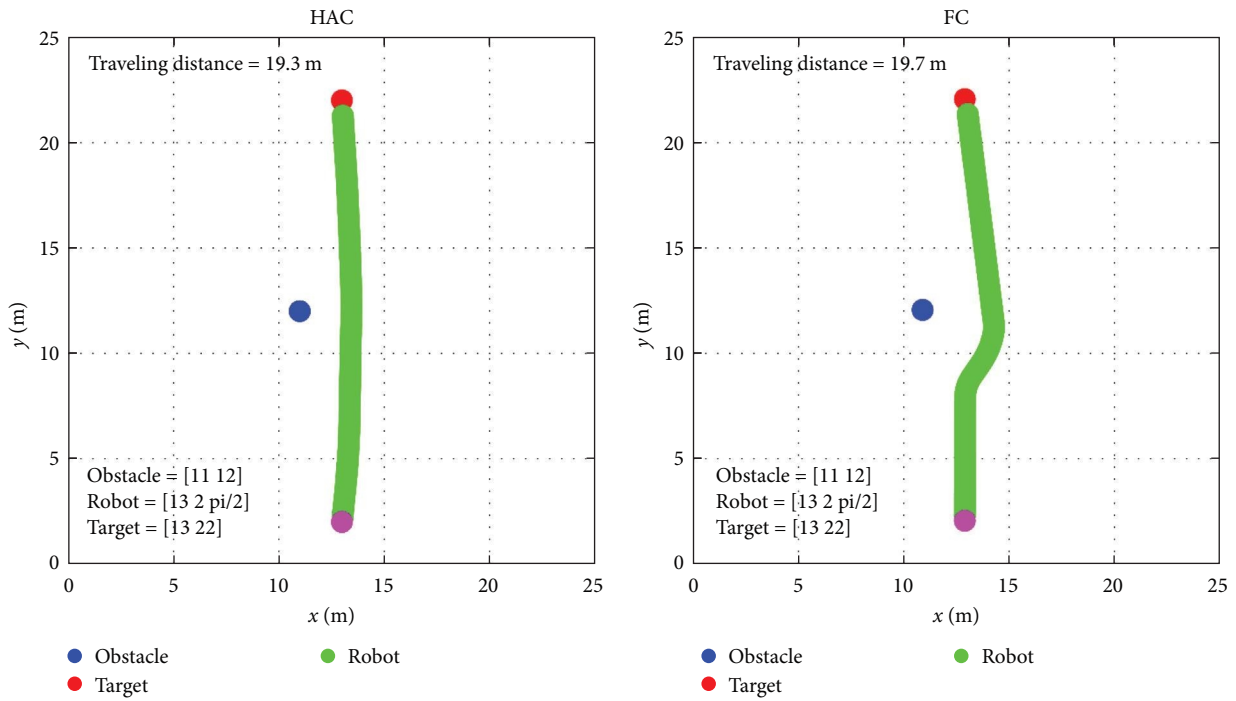


FIGURE 6: Scenario 1.

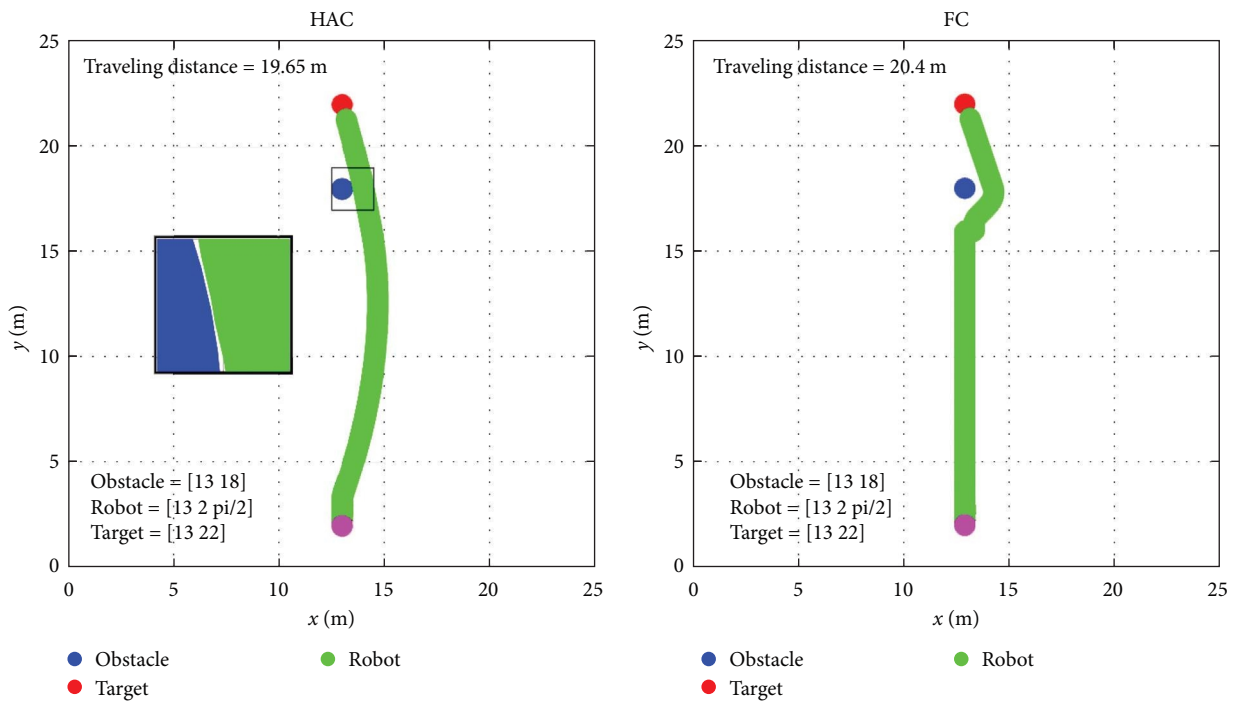


FIGURE 7: Scenario 2.

4.2. Study in the Case of Two Obstacles. Next, an obstacle is added to the investigated model to validate the effectiveness of the proposed HAC. It is noted that the parameters of the HAC are kept the same as in Subsection 4.1. To suit the new environment, the HAC just adds one condition as follows: the obstacle to be concerned about (avoid) is the obstacle whose distance to the robot is smaller. The simulation

scenarios in this subsection are shown in Table 6. The simulation results are shown in Figures 10–13.

The result, as shown in Figures 10–13, shows that, in the case of the environment with two obstacles, the robot still meets the control objectives, including avoiding obstacles and reaching the target. The robot's trajectory is smooth, and the robot's traveling distance is quite optimal.

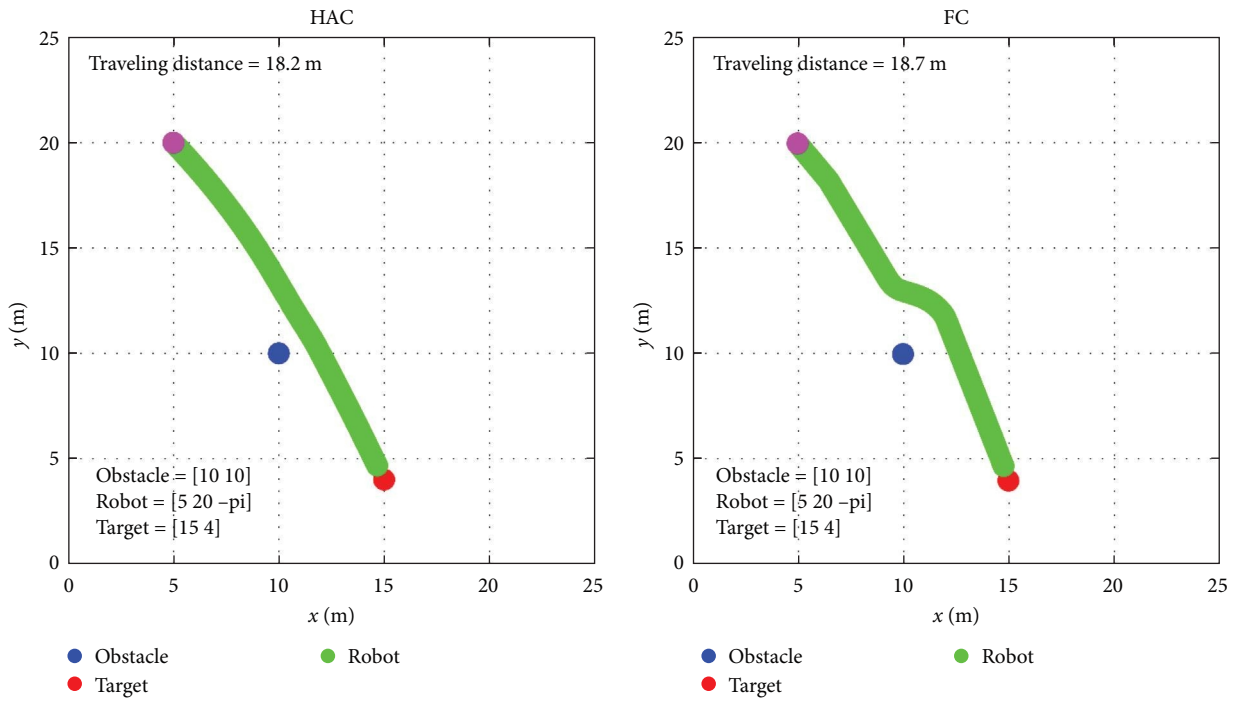


FIGURE 8: Scenario 3.

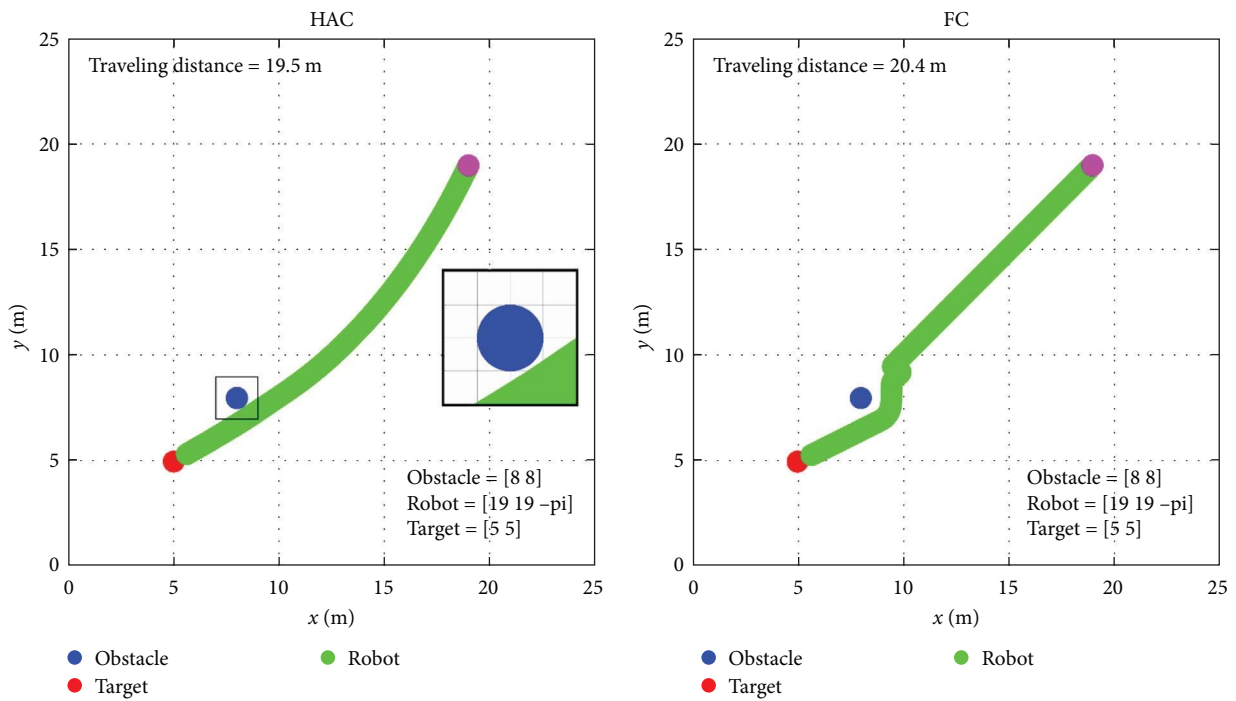


FIGURE 9: Scenario 4.

TABLE 4: The traveling distance (m) of the robot.

Scenario	1	2	3	4	Mean value
HAC	19.30	19.65	18.20	19.50	19.16
FC	19.70	20.40	18.70	20.40	19.80

TABLE 5: CPU time (s).

Controller	HAC	FC	Reduction ratio (%)
CPU time (s)	~0.318	~0.566	~44

TABLE 6: The simulated scenarios with two obstacles.

Scenario	Obstacle position (m)	Robot's initial position (m)	Robot's initial direction (rad)	Target position (m)
5	[12.5 10; 14.5 18]	[13 2]	$\pi/2$	[13 22]
6	[10 12.5; 17 15]	[1 13]	0	[22 13]
7	[9 10; 9 15]	[5 20]	$-\pi/2$	[15 4]
8	[8 10; 14 13]	[19 19]	$-\pi/1.5$	[5 5]

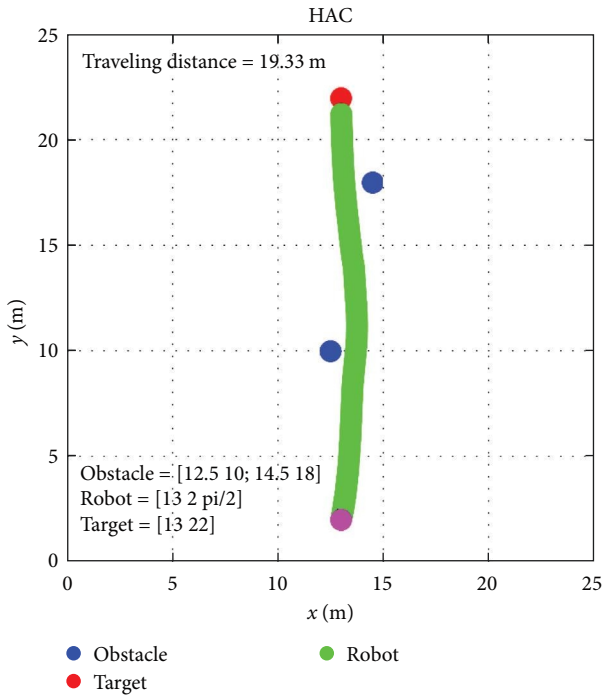


FIGURE 10: Scenario 5.

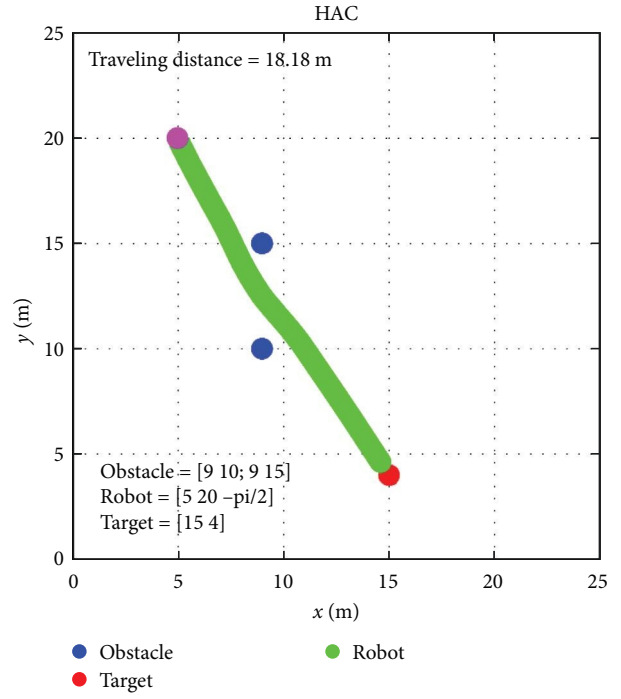


FIGURE 12: Scenario 7.

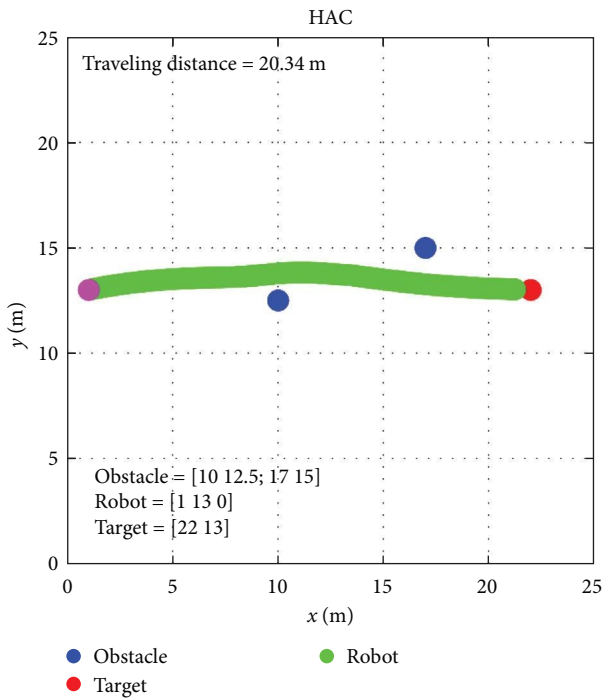


FIGURE 11: Scenario 6.

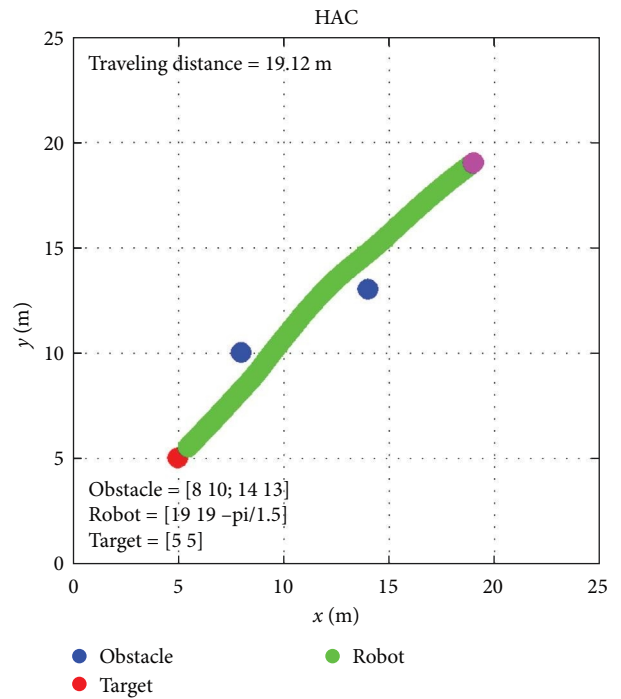


FIGURE 13: Scenario 8.

5. Conclusion

In this study, the motion control problem of a mobile robot to avoid obstacles and reach the target using the HAC is performed. This work is the first time the HA theory using the BCMO optimization algorithm has been developed for a mobile robot application. In order to prove the effectiveness of the proposed method, two numerical simulations are implemented, and the simulation results between HAC and FC [28] are compared. These observations highlight differences between the HAC and FC approaches regarding trajectory smoothness, obstacle handling, and traveling efficiency. The smoother trajectories and reduced traveling distance with HAC can indicate more efficient and optimized robot movements such as a 3% reduction in traveling distance in case the environment has one obstacle, and especially the CPU time of HAC is much faster than that of FC, reducing the computation time by 44% compared with the FC controller by using BCMO optimization algorithm for HAC. When changing the simulation environment (switching from one to two obstacles), the HAC still gives good control performance with a minor additional condition. Especially in some scenarios, when using FC, the robot encounters a stuck phenomenon when approaching an obstacle, HA still controls the robot to reach the target successfully. These advantages of HAC allow developing and applying HAC in more complex navigation problems of mobile robot systems with many applications in life, such as agricultural, exploration, or military robots.

This study's limitations include as follows: the robot model is simple, the number of obstacles is small, complex shapes of obstacles have not been considered, ignoring the robot's dynamic model and unknown and dynamic obstacles, and many simplified assumptions of the robot's velocity and sensor and dynamic constraints of the model are used. Simulations under different conditions, such as the effect of different dimensions on throughput and navigation time, have not been evaluated. In addition, experimental studies on a real mobile robot in a real-time environment have not been performed to verify the simulation results. These limitations, as well as the disturbance and model verification, are the issues that need to be addressed in future investigations of this research direction. Moreover, it is also necessary to evaluate the proposed control algorithm's computational complexity and compare it with other control algorithms, such as the type 2 fuzzy controller. In addition, the proposed method can be integrated with a machine learning algorithm to increase the robot system's intelligence to solve robot navigation in complex environments with different conditions. In this sense, other optimization techniques, for example, the whale optimization algorithm (WOA) [31] or the particle swarm optimization (PSO) [32], should also be considered in the optimization design of controllers based on HA theory.

Appendix

A. The Basic Formulas in HA Theory

This section presents the basic formulas in HA theory for the case that the HA structure of the linguistic variable X consists of two hedges h^- and h^+ .

The fuzziness measure of primary terms $fm \in [0, 1]$ is given as follows:

$$fm(c^-) + fm(c^+) = 1 \text{ and } \sum_{h \in H} fm(hu) = fm(u), \text{ for } \forall u \in X, \quad (\text{A.1})$$

$$\text{For the constants } 0, W \text{ and } 1, fm(0) = fm(W) = fm(1) = 0, \quad (\text{A.2})$$

$$\text{For } \forall x, y \in X, \forall h \in H, \frac{fm(hx)}{fm(x)} = \frac{fm(hy)}{fm(y)} = v(h) \quad (\text{A.3})$$

The notation $v(h)$ is called the fuzziness measure of the hedge h . From Equation (A.3), we have:

$$fm(hx) = v(h)fm(x), \text{ for } \forall x \in X, \quad (\text{A.4})$$

$$fm(h^-c) = fm(c), c \in \{c^-, c^+\}, \quad (\text{A.5})$$

$$fm(h^-x) = fm(x), \quad (\text{A.6})$$

$$v(h^-) = \alpha; v(h^+) = \beta, \text{ in which } \alpha, \beta > 0, \alpha + \beta = 1. \quad (\text{A.7})$$

The function $\text{Sign}: X \rightarrow \{-1, 0, 1\}$ is defined as follows:

$$\text{Sign}(c^-) = -1, \text{Sign}(c^+) = +1, \quad (\text{A.8})$$

$$\text{Sign}(hc) = -\text{Sign}(c), \text{ if } h \text{ is negative with respect to } c, \quad (\text{A.9})$$

$$\text{Sign}(hc) = +\text{Sign}(c), \text{ if } h \text{ is positive with respect to } c, \quad (\text{A.10})$$

$$\text{Sign}(h'hx) = -\text{Sign}(hx), \text{ if } h'hx \neq hx \text{ and } h' \text{ are negative with respect to } h, \quad (\text{A.11})$$

$$\text{Sign}(h'hx) = +\text{Sign}(hx), \text{ if } h'hx \neq hx \text{ and } h' \text{ are positive with respect to } h, \quad (\text{A.12})$$

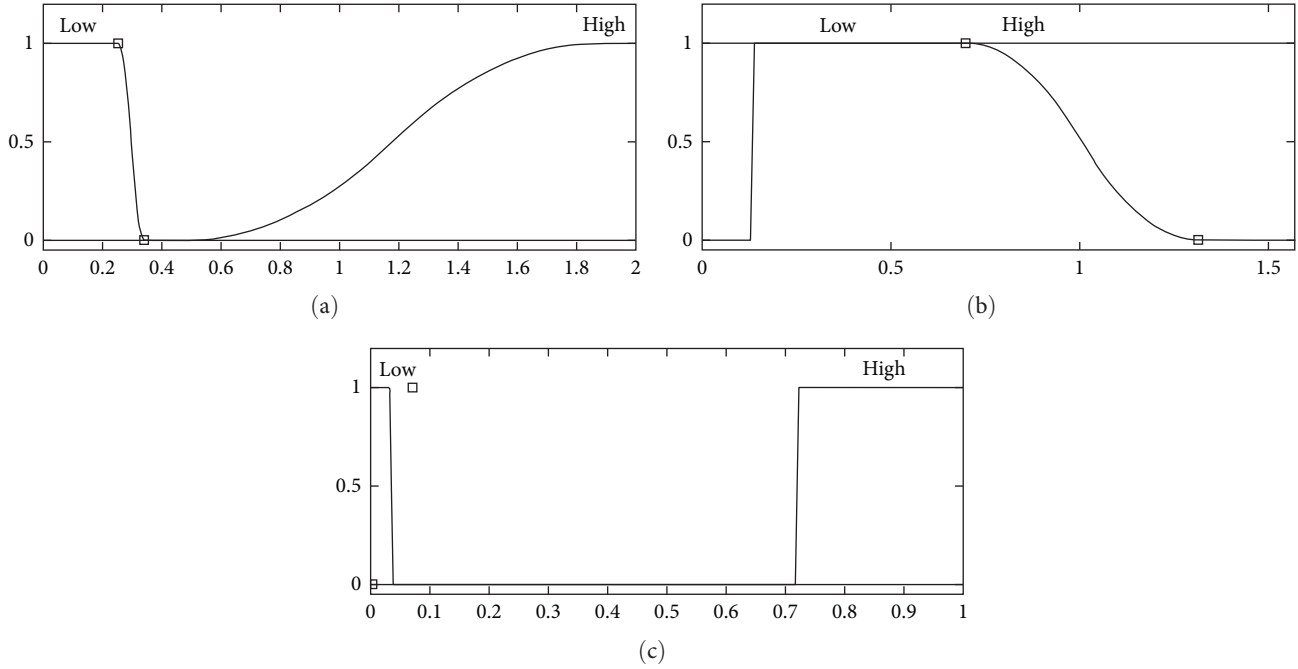
$$\text{Sign}(h'hx) = 0 \text{ if } h'hx = hx. \quad (\text{A.13})$$

The fuzziness measure $S: X \rightarrow [0,1]$ of linguistic values is defined as follows:

$$S(W) = \gamma = fm(c^-), S(c^-) = \gamma - \alpha fm(c^-) = \beta fm(c^-), S(c^+) = \gamma + \alpha fm(c^+), \quad (\text{A.14})$$

$$S(h_jx) = S(x) + \text{Sign}(h_jx) \left\{ \sum_{i=\text{Sign}(j)}^j fm(h_i x) - \omega(h_jx) - fm(h_jx) \right\}, \quad (\text{A.15})$$

where $h_{-1} = h^-$, $h_1 = h^+$, and $\omega(h_jx) = 0.5[1 + \text{Sign}(h_jx)\text{Sign}(h_p h_jx)(\beta - \alpha)]$.

FIGURE 14: Fuzzification of variables: α (a), $\theta_{t,o}$ (b), and w (c).

For example, when linguistic values are modeled symmetrically, $fm(c^-) = v(h^-) = 0.5$, the fuzziness measure S of several linguistic values is calculated as follows, where $(c^-, c^+) = (L, H)$ and $(h^-, h^+) = (l, v)$, as described in Section 3:

$$v(L) = \alpha = 0.5; \quad v(H) = \beta = 1 - v(L) = 0.5; \quad fm(L) = 0.5; \quad fm(H) = 1 - fm(L) = 0.5.$$

$$S(W) = fm(L) = 0.5.$$

$$S(L) = fm(L) - \alpha fm(L) = 0.5 - 0.5 \times 0.5 = 0.25.$$

$$S(vL) = S(L) + \text{Sign}(vL) \times (fm(vL) - 0.5fm(vL)) = 0.25 + (-1) \times 0.5 \times 0.5 \times 0.5 = 0.125.$$

$$S(lL) = S(L) + \text{Sign}(lL) \times (fm(lL) - 0.5fm(lL)) = 0.25 + (+1) \times 0.5 \times 0.5 \times 0.5 = 0.375.$$

$$S(H) = \beta + \alpha fm(H) = 0.5 + 0.5 \times 0.5 = 0.75.$$

$$S(vH) = S(H) + \text{Sign}(vH) \times (fm(vH) - 0.5fm(vH)) = 0.75 + (+1) \times 0.5 \times 0.5 \times 0.5 = 0.875.$$

$$S(lH) = S(H) + \text{Sign}(lH) \times (fm(lH) - 0.5fm(lH)) = 0.75 + (-1) \times 0.5 \times 0.5 \times 0.5 = 0.625.$$

$$S(vvL) = S(vL) + \text{Sign}(vvL) \times (fm(vvL) - 0.5fm(vvL)) = 0.125 + (-1) \times 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.0625.$$

$$S(lvL) = S(vL) + \text{Sign}(lvL) \times (fm(lvL) - 0.5fm(lvL)) = 0.125 + (+1) \times 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.1875.$$

$$S(vlL) = S(lL) + \text{Sign}(vlL) \times (fm(vlL) - 0.5fm(vlL)) = 0.375 + (-1) \times 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.3125.$$

$$S(llL) = S(lL) + \text{Sign}(llL) \times (fm(llL) - 0.5fm(llL)) = 0.375 + (+1) \times 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.4375.$$

$$S(llH) = S(lH) + \text{Sign}(llH) \times (fm(llH) - 0.5fm(llH)) = 0.625 + (-1) \times 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.5625.$$

$$S(vlH) = S(lH) + \text{Sign}(vlH) \times (fm(vlH) - 0.5fm(vlH)) = 0.625 + (+1) \times 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.6875.$$

$$S(lvH) = S(vH) + \text{Sign}(lvH) \times (fm(lvH) - 0.5fm(lvH)) = 0.875 + (-1) \times 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.8125.$$

$$S(vvH) = S(vH) + \text{Sign}(vvH) \times (fm(vvH) - 0.5fm(vvH)) = 0.875 + (+1) \times 0.5 \times 0.5 \times 0.5 \times 0.5 = 0.9375.$$

B. The Parameters of the Controller FC

This section presents the parameters of the controller FC [28]. The control flowchart of FC is similar to that of HAC, as shown in Figure 4(a). The normalization and denormalization blocks have the same role as the FC fuzzification and defuzzification steps. The number of control rules and the parameters of the input functions of FC were optimized using genetic algorithms. The fuzzification diagram of the variables of FC is shown in Figure 14.

The rule base of FC contains two control rules as below:

IF α is “low” and $\theta_{t,o}$ is “low” then w is “high.”

IF $\theta_{t,o}$ is “high” then w is “low.”

The Mamdani style and the centroid method are used for the FC’s inference engine and defuzzification.

Data Availability

The authors confirm that the data supporting the findings of this study are available within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Authors’ Contributions

Sy-Tai Nguyen: supervision, data curation, software, investigation, writing—original draft preparation, and writing—

reviewing and editing. Thi-Thoa Mac: data curation, software, investigation, and writing—original draft preparation. Hai-Le Bui: conceptualization, methodology, software, validation, writing—original draft preparation, and writing—reviewing and editing.

Acknowledgments

This research is funded by the Hanoi University of Science and Technology (HUST) under project number T2023-PC-007.

References

- [1] S. G. Tzafestas, “Mobile robot control and navigation: a global overview,” *Journal of Intelligent & Robotic Systems*, vol. 91, no. 1, pp. 35–58, 2018.
- [2] M. N. A. Wahab, S. Nefti-Meziani, and A. Atyabi, “A comparative review on mobile robot path planning: classical or meta-heuristic methods?” *Annual Reviews in Control*, vol. 50, pp. 233–252, 2020.
- [3] M. A. K. Niloy, A. Shama, R. K. Chakraborty et al., “Critical design and control issues of indoor autonomous mobile robots: a review,” *IEEE Access*, vol. 9, pp. 35338–35370, 2021.
- [4] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, “A review: on path planning strategies for navigation of mobile robot,” *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.
- [5] R. Zhao, D. H. Lee, and H. K. Lee, “Mobile robot navigation using optimized fuzzy controller by genetic algorithm,” *The International Journal of Fuzzy Logic and Intelligent Systems*, vol. 15, no. 1, pp. 12–19, 2015.
- [6] C.-T. Yen and M.-F. Cheng, “A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance,” *Microsystem Technologies*, vol. 24, no. 1, pp. 125–135, 2018.
- [7] B. Sun, D. Zhu, and S. X. Yang, “An optimized fuzzy control algorithm for three-dimensional AUV path planning,” *International Journal of Fuzzy Systems*, vol. 20, no. 2, pp. 597–610, 2018.
- [8] C.-H. Chiu and C.-M. Lin, “Control of an omnidirectional spherical mobile robot using an adaptive Mamdani-type fuzzy control strategy,” *Neural Computing and Applications*, vol. 30, no. 4, pp. 1303–1315, 2018.
- [9] L. Kong, W. He, C. Yang, Z. Li, and C. Sun, “Adaptive fuzzy control for coordinated multiple robots with constraint using impedance learning,” *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3052–3063, 2019.
- [10] X. Wu, P. Jin, T. Zou, Z. Qi, H. Xiao, and P. Lou, “Backstepping trajectory tracking based on fuzzy sliding mode control for differential mobile robots,” *Journal of Intelligent & Robotic Systems*, vol. 96, no. 1, pp. 109–121, 2019.
- [11] A. Aouf, L. Boussaid, and A. Sakly, “Same fuzzy logic controller for two-wheeled mobile robot navigation in strange environments,” *Journal of Robotics*, vol. 2019, Article ID 2465219, 11 pages, 2019.
- [12] J. Guo, C. Li, and S. Guo, “A novel step optimal path planning algorithm for the spherical mobile robot based on fuzzy control,” *IEEE Access*, vol. 8, pp. 1394–1405, 2019.
- [13] N. Hacene and B. Mendil, “Fuzzy behavior-based control of three wheeled omnidirectional mobile robot,” *International Journal of Automation and Computing*, vol. 16, no. 2, pp. 163–185, 2019.
- [14] R. Hartono and T. N. Nizar, “Speed control of a mobile robot using fuzzy logic controller,” *IOP Conference Series: Materials Science and Engineering*, vol. 662, Article ID 022063, 2019.
- [15] A. J. Humaidi, H. T. Najem, A. Q. Al-Dujaili, D. A. Pereira, I. K. Ibraheem, and A. T. Azar, “Social spider optimization algorithm for tuning parameters in PD-like interval type-2 fuzzy logic controller applied to a parallel robot,” *Measurement and Control*, vol. 54, no. 3-4, pp. 303–323, 2021.
- [16] C. B. Jabeur and H. Seddik, “Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot,” *Asian Journal of Control*, vol. 23, no. 1, pp. 23–41, 2021.
- [17] V. Q. Ha, S. H.-T. Pham, and N. T.-T. Vu, “Adaptive fuzzy type-II controller for wheeled mobile robot with disturbances and wheel slips,” *Journal of Robotics*, vol. 2021, Article ID 6946210, 11 pages, 2021.
- [18] S. Kumar, D. R. Parhi, A. K. Kashyap, and M. K. Muni, “Static and dynamic path optimization of multiple mobile robot using hybridized fuzzy logic-whale optimization algorithm,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 235, no. 21, pp. 5718–5735, 2021.
- [19] N. C. Ho and W. Wechler, “Extended hedge algebras and their application to fuzzy logic,” *Fuzzy Sets and Systems*, vol. 52, no. 3, pp. 259–281, 1992.
- [20] N. C. Ho and N. Van Long, “Fuzziness measure on complete hedge algebras and quantifying semantics of terms in linear hedge algebras,” *Fuzzy Sets and Systems*, vol. 158, no. 4, pp. 452–471, 2007.
- [21] H.-L. Bui, M.-N. Pham, and T.-T.-H. Nguyen, “Swing-up control of an inverted pendulum cart system using the approach of Hedge-algebras theory,” *Soft Computing*, vol. 26, no. 10, pp. 4613–4627, 2022.
- [22] H.-L. Bui, N.-A. Tran, and H. Q. Cao, “Active control based on hedge-algebras theory of seismic-excited buildings with upgraded tuned liquid column damper,” *Journal of Engineering Mechanics*, vol. 149, no. 1, Article ID 04022091, 2023.
- [23] V.-B. Bui, T.-T. Mac, and H.-L. Bui, “Design optimization considering the stability constraint of the Hedge-algebras-based controller for building structures subjected to seismic excitations,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems Control Engineering*, vol. 237, no. 10, pp. 1822–1837, 2023.
- [24] T.-T. Mac, T.-D. Nguyen, H.-L. Bui, and N.-A. Tran, “Optimal design of hedge-algebras-based controller for vibration control of vehicle suspension systems,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2023.
- [25] P. B. Khoi and N. V. Toan, “Hedge-algebras-based controller for mechanisms of relative manipulation,” *International Journal of Precision Engineering and Manufacturing*, vol. 19, no. 3, pp. 377–385, 2018.
- [26] T.-L. Bui, T.-H. Nguyen, and X.-T. Nguyen, “A controller for delta parallel robot based on hedge algebras method,” *Journal of Robotics*, vol. 2023, Article ID 2271030, 20 pages, 2023.
- [27] T. Le-Duc, Q.-H. Nguyen, and H. Nguyen-Xuan, “Balancing composite motion optimization,” *Information Sciences*, vol. 520, pp. 250–270, 2020.
- [28] “Tune fuzzy robot obstacle avoidance system using custom cost function,” Since R2019a.
- [29] T.-D. Nguyen and H.-L. Bui, “General optimization procedure of the Hedge-algebras controller for controlling dynamic systems,” *Artificial Intelligence Review*, vol. 56, pp. 2749–2784, 2023.

- [30] C. Hernández-Mejía, H. Vázquez-Leal, A. Sánchez-González, and Á. Corona-Avelizapa, "A novel and reduced cpu time modeling and simulation methodology for path planning based on resistive grids," *Arabian Journal for Science and Engineering*, vol. 44, no. 3, pp. 2321–2333, 2019.
- [31] Z. A. Waheed and A. J. Humaidi, "Design of optimal sliding mode control of elbow wearable exoskeleton system based on whale optimization algorithm," *Journal Européen des Systèmes Automatisés*, vol. 55, no. 4, pp. 459–466, 2022.
- [32] A. J. Humaidi, S. K. Kadhim, and A. S. Gataa, "Optimal adaptive magnetic suspension control of rotary impeller for artificial heart pump," *Cybernetics and Systems*, vol. 53, no. 1, pp. 141–167, 2022.