Neuro-Fuzzy Combination for Reactive Mobile Robot Navigation: A Survey

Brahim Hilali[1], Mohammed Ramdani[2], Abdelwahab Naji[3]
[1,2]Department of Computer Science, Faculty of Sciences and Technics of Mohammedia, Hassan II University Casablanca, Morocco
[3]Department of Electronics, Computer Science and Telecommunications, National School of Applied Sciences of Oujda, Mohammed I University, Morocco

Article Info

Article history:

Received Jul 20, 2022
Revised Mar 20, 2023
Accepted Apr 26, 2023

Keywords:

Reactive navigation
Neuro-fuzzy combination
Mobile robot path planning
Fuzzy inference system
ANFIS
Metaheuristic

ABSTRACT

Autonomous navigation of mobile robots is a fruitful research area because of the diversity of methods adopted by artificial intelligence. Recently, several works have generally surveyed the methods adopted to solve the path-planning problem of mobile robots. But in this paper, we focus on methods that combine neuro-fuzzy techniques to solve the reactive navigation problem of mobile robots in a previously unknown environment. Based on information sensed locally by an onboard system, these methods aim to design controllers capable of leading a robot to a target and avoiding obstacles encountered in a workspace. Thus, this study explores the neuro-fuzzy methods that have shown their effectiveness in reactive mobile robot navigation to analyze their architectures and discuss the algorithms and metaheuristics adopted in the learning phase.

Corresponding Author:

Brahim Hilali,
Department of Computer Science, Faculty of Sciences and Technics of Mohammedia,
Hassan II University Casablanca, Morocco.
Email: brahim.hilali-etu@etu.univh2c.ma

1. INTRODUCTION

Autonomous navigation is a crucial task in the field of mobile robotics. It consists in giving the robot the ability to move in workspaces to reach its target without human intervention. This task is divided into three phases (Figure 1):

- Perception of the environment: the robot uses onboard sensors to detect objects (walls, moving objects, targets) in the environment.
- Planning: using the information of the perception phase, the robot employs a control system (planner) to determine the suitable behavior.
- Action: using a mechanical system, the robot performs the behavior decided by the planner.

In the literature, we distinguish two categories of path planning methods, classical methods that aim to find an optimized path between the starting point and the target in a previously known environment (global planning or offline planning), and reactive methods that operate in unknown environments based on sensor information at each state of the robot (local planning or online planning) [1], [2]. The second type of planning performs reactive navigation.

Several articles [2]–[7] have reviewed path planning methods to discuss their characteristics and compare their effectiveness in solving different scenarios of autonomous navigation of mobile robots. But our work focuses on neuro-fuzzy methods because they have shown their effectiveness in developing reactive navigation controllers and they present several architectures that require exploration to analyze and discuss them. These neuro-fuzzy architectures consist in combining two approaches: neural network and fuzzy logic. Thus, several researchers [8]–[35] have designed reactive navigation controllers based on the neuro-fuzzy combination to take advantage of the strengths of these two approaches while avoiding their drawbacks. As a result, we discovered that several neuro-fuzzy architectures are used for robot navigation. In [10]–[12], the authors adopted a cascaded neuro-fuzzy architecture that contains two consecutive phases, a neural network as a preprocessor and a controller based on a fuzzy inference system. While in [14], the author adopted a hybrid neuro-fuzzy architecture based on type-2 fuzzy inference system. This architecture consists in designing an adaptive neural network to adjust the parameters of the premise and the consequent parts using the Least Mean Square (LMS) algorithm. Another hybrid architecture that has shown its effectiveness in control systems is adaptive-network-based fuzzy inference system (ANFIS). Proposed by Jang in 1993 [36], this approach implements fuzzy inference systems of the Takagi-Sugeno type [37] (or type-3 according to Jang [36]) in an adaptive network framework. To minimize the computation, it is based on hybrid learning, i.e., it uses the gradient descent method to adjust the premise parameters and Least Square Estimation (LSE) to determine the consequent parameters. In this way, the author of [18] implemented a reactive navigation controller based on the standard ANFIS. Whereas in [25]–[27] the authors developed reactive navigation controllers based on variants of the ANFIS architecture to avoid the computational complexity and local minima problems. Thus, the author of [25] used the Cuckoo Search (CS) metaheuristic, the author of [26] used the Invasive Weed Optimization (IWO) metaheuristic, and the author of [27] used the Teaching-Learning-Based Optimization (TLBO) metaheuristic instead of gradient descent to adjust the parameters of the premise part. In other works [30]–[35], the authors designed ANFIS-based mobile robot navigation controllers by adding methods to avoid various problems such as deadlock escape [30], irregular obstacle avoidance [31], and fuzzy rule base reduction [33]–[35].

In this perspective, we aim to survey the neuro-fuzzy architectures designed for reactive mobile robot navigation. Accordingly, the rest of this paper is presented as follows. In section 2, we will explain the context and characteristics of the reactive navigation problem addressed in this work. In section 3, we will discuss the advantage and the process of using the neural network approach to solve the reactive navigation problem. In section 4, we will explain the architecture of a reactive navigation controller based on a fuzzy inference system and we will discuss the three types of fuzzy systems that exist in the literature. In section 5, we will explore the neuro-fuzzy combination architectures adopted for reactive navigation which are the cascaded neuro-fuzzy architecture, the hybrid adaptive neuro-fuzzy architecture based on type-2 fuzzy inference system, and the hybrid adaptive neuro-fuzzy architecture based on type-3 fuzzy inference system with its variants ANFIS, CS-ANFIS, IWO-ANFIS, and TLBO-ANFIS. The exploration of these architectures aims to explain their components and the algorithms used in the learning phase. Finally, in section 6, We will discuss the performance of these architectures as controllers for mobile robot navigation through a detailed analysis that considers parameters such as input/output of the control system, navigation in environments with dynamic obstacles and targets, navigation in concave obstacles, and results in simulation and real-time environments.

Figure 1. Autonomous navigation phases

2. REACTIVE NAVIGATION PROBLEM: CHARACTERISTICS AND WORKSPACE OF THE MOBILE ROBOT

In the field of mobile robotics, navigation is considered a crucial task that allows the robot to move in workspaces. Indeed, several types of problems and contexts can be encountered depending on the characteristics of the robot and environment. The case of the problem addressed in this work is as follows:

In a two-dimensional workspace, a mobile robot with two differential wheels moves autonomously to reach a given target without having any prior information about the environment. This robot is equipped with an onboard system that allows it to measure the distances to the nearest obstacles. This system consists of sensors that measure the left obstacle distance (LOD), sensors that measure the right obstacle distance (ROD), and sensors that measure the front obstacle distance (FOD). Another sensor is also used to detect the target's position, which determines the target's angle (TA) between the direction of the robot's motion and the direction linking the robot with the target. Using an odometer, the robot can determine at each state the left wheel velocity (LWV) and the right wheel velocity (RWV) (Figure 2). The direction of robot depends on the right wheel velocity and the left wheel velocity. thus, to deviate the robot to the left, the left wheel is slowed down and the right wheel is accelerated, and vice versa. Therefore, reactive navigation consists in performing a sequence of translations and rotations to reach the target while avoiding obstacles (Figure 3). To perform this task, the robot must have a controller that allows it to predict its behaviors according to the sensor information. In other words, this controller receives inputs (LOD, ROD, FOD and TA) at each state and returns outputs (RWV and LWV) or steering angle (SA) that determine the robot's behavior.

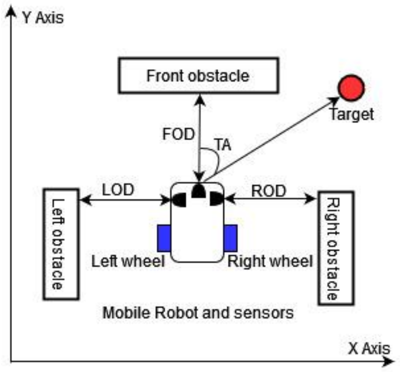

Figure 2. Mobile robot configuration

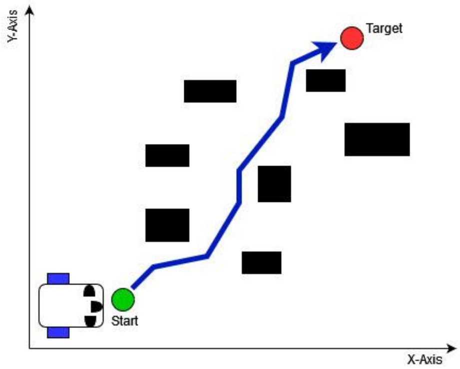

Figure 3. Scenario for reactive mobile robot navigation

3. NEURAL NETWORK FOR REACTIVE NAVIGATION

Neural network is a widely used technique for the implementation of mobile robot navigation applications. It can be adopted for environment perception, obstacle avoidance, and path planning to reach a target [38].

Several works [39]–[46] have used neural approach to solve the reactive navigation problems. This choice is made because of several advantages such as the ability to learn from data, the remarkable generalization ability if the training phase is perfectly carried out, the ability to approximate non-linear functions and parallel data processing.

Most neural controllers designed for mobile robot navigation use a feed-forward backpropagation multi-layer network (Figure 4) to model the relationship between the inputs (LOD, FOD, ROD, TA) and the output, which can be the steering angle (SA) or the wheel velocity (LWV, RWV). The choice of the number of hidden layers and the number of neurons is done empirically to facilitate the training of neural network [10].

The learning phase of neural network is considered an important factor in the success of optimal and efficient navigation. This phase is based on generating a representative dataset (input, desired output), the choice of the error function (objective function) and the optimization method to adjust the parameters of neural network. The most commonly used error function in the literature is the mean square error function:

$$E = \frac{1}{N} \sum_{\substack{all\ training\\ patterns}} (\theta_{desired} - \theta_{actual})^2 \quad (1)$$

Where θ_{actual} is the output of neural network, $\theta_{desired}$ is the desired output and N is the number of training patterns.

The gradient descent is a method for minimizing the error function. At each iteration *t*, the network's parameters W are adjusted by computing the gradient of the MSE function and applying the following formula:

$$W(t+1) = W(t) - \alpha\, \partial E / \partial W \quad (2)$$

However, studies [5], [10]–[12] show that combining neural network with other artificial intelligence technique like fuzzy inference system can improve navigation performance and solve some problems such as the path instability due to unexplored regions of inputs [12] and the inability to process uncertain sensor information.

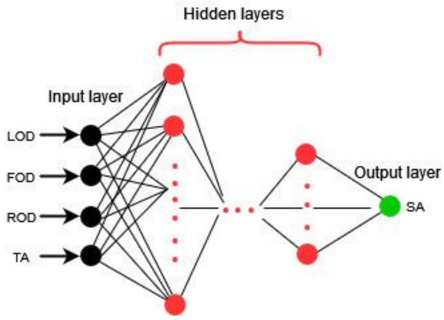

Figure 4. Neural network architecture for reactive mobile robot navigation

4. FUZZY LOGIC TECHNIQUE FOR MOBILE ROBOT NAVIGATION

The principle of reactive robot navigation is very similar to human movement. It uses uncertain measurements to locate the target and the near obstacles. For this reason, human expertise is highly demanded to design a control system that allows the robot to move through a workspace efficiently and safely. As a result, several researchers in mobile robotics [47]–[55] have approached reactive navigation using fuzzy inference systems. These systems are based on fuzzy logic and fuzzy sets. Fuzzy logic is an extension of Boolean logic, introduced by Lotfi Zadeh [56]. Among the advantages of this approach is the formalization of human reasoning in the form of natural language.

Figure 5 shows the architecture of a fuzzy inference system for path planning in a reactive navigation task. In this architecture, the inputs represent the distances of sensed obstacles (LOD, ROD, and FOD) and the target angle (TA), while the output constitutes the robot's motion that can be the left wheel velocity (LWV) and the right wheel velocity (RWV) or the steering angle (SA).

This architecture consists of three phases:

- Fuzzification phase: It is an operation that transforms a crisp value into a linguistic value. For this reason, membership functions are used to provide degrees of truth for each linguistic term, with a maximum value equal to 1 and a minimum value equal to 0. Several types of functions are used for fuzzification such as triangular, trapezoidal or Gaussian functions.
- Inference phase: It consists in using a rule base of fuzzy IF-THEN rules that are expressed in natural language with linguistic variables defined in the fuzzification phase. Several inference techniques exist in the literature, the most used are Mamdani inference and Sugeno inference.
- Defuzzification phase: It is the last step in fuzzy inference system. It consists in computing a crisp value for the overall fuzzy output. Several methods exist, such as the weighted average method, the centroid method and the maximum membership principle.

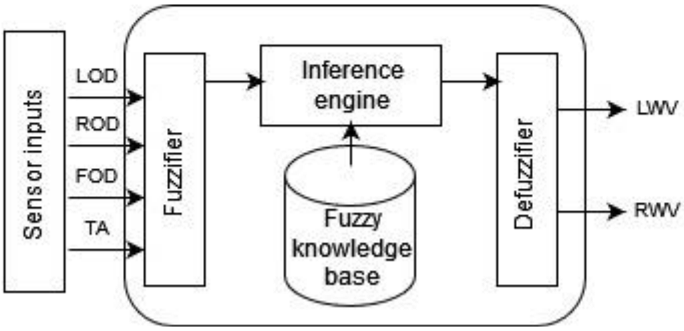

Figure 5. Fuzzy controller diagram for reactive mobile robot navigation

To perform safe and efficient navigation, a mobile robot follows the three behaviors: target-seeking (TS), obstacle-avoidance (OA) and wall-following (WF). In the literature, several strategies exist for the coordination of these behaviors. However, most fuzzy inference systems designed for navigation problems use behavior-based fuzzy reasoning. That is, each IF-THEN rule contributes to the final output according to its firing strength computed during the rule evaluation phase. For example, in [11], the author designed a fuzzy controller based on the three behaviors already mentioned by taking as input variables the distances (LOD, FOD and ROD) with linguistic terms (NEAR, MEDIUM and FAR) and also the target angle (TA) with linguistic terms (NEGATIVE, ZERO and POSITIVE) and as output variables the wheel velocities (LWV and RWV) with linguistic terms (SLOW, MEDIUM and FAST). The rule base of this controller contains twenty-four rules. As an example, we present below three fuzzy rules that represent respectively the obstacle-avoidance, wall-following and target-seeking behaviors:

- if LOD is FAR and FOD is NEAR and ROD is MEDIUM and TA is NEGATIVE then LWV is SLOW and RWV is FAST
- if LOD is FAR and FOD is FAR and ROD is NEAR and TA is ZERO then LWV is MEDIUM and RWV is MEDIUM
- if LOD is FAR and FOD is FAR and ROD is FAR and TA is POSITIVE then LWV is FAST and RWV is SLOW

Generally, Jang [36] classified the fuzzy inference systems proposed in the literature into three types according to the type of the adopted reasoning and the used fuzzy rules. Type-1 uses a monotonic membership function in the consequent part, and the overall output of the system is computed using the weighted average of the crisp output of each rule induced by the rule's firing strength and the output membership function, where the firing strength is the product or the minimum of premise part's membership degrees. Whereas, type-2 is characterized by a fuzzy output induced by the firing strength and the output membership function for each rule. The overall system output is computed by a technique as centroid of area after performing the aggregation of all the rules' fuzzy outputs. But Type-3 uses Takagi-Sugeno IF-THEN rules [37], in which each IF-THEN rule's output is a linear combination of the system's input variables, and the overall output is the weighted average of rules' outputs.

To conclude this section, fuzzy logic has a significant advantage in solving reactive navigation problem because it allows to deal with uncertain data and at the same time, it allows to express human expertise in the form of IF-THEN rules that resemble natural language. However, the difficulty is very clear in adjusting the parameters of membership functions as well as creating an optimal and efficient rule base. Therefore, the neuro-fuzzy combination is an effective solution to overcome this difficulty thanks to the self-learning capacity of the neural approach.

5. NEURO-FUZZY COMBINATION FOR MOBILE ROBOT NAVIGATION

Neural networks and fuzzy logic are among the techniques used in the field of artificial intelligence. To benefit from the strengths of these two techniques and to reduce their weaknesses, several studies [8]–[35] have proposed control systems based on the combination of these two techniques.

In this section, we explore neuro-fuzzy controllers designed during the last two decades for solving the reactive navigation problem of mobile robot, and also, we discuss the optimization algorithms and metaheuristics adopted in the learning phase.

Indeed, we distinguish three major categories of neuro-fuzzy architectures (Figure 6) for mobile robot navigation applications:

- A cascaded neuro-fuzzy architecture [10]–[12] in which the neural approach and the fuzzy approach cooperate and work separately, one is a pre-processor of the other.
- An integrated and adaptive neuro-fuzzy architecture [9], [13]–[35] in which the components of a fuzzy inference system are represented in a neural network to design the fuzzy rule base and adjust the parameters of membership functions. For this second category, two architectures are proposed depending on the type of the adopted reasoning and the used fuzzy rules.

First, we present a cascaded neuro-fuzzy architecture [10]–[12]. Next, we present an adaptive neuro-fuzzy architecture based on a type-2 fuzzy inference system [14]. Then, we present an adaptive neuro-fuzzy architecture based on a type-3 fuzzy inference system [18], [25]–[27]. Also, we explain its different variants according to the optimization algorithms used in the learning phase. After that, we describe a use case of ANFIS architecture for reactive navigation.

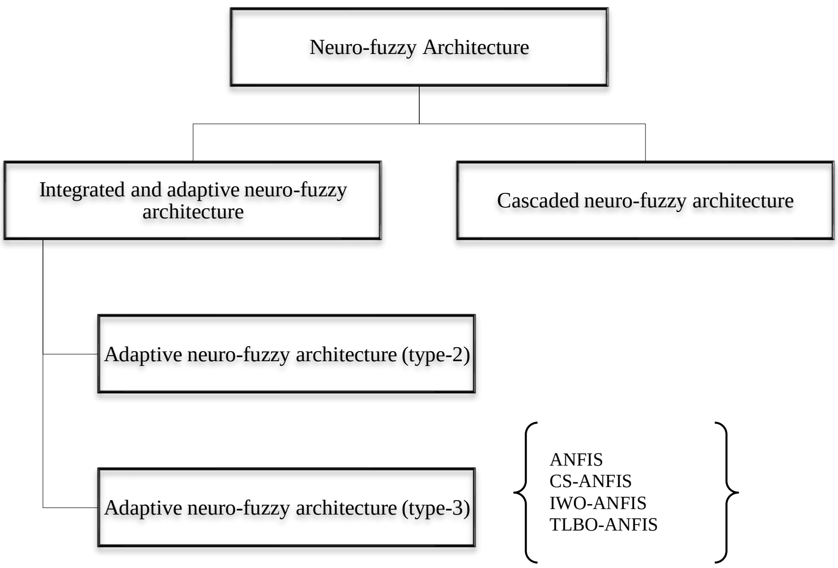

Figure 6. Neuro-fuzzy architectures for mobile robot navigation

5.1. Cascaded neuro-fuzzy architecture for reactive navigation

Several papers [10]–[12] have worked on this architecture to implement reactive navigation controllers. It consists of two distinct and cooperative stages (Figure 7). The first stage is a neural network that is a preprocessor of sensor information. It models the relationship between the inputs (LOD, ROD, FOD, and TA) and the output that represents the initial steering angle (Initial-SA). While the second stage is a fuzzy inference system that is a controller of the robot's final behavior. It receives as input the initial steering angle (output of the first stage) and the initial sensor information to predict the right wheel velocity (RWV) and the left wheel velocity (LWV), which determine the final steering angle (SA) of the robot.

In the first stage, the neural approach is chosen because it efficiently interprets the information sensed from the workspace thanks to its computational capacity [12]. in the second stage, the fuzzy system is chosen because of its ability to process uncertain information and to control the robot's final action.

The authors of [11], [12] adopted this architecture using for the first stage a feed-forward backpropagation multi-layer network. In [11], the training phase is performed using a set of training patterns representing typical scenarios. Whereas the training phase is based on discrete uniform sampling in [12]. Thus, the author performed quantization of the sensed values to provide discrete samples for neural training. For the second stage, the authors of [11], [12] adopted a fuzzy inference system using behavior-based fuzzy reasoning. So, fuzzy rules contribute to the realization of behaviors (target seeking, obstacle avoidance, and wall following) according to their firing strengths.

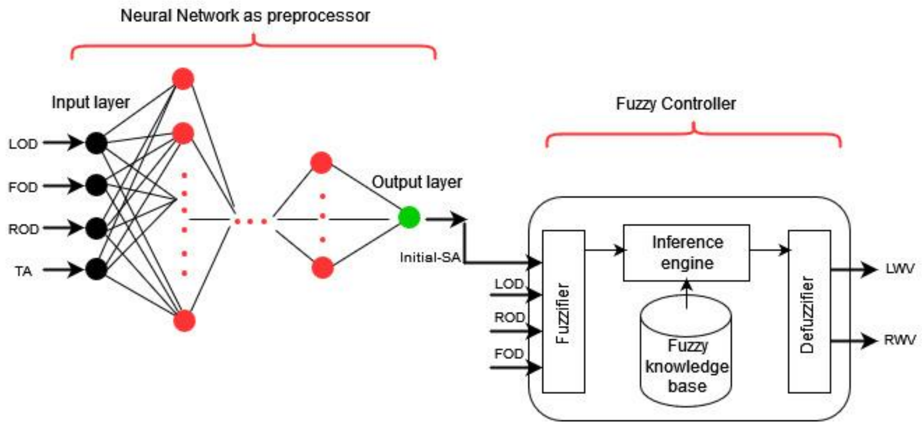

Figure 7. Cascaded Neuro-Fuzzy Controller

5.2. Adaptive neuro-fuzzy architecture (type-2) for reactive navigation

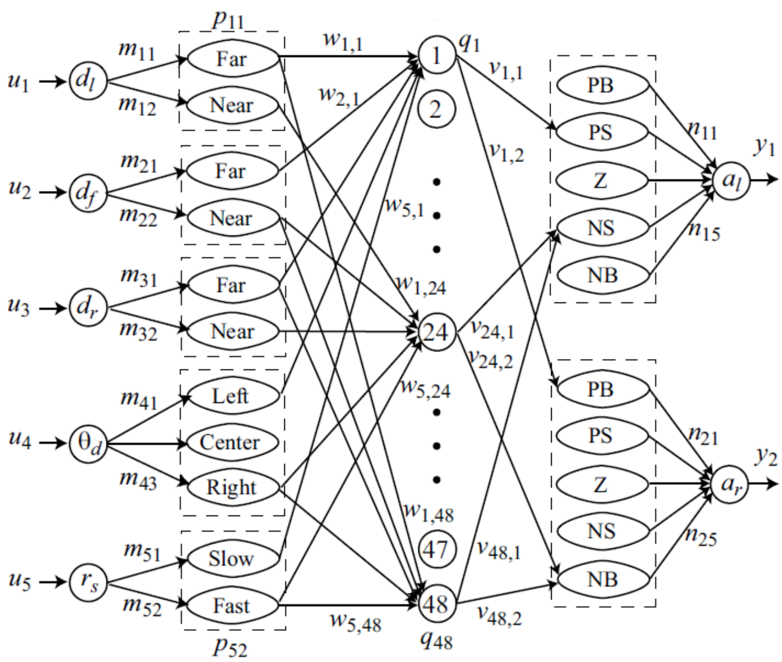

Figure 8. Adaptive Neuro-Fuzzy Controller *[14]*

d_l, left obstacle distance ; d_f, front obstacle distance ; d_r, right obstacle distance ; θ_d, target's angle ; r_s, robot's speed ; a_l, left wheel acceleration ; a_r, right wheel acceleration ; PB, positive big ; PS, positive small ; Z, zero ; NS, negative small; NB, negative big

Fuzzy inference systems-based controllers are regarded as an efficient solution for a reactive navigation task of mobile robots because they can handle uncertain and imprecise information and they can exploit human knowledge and experience. But the transformation of human knowledge into a fuzzy rule base and the tuning of the parameters of membership functions are a delicate task because there are no systematic methods. For this reason, some studies have integrated the components of a fuzzy inference system into a neural network to take advantage of its ability to learn from a sample of data. Thus, the author of [14] designed an adaptive neuro-fuzzy controller for mobile robot navigation that consists of a neural network representing the different components of type-2 fuzzy system (Figure 8). This controller receives as inputs the local information (LOD), (FOD), (ROD), (TA) and the robot's speed to predict the left wheel acceleration (LWA) and the right wheel acceleration (RWA). these accelerations determine the robot's movement to avoid the obstacles and reach the target.

Table 1a and Table 1b show a description of layers and parameters of this architecture respectively.

Table 1a. Description of neuro-fuzzy network layers

Layer	Description	Phase
1	It's an input layer. It receives crisp sensor values.	
2	It computes the degrees of membership to the linguistic terms for each input variable.	Fuzzification
3	Each node represents the conjunction of a rule's premise part(Forty-eight rules)	Inference
4	It expresses the consequent part of each rule (fuzzy rule output).	Inference
5	It computes the crisp values of the overall outputs (left wheel acceleration and right wheel acceleration) using the defuzzification formula.	Defuzzification

Table 1b. Description of neuro-fuzzy network's parameters and outputs

Parameters and node outputs	Designation	Computation Method
m_{ij}	Centers of input membership function	It is adjusted in the learning phase
p_{ij}	Input membership degrees to the linguistic terms	It is computed by applying the membership function on an input value
w_{ik}	Weights related to centers of input membership function	
q_k	The conjunction of Premise part of fuzzy rule k(firing strength of rule)	It is determined by applying the minimum of the membership degrees to the linguistic terms constituting the premise part of a rule
n_{ls}	Centers of output membership function	It is adjusted in the learning phase
v_{kl}	Weights related to centers of output membership function	

Forty-eight rules are designed to define this fuzzy controller and three behaviors are addressed in the rule base: target seeking, obstacle avoidance and wall following.

The defuzzification procedure maps the fuzzy output of the inference mechanism to the overall crisp output. As explained before, several methods can be used to convert a conclusion of the inference mechanism into a crisp output of the fuzzy controller. As long as the author of [14] used a type-2 fuzzy inference system, the center of gravity method is appropriate to compute the final output. Thus, the output values a_l and a_r are given by:

$$a_l = \frac{\sum_{k=1}^{48} v_{k,1} q_k}{\sum_{k=1}^{48} q_k} \tag{3}$$

$$a_r = \frac{\sum_{k=1}^{48} v_{k,2} q_k}{\sum_{k=1}^{48} q_k} \tag{4}$$

To generate a smoothed path by this controller, a Least Mean Square (LMS) learning algorithm is developed. It consists in optimizing the error between the controller's output and the desired output to adjust the parameters that represent the centers of membership functions of the input and output variables. For this purpose, the robot is trained (off-line) in a relatively complicated environment that contains all possible situations.

After adjusting the model's parameters, each rule is presented as a weight vector. To optimize the fuzzy inference step, the author of [14] proposed an algorithm that eliminates redundant rules. This algorithm consists in defining the similarity between rules by computing the Euclidean difference between the weight vectors. Therefore, the number of rules becomes less than forty-eight based on a tolerance value.

5.3. Adaptive neuro-fuzzy architecture (type-3) for reactive navigation

This architecture is an adaptive network-based type-3 fuzzy inference system (ANFIS). In other words, it consists in implementing a type-3 fuzzy inference system in an adaptive network framework using a hybrid learning procedure [36]. It aims to exploit the advantage of fuzzy logic in representing human expertise using IF-THEN rules, and at the same time to map between input-output data pairs through neural network to adjust the parameters of membership function. Designed by Jang [36] in 1993, ANFIS adopts a Takagi-Sugeno [37] type fuzzy inference system. That is, the output of each IF-THEN rule is a linear combination of the input variables. For this reason, Jang [36] adopted a hybrid learning to adjust the parameters of the premise and consequent parts. This learning strategy consists of a forward pass and a backward pass. In the forward pass of the hybrid learning algorithm, the premise parameters are fixed, the outputs of the neurons are computed until the fourth layer, and the consequent parameters are then determined by the least-squares estimation (LSE) method. In the backward pass, the consequent parameters are fixed and the backpropagation of the committed error is done to adjust the premise parameters using the iterative gradient descent method. The subsection (5.4) describes a use case of the ANFIS architecture to design a reactive navigation controller for mobile robot.

Several researchers have adopted this architecture to solve the reactive mobile robot navigation problem. In [18], the author proposed a reactive navigation controller based on a standard ANFIS, i.e., the adjustment of the adaptive network's parameters is performed by the same learning algorithms (least squares estimation and gradient descent) as those used by Jang [36]. But in [25]–[27] the authors proposed reactive navigation controllers based on variants of ANFIS which are CS-ANFIS, IWO-ANFIS, and TLBO-ANFIS. They adopted population-based metaheuristics to adjust the premise parameters. This choice is made to overcome the complexity problem and improve the accuracy of the model [25]. For this fact, the author of [25] used the Cuckoo search (CS) metaheuristic that was introduced by Yang and Deb in 2009 [57], the author of [26] used the Invasive weed Optimization (IWO) metaheuristic that was introduced by Mehrabian and Lucas in 2006 [58], and the author of [27] used the Teaching-Learning-based Optimization (TLBO) metaheuristic that was introduced by Rao and al. in 2011 [59].

These metaheuristics are optimization techniques that consist in generating a population of solutions in the search space. At each iteration, the metaheuristic uses two strategies, an exploration strategy that allows exploring the search space to find the best solutions for minimizing the objective function and an exploitation strategy that changes the values of the candidate solutions by using a formula according to the adopted metaheuristic. Thus, this principle overcomes the problem of complexity and local minima that can be encountered when solving some optimization problems.

To summarize, Table 2 shows the learning methods adopted to adjust the premise and consequent parameters in each ANFIS variant used to design the mobile robot navigation controllers presented above.

Table 2. Learning methods used in ANFIS variants for reactive navigation

Navigation controller	Learning method for premise parameters	Learning method for consequent parameters
Standard ANFIS [18]	Gradient descent algorithm	LSE
CS-ANFIS [25]	CS algorithm	LSE
IWO-ANFIS [26]	IWO algorithm	LSE
TLBO-ANFIS [27]	TLBO algorithm	LSE

5.4. Description of ANFIS architecture use case for reactive navigation

Assume that a mobile robot navigates through a workspace that contains obstacles and a target to reach. This robot is equipped with a sensor system that detects the distances (LOD, FOD and ROD) and the position of the target to determine the target's angle (TA). Thus, the ANFIS controller (Figure 9) receives four inputs (x, y, z, v) that respectively represent the measurements (LOD, FOD, ROD, TA) coming from the robot's sensors to return as output a value that represents the steering angle (SA). This system is based on the Takagi-Sugeno fuzzy model [37]. So, each rule of the fuzzy system is expressed in the following form:

Rule i: If x is X and y is Y and z is Z and v is V Then fi = pi x + qi y + ri z + si v + ti

This controller consists of five layers, each of them is related to a phase of the fuzzy inference system. Table 3 describes the layers of this ANFIS controller.

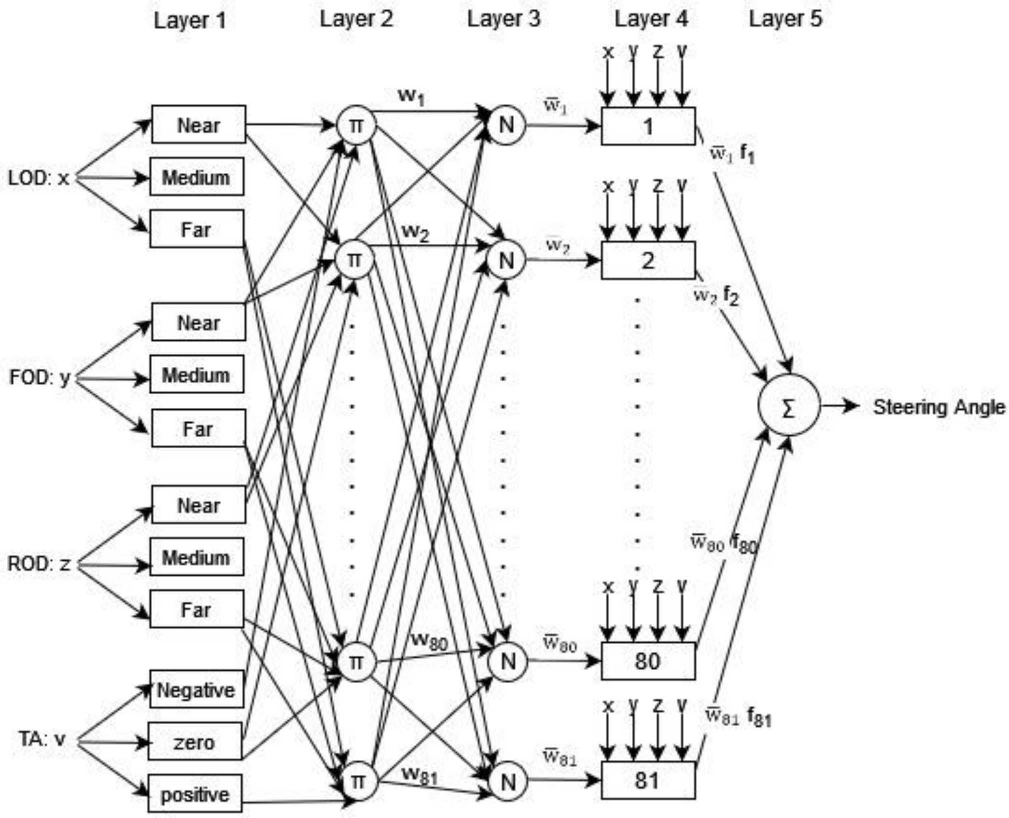

Figure 9. ANFIS based mobile robot navigation controller

Table 3. Description of ANFIS layers

Layer	Description	Node function
1	It represents the fuzzification phase. Each node computes the membership degree u_{A_i} of input to the linguistic terms A_i. Several function shapes are used (triangular, trapezoidal, …), but the bell-shaped function is more general. This function is related to the premise parameters (a_i, b_i , c_i). So, the nodes are adaptive (square shape).	$O_i^1 = u_{A_i}(x) = \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}}$
2	It represents the conjunction operator applied on the premise part. It expresses the firing strength w_i of each rule. Several T-norm operators are used for this fact. We can use the product operator. The nodes are not adaptive (circle shape). The number of nodes in this layer is the number of possible combinations of the linguistic terms defined in the previous layer (3*3*3*3 = 81 nodes)	$O_i^2 = w_i = u_{A_i} \times u_{B_j} \times u_{C_k}$
3	it is a normalization layer. Each node consists in normalizing the firing strength of a fuzzy rule. The nodes are not adaptive (circle shape).	$O_i^3 = \bar{w}_i = \frac{w_i}{\sum_{n=1}^{81} w_n}$
4	Each node of this layer represents the consequent part of a fuzzy rule. It determines the rule contribution in the overall output. According to Takagi-Sugeno's fuzzy model, it is expressed as a linear function with the parameters p_i, q_i, r_i, s_i, t_i. So, the nodes of this layer are adaptive (square shape).	$O_i^4 = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i z + s_i v + t_i)$
5	This layer is represented by a single circular node. It consists in computing the overall output O_i^5 that represents the steering angle (SA) of the robot based on the defuzzification formula.	$O_i^5 = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$

6. DISCUSSION

Table 4. Analysis of various mobile robot navigation controllers based on neuro-fuzzy architectures

Paper	Neuro-fuzzy architecture	Added Method	year	Input	Output	Dynamic target	Dynamic obstacle	Escape U-shaped obstacles	Simulation result	Real-time result
[10]	Cascaded neuro-fuzzy		2006	LOD, FOD, ROD, TA	LWV, RWV	N	N	N	Y	Y
[11]			2008	LOD, FOD, ROD, TA	LWV, RWV	N	N	N	Y	Y
[12]			2011	LOD, FOD, ROD, TA	LWV, RWV	N	N	N	Y	N
[13]	Adaptive neuro-fuzzy (type-2)		2006	Distance, Angle	SA, Acceleration	N	Y	N	Y	N
[14]		State Memorizing Strategy	2007	LOD, FOD, ROD, TA, Velocity	LWA, RWA	Y	Y	Y	Y	N
[15]			2012	8 Distances	Position	N	N	N	Y	N
[16]			2019	LOD, FOD, ROD, TA, Velocity, Target-Distance	SA, Injected-Velocity	N	N	N	Y	N
[17]	ANFIS		2003	Distance, Velocity	SA	N	N	N	Y	N
[18]			2015	LOD, FOD, ROD, TA	SA	N	Y	N	Y	Y
[19]			2016	LOD, FOD, ROD, TA	SA	N	N	N	Y	Y
[20]	Multi-ANFIS		2014	LOD, FOD, ROD, TA	LWV, RWV	N	N	N	Y	Y
[21]			2014	LOD, FOD, ROD, TA	LWV, RWV	N	N	N	Y	N
[22]			2018	LOD, FOD, ROD	LWV, RWV	N	Y	N	Y	Y
[23]			2019	LOD, FOD, ROD	LWV, RWV	N	N	N	Y	N
[24]			2022	LOD, FOD, ROD, TA	LWV, RWV	N	Y	N	Y	N
[25]	CS-ANFIS		2015	LOD, FOD, ROD, TA	SA	N	N	N	Y	Y
[26]	IWO-ANFIS		2015	LOD, FOD, ROD, TA, LWV, RWV	SA	N	N	N	Y	Y
[27]	TLBO-ANFIS		2018		SA	N	N	N	Y	N
[28]	AKH-NFIS		2021	LOD, FOD, ROD, TA	LWV, RWV	N	N	N	Y	N
[29]	ANFIS/PSO		2022	LOD, FOD, ROD, TA	SA	N	N	N	Y	N
[30]	ANFIS + Method	Virtual target strategy	2017	LOD, FOD, ROD, TA	Velocity difference	N	N	Y	Y	N
[31]		safe boundary algorithm	2017	LOD, FOD, ROD, TA	LWV, RWV	N	N	N	Y	Y
[32]		Q-Learning	2019	LOD, FOD, ROD, TA	SA	N	N	Y	Y	N
[33]		GPS based-method	2020	LOD, FOD, ROD	SA	N	N	N	Y	N
[34]		Utility-function	2021	LOD, FOD, ROD	SA	N	N	N	Y	N
[35]		GPS based-method	2022	LOD, FOD, ROD	SA	N	N	N	Y	N

In this paper, we focused on the neuro-fuzzy combinations used to develop reactive navigation controllers for mobile robot. Indeed, we explored the cascaded neuro-fuzzy architecture [10]–[12], the adaptive neuro-fuzzy architecture based on type-2 fuzzy inference system [13]–[16], and the adaptive neuro-fuzzy architecture based on type-3 fuzzy inference system with the variants: ANFIS, CS-ANFIS, IWO-ANFIS, and TLBO-ANFIS adopted respectively in [18], [25], [26], and [27].

The cascaded neuro-fuzzy architecture uses in the first stage a neural network as a preprocessor of sensor information. But in the second stage, uses a fuzzy inference system that controls the robot's movement based on three behaviors which are target seeking, obstacle avoidance, and wall following. This combination was successful in avoiding the instability of trajectories generated by controllers that were based only on the neural approach. However, the design of a fuzzy inference system is difficult due to the lack of a systematic method for generating fuzzy rules and adjusting the parameters of membership functions.

The adaptive neuro-fuzzy architecture designed in [14] is based on type-2 fuzzy inference system. That is, fuzzy sets are involved in the premise and consequent parts of the fuzzy rules. This architecture automatically adjusts the parameters of membership functions and keeps the physical meaning of the variables and parameters of the fuzzy system during the processing. In addition, it can provide multiple outputs instead of only one in the case of ANFIS architecture, which improves the control capability. But the choice of the fuzzy sets in the consequent part of each rule is done using human expertise to eliminate irrelevant combinations, reduce the number of rules, and minimize the computation time. Despite this, it generates a base of forty-eight fuzzy rules in [14]. For this reason, the author developed another algorithm to reduce the number of rules based on the similarity computed using the Euclidean distance between the rule’s vectors.

Another architecture explored in this paper is the adaptive network-based type-3 fuzzy inference system (ANFIS). It is based on a hybrid learning strategy, i.e., the premise parameters are adjusted using the gradient descent method, while the consequent parameters are determined using the Least Square Estimation (LSE) method. Thus, the author of [18] adopted this architecture by keeping the same learning algorithms (gradient descent and least squares estimation) used in the standard ANFIS. Whereas the authors of [25]–[27] adopted variants of the ANFIS architecture to develop a controller for reactive navigation. they replaced the gradient descent method with a metaheuristic in order to overcome the complexity problem and to improve the accuracy of the model. Indeed, the author of [25] adopted the cuckoo search (CS) metaheuristic, the author of [26] adopted the Invasive Weed Optimization (IWO) metaheuristic, and the author of [27] adopted the Teaching-Learning-based Optimization (TLBO) metaheuristic.

According to the literature, we can conclude that the ANFIS architecture is widely adopted by researchers to design autonomous navigation controllers for several reasons, such as the automatic adjustment of the parameters of the fuzzy inference system and the use of Sugeno's fuzzy model which makes it possible to adopt a hybrid learning strategy. Nevertheless, the ANFIS architecture has some limitations. It provides a single overall output which limits the capacity of the control system, it generates a large base of fuzzy rules based on the fuzzy sets of inputs which increases the network complexity and computational cost, and it does not allow designing a navigation controller able to make the robot escape from U-shaped obstacles (concave environment), without adding other methods, because it falls into the infinite loop problem. For this purpose, several works have been proposed to reduce these limitations. For example, the authors of [20]–[24] adopted multi-ANFIS-based controllers to have two outputs which are the left and right wheel velocities, the author of [30] designed an ANFIS-based controller with a virtual target strategy to escape infinite loop in concave environments, the author of [31] designed an ANFIS-based controller with an integrated safe boundary algorithm to avoid irregular-shaped obstacles, and the authors of [33]–[35] designed mobile robot navigation controllers using ANFIS for obstacle-avoidance and methods for target-seeking to reduce the number of rules in fuzzy inference systems.

To summarize, Table 4 presents a detailed analysis of mobile robot navigation controllers that are based on the neuro-fuzzy architectures studied in this paper. The performance of each controller is evaluated based on parameters such as input-output of the control system, navigation in environments with dynamic obstacles and targets, navigation in concave obstacles, and results in simulation and real-time environments.

7. CONCLUSION

Path planning methods for mobile robot navigation are classified in the literature into classical and reactive methods. Classical methods are widely used for navigation applications in previously known environments to find optimal paths to the target. Whereas, reactive methods have shown their effectiveness in unknown environments and the development of real-time navigation applications. In this study, we have focused on the neuro-fuzzy combination which is a reactive method with several architectures. Thus, we explored the cascaded neuro-fuzzy architecture, the adaptive neuro-fuzzy architecture (type-2) and the adaptive neuro-fuzzy architecture (type-3) with the variants: ANFIS, CS-ANFIS, IWO-ANFIS, and TLBO-ANFIS.

According to this survey, we affirm that the combination of the fuzzy and neural approaches has significantly improved the performance of mobile robot navigation controllers thanks to the self-learning capability of the neural network and the ability to process uncertain information by the fuzzy inference system. Furthermore, the use of the ANFIS architecture with metaheuristics has shown its effectiveness in the design of control systems such as the mobile robot navigation controllers for the following reasons:

- It allows generating the fuzzy rule base and automatically adjusting the parameters of membership functions.
- It allows optimizing the computation time in the learning phase by using the hybrid learning strategy because of the linearity of the consequent part of the fuzzy rules.
- It allows reducing convergence problems in the learning phase by using metaheuristics such as cuckoo search (CS), invasive weed optimization (IWO), and Teaching-Learning-Based Optimization (TLBO).
- It is suitable for the development of real-time applications of the control system.

Nevertheless, the performance of ANFIS-based mobile robot navigation controllers can be improved by integrating optimization methods for the rule base of the fuzzy inference system. According to the control system literature [60]–[63], we can minimize the fuzzy rule base either by eliminating rules according to a potentiality threshold (firing strength of rules) or by extracting significant fuzzy rules using data clustering algorithms. All this must be done while maintaining high controller accuracy.

REFERENCES

[1] Y.-Q. Qin, D.-B. Sun, N. Li, and Y.-G. Cen, ‘Path planning for mobile robot using the particle swarm optimization with mutation operator’, in Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826), Aug. 2004, pp. 2473–2478 vol.4. doi: 10.1109/ICMLC.2004.1382219.

[2] P. Raja and S. Pugazhenthi, ‘Optimal path planning of mobile robots: A review’, Int. J. Phys. Sci., vol. 7, no. 9, pp. 1314–1320, Feb. 2012, doi: 10.5897/IJPS11.1745.

[3] M. N. A. Wahab, S. Nefti-Meziani, and A. Atyabi, ‘A comparative review on mobile robot path planning: Classical or meta-heuristic methods?’, Annu. Rev. Control, vol. 50, pp. 233–252, 2020, doi: 10.1016/j.arcontrol.2020.10.001.

[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’, Def. Technol., vol. 15, no. 4, pp. 582–606, Aug. 2019, doi: 10.1016/j.dt.2019.04.011.

[5] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, ‘Heuristic approaches in robot path planning: A survey’, Robot. Auton. Syst., vol. 86, pp. 13–28, Dec. 2016, doi: 10.1016/j.robot.2016.08.001.

[6] A. Pandey, ‘Mobile Robot Navigation and Obstacle Avoidance Techniques: A Review’, Int. Robot. Autom. J., vol. 2, no. 3, May 2017, doi: 10.15406/iratj.2017.02.00023.

[7] M. Algabri, H. Mathkour, H. Ramdane, and M. Alsulaiman, ‘Comparative study of soft computing techniques for mobile robot navigation in an unknown environment’, Comput. Hum. Behav., vol. 50, pp. 42–56, Sep. 2015, doi: 10.1016/j.chb.2015.03.062.

[8] K.-T. Song and L.-H. Sheen, ‘Heuristic fuzzy-neuro network and its application to reactive navigation of a mobile robot’, Fuzzy Sets Syst., vol. 110, no. 3, pp. 331–340, Mar. 2000, doi: 10.1016/S0165-0114(97)00401-6.

[9] G. N. Marichal, L. Acosta, L. Moreno, J. A. Méndez, J. J. Rodrigo, and M. Sigut, ‘Obstacle avoidance for a mobile robot: A neuro-fuzzy approach’, Fuzzy Sets Syst., vol. 124, no. 2, pp. 171–179, Dec. 2001, doi: 10.1016/S0165-0114(00)00095-6.

[10] S. K. Pradhan, D. R. Parhi, and A. K. Panda, ‘Neuro-fuzzy technique for navigation of multiple mobile robots’, Fuzzy Optim. Decis. Mak., vol. 5, no. 3, pp. 255–288, Jul. 2006, doi: 10.1007/s10700-006-0014-z.

[11] D. R. Parhi, ‘Neuro-Fuzzy Navigation Technique for Control of Mobile Robots’, in Mobile Robot Motion Planning, X.-J. Jing, Ed., InTech, 2008, pp. 409–432. doi: 10.5772/6004.

[12] M. M. Joshi and M. A. Zaveri, ‘Reactive Navigation of Autonomous Mobile Robot Using Neuro- Fuzzy System’, Int. J. Robot. Autom., vol. 2, no. 3, pp. 128–145, 2011.

[13] N. B. Hui, V. Mahendar, and D. K. Pratihar, ‘Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches’, Fuzzy Sets Syst., vol. 157, no. 16, pp. 2171–2204, Aug. 2006, doi: 10.1016/j.fss.2006.04.004.

[14] A. Zhu and S. X. Yang, ‘Neurofuzzy-Based Approach to Mobile Robot Navigation in Unknown Environments’, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., vol. 37, no. 4, pp. 610–621, Jul. 2007, doi: 10.1109/TSMCC.2007.897499.

[15] D. Kumar and K. Dhama, ‘Neuro-fuzzy Control of an Intelligent Mobile Robot’, in 2012 Second International Conference on Advanced Computing & Communication Technologies, Rohtak,Haryana, India: IEEE, Jan. 2012, pp. 106–111. doi: 10.1109/ACCT.2012.77.

[16] S. Brahimi, O. Azouaoui, and M. Loudini, ‘Intelligent mobile robot navigation using a neuro-fuzzy approach’, Int J Comput. Aided Eng. Technol., vol. 11, no. 6, pp. 710–726, 2019.

[17] P. Rusu, E. M. Petriu, T. E. Whalen, A. Cornell, and H. J. W. Spoelder, ‘Behavior-based neuro-fuzzy controller for mobile robot navigation’, IEEE Trans. Instrum. Meas., vol. 52, no. 4, pp. 1335–1340, Aug. 2003, doi: 10.1109/TIM.2003.816846.

[18] J. K. Pothal and D. R. Parhi, ‘Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system’, Robot. Auton. Syst., vol. 72, pp. 48–58, Oct. 2015, doi: 10.1016/j.robot.2015.04.007.

[19] A. Pandey, S. Kumar, K. K. Pandey, and D. R. Parhi, ‘Mobile robot navigation in unknown static environments using ANFIS controller’, Perspect. Sci., vol. 8, pp. 421–423, Sep. 2016, doi: 10.1016/j.pisc.2016.04.094.

[20] M. Algabri, H. Mathkour, and H. Ramdane, 'Mobile Robot Navigation and Obstacle-avoidance using ANFIS in Unknown Environment', Int. J. Comput. Appl., vol. 91, no. 14, pp. 36–41, Apr. 2014, doi: 10.5120/15952-5400.

[21] A. Al-Mayyahi, W. Wang, and P. Birch, 'Adaptive Neuro-Fuzzy Technique for Autonomous Ground Vehicle Navigation', Robotics, vol. 3, no. 4, pp. 349–370, Nov. 2014, doi: 10.3390/robotics3040349.

[22] A. Pandey, A. K. Kashyap, D. R. Parhi, and B. K. Patle, 'Autonomous mobile robot navigation between static and dynamic obstacles using multiple ANFIS architecture', World J. Eng., vol. 16, no. 2, pp. 275–286, Apr. 2019, doi: 10.1108/WJE-03-2018-0092.

[23] P. Subbash and K. T. Chong, 'Adaptive network fuzzy inference system based navigation controller for mobile robot', Front. Inf. Technol. Electron. Eng., vol. 20, no. 2, pp. 141–151, Feb. 2019, doi: 10.1631/FITEE.1700206.

[24] D. Patel and K. Cohen, 'Obstacle Avoidance and Target Tracking by Two Wheeled Differential Drive Mobile Robot Using ANFIS in Static and Dynamic Environment', in Fuzzy Information Processing 2020, B. Bede, M. Ceberio, M. De Cock, and V. Kreinovich, Eds., in Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2022, pp. 337–348. doi: 10.1007/978-3-030-81561-5_28.

[25] P. K. Mohanty and D. R. Parhi, 'A new hybrid optimization algorithm for multiple mobile robots navigation based on the CS-ANFIS approach', Memetic Comput., vol. 7, no. 4, pp. 255–273, Dec. 2015, doi: 10.1007/s12293-015-0160-3.

[26] D. R. Parhi and P. K. Mohanty, 'IWO-based adaptive neuro-fuzzy controller for mobile robot navigation in cluttered environments', Int. J. Adv. Manuf. Technol., vol. 83, no. 9–12, pp. 1607–1625, Apr. 2016, doi: 10.1007/s00170-015-7512-5.

[27] A. Aouf, L. Boussaid, and A. Sakly, 'TLBO-Based Adaptive Neurofuzzy Controller for Mobile Robot Navigation in a Strange Environment', Comput. Intell. Neurosci., vol. 2018, pp. 1–8, 2018, doi: 10.1155/2018/3145436.

[28] M. S. Das, A. Samanta, S. Sanyal, and S. Mandal, 'AKH-NFIS: Adaptive Krill Herd Network Fuzzy Inference System for Mobile Robot Navigation', Wirel. Pers. Commun., vol. 120, no. 4, pp. 3389–3413, Oct. 2021, doi: 10.1007/s11277-021-08619-5.

[29] M. Lazreg and N. Benamrane, 'Hybrid system for optimizing the robot mobile navigation using ANFIS and PSO', Robot. Auton. Syst., vol. 153, p. 104114, Jul. 2022, doi: 10.1016/j.robot.2022.104114.

[30] J. Lu and U. Kintak, 'Mobile robot navigation based on adaptive neuro-fuzzy inerence ssystem with virtual target strategy', in 2017 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Jul. 2017, pp. 132–136. doi: 10.1109/ICWAPR.2017.8076677.

[31] A. M. Rao, K. Ramji, B. S. K. Sundara Siva Rao, V. Vasu, and C. Puneeth, 'Navigation of non-holonomic mobile robot using neuro-fuzzy logic with integrated safe boundary algorithm', Int. J. Autom. Comput., vol. 14, no. 3, pp. 285–294, Jun. 2017, doi: 10.1007/s11633-016-1042-y.

[32] L. Song, D. Li, and Z. Sun, 'Research on Local Path Planning for the Mobile Robot Based on QL-anfis Algorithm', in 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Dec. 2019, pp. 141–148. doi: 10.1109/SSCI44817.2019.9003062.

[33] M. S. Gharajeh and H. B. Jond, 'Hybrid Global Positioning System-Adaptive Neuro-Fuzzy Inference System based autonomous mobile robot navigation', Robot. Auton. Syst., vol. 134, p. 103669, Dec. 2020, doi: 10.1016/j.robot.2020.103669.

[34] M. Samadi Gharajeh and H. B. Jond, 'An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system', Ain Shams Eng. J., vol. 13, no. 1, p. 101491, Jan. 2022, doi: 10.1016/j.asej.2021.05.005.

[35] M. H. Haider et al., 'Robust mobile robot navigation in cluttered environments based on hybrid adaptive neuro-fuzzy inference and sensor fusion', J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 10, pp. 9060–9070, Nov. 2022, doi: 10.1016/j.jksuci.2022.08.031.

[36] J.-S. R. Jang, 'ANFIS: adaptive-network-based fuzzy inference system', IEEE Trans. Syst. Man Cybern., vol. 23, no. 3, pp. 665–685, Jun. 1993, doi: 10.1109/21.256541.

[37] T. Takagi and M. Sugeno, 'Derivation of Fuzzy Control Rules from Human Operator's Control Actions', IFAC Proc. Vol., vol. 16, no. 13, pp. 55–60, Jul. 1983, doi: 10.1016/S1474-6670(17)62005-6.

[38] A.-M. Zou, Z.-G. Hou, S.-Y. Fu, and M. Tan, 'Neural Networks for Mobile Robot Navigation: A Survey', in Advances in Neural Networks - ISNN 2006, J. Wang, Z. Yi, J. M. Zurada, B.-L. Lu, and H. Yin, Eds., in Lecture Notes in Computer Science, vol. 3972. Springer, Berlin, Heidelberg: Springer, 2006, pp. 1218–1226. doi: 10.1007/11760023_177.

[39] S. X. Yang and M. Meng, 'An efficient neural network approach to dynamic robot motion planning', Neural Netw., vol. 13, no. 2, pp. 143–148, Mar. 2000, doi: 10.1016/S0893-6080(99)00103-3.

[40] D. Janglova, 'Neural Networks in Mobile Robot Motion', Int. J. Adv. Robot. Syst., vol. 1, no. 1, p. 8, Mar. 2004, doi: 10.5772/5615.

[41] I. Engedy and G. Horvath, 'Artificial neural network based mobile robot navigation', in 2009 IEEE International Symposium on Intelligent Signal Processing, Budapest, Hungary: IEEE, Aug. 2009, pp. 241–246. doi: 10.1109/WISP.2009.5286557.

[42] D. R. Parhi and M. K. Singh, 'Real-time navigational control of mobile robots using an artificial neural network', Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci., vol. 223, no. 7, pp. 1713–1725, Jul. 2009, doi: 10.1243/09544062JMES1410.

[43] M. K. Singh and D. R. Parhi, 'Path optimisation of a mobile robot using an artificial neural network controller', Int. J. Syst. Sci., vol. 42, no. 1, pp. 107–120, Jan. 2011, doi: 10.1080/00207720903470155.

[44] M. Al-Sagban and R. Dhaouadi, 'Neural-based navigation of a differential-drive mobile robot', in 2012 12th International Conference on Control Automation Robotics & Vision (ICARCV), Guangzhou, China: IEEE, Dec. 2012, pp. 353–358. doi: 10.1109/ICARCV.2012.6485184.
[45] S. H. Dezfoulian, D. Wu, and I. S. Ahmad, 'A Generalized Neural Network Approach to Mobile Robot Navigation and Obstacle Avoidance', in Intelligent Autonomous Systems 12, S. Lee, H. Cho, K.-J. Yoon, and J. Lee, Eds., in Advances in Intelligent Systems and Computing, vol. 193. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 25–42. doi: 10.1007/978-3-642-33926-4_3.
[46] X. Chen and Y. Li, 'Smooth formation navigation of multiple mobile robots for avoiding moving obstacles', Int. J. Control Autom. Syst., vol. 4, no. 4, pp. 466–479, Aug. 2006.
[47] Anmin Zhu and S. X. Yang, 'A fuzzy logic approach to reactive navigation of behavior-based mobile robots', in IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, New Orleans, LA, USA: IEEE, 2004, pp. 5045-5050 Vol.5. doi: 10.1109/ROBOT.2004.1302517.
[48] F. Abdessemed et al., 'A Hierarchical Fuzzy Control Design for Indoor Mobile Robot', Int. J. Adv. Robot. Syst., vol. 11, no. 3, p. 33, Mar. 2014, doi: 10.5772/57434.
[49] T. Sai, D. Nakhaeinia, and B. Karasfi, 'Application of Fuzzy Logic in Mobile Robot Navigation', in Fuzzy Logic - Controls, Concepts, Theories and Applications, E. Dadios, Ed., InTech, 2012. doi: 10.5772/36358.
[50] H. Chang and T. Jin, 'Command Fusion Based Fuzzy Controller Design for Moving Obstacle Avoidance of Mobile Robot', in Future Information Communication Technology and Applications, H.-K. Jung, J. T. Kim, T. Sahama, and C.-H. Yang, Eds., in Lecture Notes in Electrical Engineering, vol. 235. Dordrecht: Springer Netherlands, 2013, pp. 905–913. doi: 10.1007/978-94-007-6516-0_99.
[51] J. H. Lilly, 'Evolution of a Negative-Rule Fuzzy Obstacle Avoidance Controller for an Autonomous Vehicle', IEEE Trans. Fuzzy Syst., vol. 15, no. 4, pp. 718–728, Aug. 2007, doi: 10.1109/TFUZZ.2006.889918.
[52] V. M. Peri and D. Simon, 'Fuzzy Logic Control for an Autonomous Robot', in NAFIPS 2005 - 2005 Annual Meeting of the North American Fuzzy Information Processing Society, Detroit, MI, USA: IEEE, 2005, pp. 337–342. doi: 10.1109/NAFIPS.2005.1548558.
[53] Wei Li, 'Fuzzy logic-based "perception-action" behavior control of a mobile robot in uncertain environments', in Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference, Orlando, FL, USA: IEEE, 1994, pp. 1626–1631. doi: 10.1109/FUZZY.1994.343939.
[54] Y. Yan and Y. Li, 'Mobile robot autonomous path planning based on fuzzy logic and filter smoothing in dynamic environment', in 2016 12th World Congress on Intelligent Control and Automation (WCICA), Jun. 2016, pp. 1479–1484. doi: 10.1109/WCICA.2016.7578767.
[55] N. Yousfi, C. Rekik, M. Jallouli, and N. Derbel, 'Optimized fuzzy controller for mobile robot navigation in a cluttered environment', in 2010 7th International Multi- Conference on Systems, Signals and Devices, Amman, Jordan: IEEE, Jun. 2010, pp. 1–7. doi: 10.1109/SSD.2010.5585508.
[56] L. A. Zadeh, 'Fuzzy sets', Inf. Control, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: 10.1016/S0019-9958(65)90241-X.
[57] X.-S. Yang and S. Deb, 'Cuckoo Search via Lévy flights', in 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Dec. 2009, pp. 210–214. doi: 10.1109/NABIC.2009.5393690.
[58] A. R. Mehrabian and C. Lucas, 'A novel numerical optimization algorithm inspired from weed colonization', Ecol. Inform., vol. 1, no. 4, pp. 355–366, Dec. 2006, doi: 10.1016/j.ecoinf.2006.07.003.
[59] R. V. Rao, V. J. Savsani, and D. P. Vakharia, 'Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems', Comput.-Aided Des., vol. 43, no. 3, pp. 303–315, Mar. 2011, doi: 10.1016/j.cad.2010.12.015.
[60] M. A. Riyadi, I. Setiawan, and A. Amir, 'EEG Multiclass Signal Classification Based on Subtractive Clustering-ANFIS and Wavelet Packet Decomposition', in 2021 International Conference on Electrical and Information Technology (IEIT), Sep. 2021, pp. 81–86. doi: 10.1109/IEIT53149.2021.9587407.
[61] K. Hussain and M. N. Mohd. Salleh, 'Analysis of techniques for ANFIS rule-base minimization and accuracy maximization', ARPN J. Eng. Appl. Sci., vol. 10, no. 20, pp. 9739–9746, 2015.
[62] S. Oladipo, Y. Sun, and A. Amole, 'Performance Evaluation of the Impact of Clustering Methods and Parameters on Adaptive Neuro-Fuzzy Inference System Models for Electricity Consumption Prediction during COVID-19', Energies, vol. 15, no. 21, p. 7863, Oct. 2022, doi: 10.3390/en15217863.
[63] S. Rajab, 'Handling interpretability issues in ANFIS using rule base simplification and constrained learning', Fuzzy Sets Syst., vol. 368, pp. 36–58, Aug. 2019, doi: 10.1016/j.fss.2018.11.010.