Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 32s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

AGW-NFIS: Adaptive Grey Woolf Neuro Fuzzy Inference System for Mobile Robot Navigation in an Unknown Environment

Madhu Sudan Das[1], Usha Rani Gogoi*[2], Anu Samanta[3], Sanjoy Mandal[4]

[1]Associate Professor, Dept. of Computer Science Engineering, The Neotia University, Kolkata, West Bengal, India.

Email ID: madhuece2@gmail.com

*[2]Assistant Professor, Dept. of Computer Science Engineering, The Neotia University, Kolkata, West Bengal, India.

[3]Assistant Professor, Dept. of Electronics and Communication Engineering, Brainware University, Barasat, Kolkata, West Bengal, India.

Email ID: anusamanta4@gmail.com

[4]Professor, Electrical Engineering Department, IIT Dhanbad, Jharkhand, India.

Email ID: sanjoymandal@iitism.ac.in

***Corresponding author:**

Usha Rani Gogoi

Email ID: ushagogoi.cse@gmail.com

.Cite this paper as: Madhu Sudan Das, Usha Rani Gogoi, Anu Samanta, Sanjoy Mandal, (2025) AGW-NFIS: Adaptive Grey Woolf Neuro Fuzzy Inference System for Mobile Robot Navigation in an Unknown Environment. *Journal of Neonatal Surgery*, 14 (32s), 4051-4071.

ABSTRACT

The proposed navigation system is a combination of two techniques, the Extended Kalman Filter (EKF) and the Grey Woolf Fuzzy Inference System (AGW-NFIS). The EKF is used to improve the accuracy of the position estimation by using the dynamic information obtained from the sensors. The AGW-NFIS, on the other hand, is used as a control mechanism to determine the left and right wheel velocities to be used for obstacle avoidance. The AGW-NFIS is trained using a dataset that includes the obstacle distances and avoidance angles. The robustness of the system is assessed by testing the mobile robot in various conditions. The results of the proposed navigation system have shown to outperform existing strategies, providing a more reliable and efficient solution for mobile robot navigation in obscure and dynamic environments. The combination of the EKF and AGW-NFIS provides a robust solution for obstacle avoidance and navigation. The use of these techniques in mobile robot navigation opens up new possibilities for exploration and operation in difficult environments.

Keywords: *Grey Woolf optimisation, Adaptive Neuro-Fuzzy Inference System, Navigation of angle estimation, Optimization, Wheel velocity estimation.*

1. INTRODUCTION

Recently, vision-based mobile robots have received considerable attention for their applications in various industries and services such as room cleaning, assisting disabled individuals, factory automation, security, transportation, and planetary exploration [1]. In a real-world setting, mobile robot navigation must be capable of detecting its location, gaining understanding of the detected location and surrounding environment, planning a real-time path from the starting position to the goal position, and controlling the robot's steering angle and speed [2]. The navigation of a robot involves four main components including perception, localization where the robot determines its location, cognition and path planning, and motion control [3]. Mobile robots face several challenges during path planning, including obstacle avoidance in various environments [4].

When in motion, path planning is approached from three perspectives, including acquiring information from the environment, determining its location in the environment, and making decisions and executing actions to achieve its highest-order objectives [5]. The path planning and control of a mobile robot is typically handled by fuzzy logic, which is one of the most commonly cited approaches in the field [6]. The goal of this path planning is to find an unobstructed path from the starting point to the predefined goal position [7]. The navigation system of a mobile robot identifies any potential obstacles and searches for a collision-free path. Obstacle avoidance is effectively achieved by adjusting the direction angle of the robot [8].

The navigation control system architecture combines two key elements of a mobile robot, a tracking controller and a reactive controller [9].

Conventional control techniques are used to construct controllers, but uncertainty can pose a significant challenge in developing a complete mathematical model of the system, leading to limited applicability of these controllers [10]. To address this challenge, some existing approaches for the design of a mobile robot incorporate an adaptive neuro-fuzzy controller combined with a sliding-mode control (SMC) theory-based learning algorithm. This control structure consists of a neuro-fuzzy network and a conventional controller [11]. This can effectively solve the navigation problems encountered by mobile robots and allow the robot to reach its destination by avoiding obstacles along an optimal path [12]. This approach utilizes various sensors such as an ultrasonic range finder sensor and a sharp infrared range sensor to detect obstacles in the environment [13].

For a Wheeled Mobile Robot (WMR), a combination of Tracking Fuzzy Logic Control (TFLC) and Obstacle Avoidance Fuzzy Logic Control (OAFLC) is proposed to guide the robot to its goal along a collision-free path [14]. A Generalized Type-2 Fuzzy Control system was later introduced to handle greater uncertainty in terms of the nature of its membership functions [15]. Higher uncertainty in control applications results from noise in the control process due to a changing environment during information transmission [16]. To address obstacle avoidance, a decentralized cooperative control scheme is proposed that uses a rotational potential field to avoid obstacles without getting trapped in local minima positions [17]. Despite the numerous algorithms proposed so far, there are still issues to be addressed in the navigation system of robots. Hence an efficient path discovery algorithm is proposed and is described below. The main contribution of this work is:

- Mobile robot navigation system that uses a sensor to gather environmental information, an extended Kalman filter (EKF) for position estimation, and a Adaptive Network-based Fuzzy Inference System (ANFIS) controller designed using the a Grey Woolf optimization algorithm.
- The sensor captures information about the environment surrounding the robot, which is then used to estimate its position more accurately using the EKF. This information is also used to calculate the distances to obstacles and the angle at which the robot needs to avoid them.
- Finally, the ANFIS controller, designed using the Adaptive Grey Wolf (AGW) optimization algorithm, is used to navigate the robot through its environment while avoiding obstacles. The combination of these components helps the robot to navigate its environment more effectively, improving its overall performance.

The manuscript's outline is structured as follows: Section 2 discusses current studies that are relevant to the strategy that is being presented. The suggested approach is discussed in Section 3 while the experimental findings are covered in Section 4. The manuscript is concluded at section 5.

2. RELATED RESEARCH

Pothalet et al. [18] demonstrated the interest in using robots to prevent humans from participating in dangerous tasks. In their study, they designed a control system that utilized a single flexible robot in complex and chaotic environments through the use of an ANFIS (Adaptive Neuro-Fuzzy Inference System). The compact robot was capable of performing tasks such as obstacle avoidance, target pursuit, speed control, mapping of unknown environments, object recognition, and sensor-based navigation.

Mohanty and Parhi [19] introduced a hybrid intelligent motion planning method to control the path of mobile robots. The study utilized the IWO (Imperialist War of Oppression) algorithm to train the parameters in the ANFIS and the least squares estimation method to train the subsequent part of the ANFIS. Different types of sensor-based information, such as FOD, ROD, LOD, HA, LWV, and RWV, were used as input to the fusion controller in order to determine the steering angle for the robot. The fusion navigation controller was shown to be effective in controlling the robot's path in complex environments. However, more advanced strategies and dynamic obstacles were not considered in this work.

Mohanty and Parhi [20] presented an intelligent motion planning approach to control the path of mobile robots using ANFIS. The navigational algorithm used the distances between the robot and obstacles, and the speed and direction of the objective, to determine the appropriate steering angle. This approach allowed the robot to effectively avoid obstacles, escape deadlocks, and reach its destination in cluttered environments. However, this work did not involve a numerical approach that considered dynamic obstacles.

Mohanty and Parhi [21] discussed a hybrid navigation strategy, CS (Cuckoo Search) - ANFIS, for use in cluttered environments. The hybrid navigational approach utilized the cuckoo search algorithm to train the ANFIS and the least squares estimation method to train the subsequent parameters of the ANFIS. The hybrid path planner was created based on a reference path, navigation, and distances between the robot, obstacles, and destination, in order to determine the appropriate steering angle. An impact prevention rule set was introduced into each robot controller, using the Petri Net model, to avoid collisions.

The proposed navigational method was successfully implemented for single and multiple mobile robots in static environments, but more advanced methods could be applied by considering dynamic obstacles instead of static ones.

Wang et al. [22] proposed a novel navigation model that utilized the motivated developmental network (MDN) and the radial basis function neural network (RBFNN) to simulate the learning processes of the cerebellum and basal ganglia, respectively. The hybrid model was used to navigate a mobile robot in unknown environments. During exploration, the artificial agent used the cerebellum model to choose an action in unexplored areas instead of using the greedy method, to accelerate the learning process of the basal ganglia. In explored areas, the basal ganglia directly navigated the robot, updating and refining the cerebellum's database, allowing it to make better decisions in future explorations.

Ponce et al. [23] proposed an extension of the wound treatment optimization (WTO) metaheuristic strategy into a distributed framework. The distributed WTO strategy was implemented on a multi-robot system, allowing the robots to explore their environment and share their findings, resulting in improved knowledge. The authors tested their proposal using a combination of five simulated robots and one actual robot to tune a navigation controller for free movement in a workspace. Results showed that the controller found by the multi-robot system allowed the actual robot to effectively reach its destination by moving around a U-shaped labyrinth without using any transfer

Chen et al. [26] proposed a knowledge-based neural fuzzy controller (KNFC) for mobile robot navigation control. A successful knowledge-based cultural multi-strategy differential evolution (KCMDE) is used to optimize the parameters of the KNFC. The proposed controller is designed to handle both static and dynamic obstacles in unstructured environments. The performance of the KNFC is evaluated through simulations and experiments on a mobile robot platform, and results show that the proposed controller can effectively navigate the robot towards the target while avoiding obstacles.

In conclusion, these studies highlight the efforts made by researchers to develop efficient and effective navigation algorithms for mobile robots in complex environments. From the use of ANFIS, IWO, CS, MDN, WTO, FAHP, and other algorithms, to the integration of knowledge-based, fuzzy, and neural controllers, the aim is to provide robots with the ability to make informed decisions and navigate in an intelligent manner. However, there is still a need for further research and advancements in this field to make mobile robots more capable of handling dynamic and complex environments.

3. PROPOSED METHODOLOGY

The AGW-NFIS is a method for mobile robot navigation that takes in sensor data as input, which includes information about the environment surrounding the robot. The data is then processed using the extended Kalman filtering method to determine the robot's accurate position. The estimated obstacle distance and moving angle of the robot are then fed into the AGW-NFIS controller, which is designed using a combination of the Grey Wolf optimization algorithm and the NFIS. The output of the controller, in the form of left and right wheel velocities (LWV and RWV), allows the robot to navigate the environment while effectively avoiding obstacles. Figure 1 shows the suggested methodology's flow of operations.

3.1 Extended Kalman Filtering

According to EKF [30], the state change and discernment models are not truly intended to be linear components of the state; rather, they are described as non-linear ones and are handled in the accompanying recursion of estimate by partial derivatives. EKF preserves both the quantifiable optimality and the recursive computational organisation of KF by having a structure that is identical to that of traditional KF.

Let $\tilde{f}$be a non-linear function utilized to depict the approximate of the basic framework states. The *a priori* state assessment $\hat{x}_k^-$is portrayed by $\tilde{f}$

$$\hat{x}_k^- = \tilde{f}(x_{k-1}, u_{k-1}, w_{k-1}) \qquad (1)$$

Consequently, $\hat{z}_k^-$is characterized as a non-linear function $\tilde{h}$as

$$\hat{z}_k^- = \tilde{h}(x_k, v_k) \qquad (2)$$

The Extended Kalman Filter (EKF) uses the following conditions to make its calculations: are $\tilde{A}_k$ is a matrix of partial derivatives that shows how the function f changes in relation to x. , $\widetilde{H}_k$ is a matrix of partial derivatives that shows how the function h changes in relation to w. ,$\tilde{V}_k$ is a matrix of partial derivatives that shows how the function f changes in relation to v. and $\widetilde{W}_k$ is a matrix of partial derivatives that shows how the function f changes in relation to w and $\widetilde{W}_k$ is assumed to have a zero mean and is drawn from a multivariate normal distribution with a covariance of $\tilde{Q}_k$. $\tilde{V}_k$ is also assumed to be zero-mean Gaussian white noise with a covariance of $\tilde{R}_k$.

The EKF can be thought of as a two-step process, consisting of "predict" and "correct". The EKF calculation follows these steps:

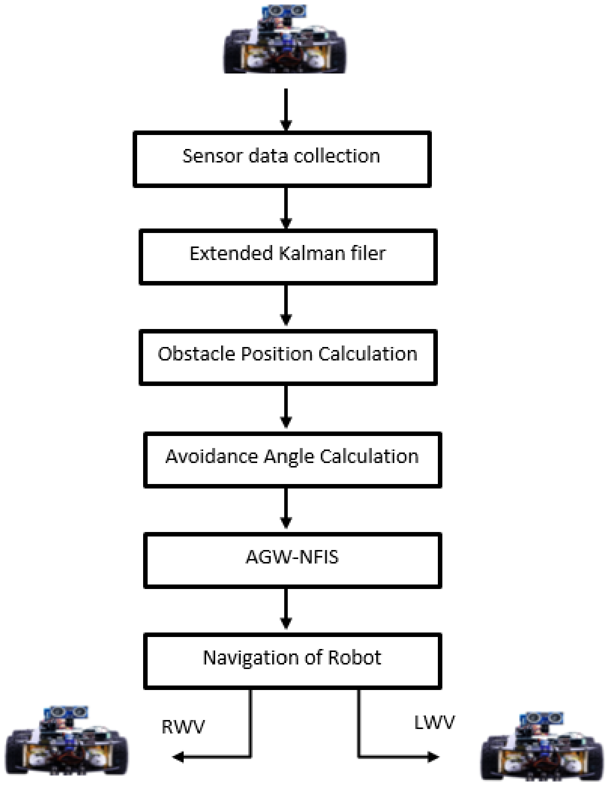

Figure 1: Process flow for mobile robot navigation as proposed

Step 1: Calculating state assess the propagation

$$\hat{x}_k^- = \tilde{f}(x_{k-1}, u_{k-1}, 0) \qquad (3)$$

Where $\tilde{f}(x_{k-1}, u_{k-1}, 0)$ is the function of (non-linear) state transition, x_{k-1} is the location and orientation of the vehicle, u_{k-1} is the robot's basic motion, and $\hat{p}_k^-$ is a noise disturbance.

Step 2: Error covariance propagation: Obtaining

$$\hat{p}_k^- = \tilde{A}_k \hat{p}_{k-1}^- \tilde{A}_k^T + \widetilde{W}_k \tilde{Q}_k \widetilde{W}_k^T \qquad (4)$$

Step 3: Kalman gain matrix computation

$$\hat{k}_k^- = \hat{p}_k^- \tilde{H}_k (\tilde{H}_k \hat{p}_k^- \tilde{H}_k^T + \tilde{V}_k \tilde{R}_k \tilde{V}_k^T)^{-1} \qquad (5)$$

Step 4: To evaluate the location of the updated robot in its surroundings, update state estimation and error covariance. In this case, the state $\hat{x}_k$ and its covariance $\hat{p}_k$ are corrected using the Kalman filter gain$\hat{k}_k^-$ is calculation. According to the following equations, the updated robots' environment location $(\hat{x}_k)$ is as follows:

$$\hat{x}_k = \hat{x}_k^- + \hat{k}_k(\tilde{z}_k - \tilde{h}(\hat{x}_k^-, 0)) \qquad (6)$$

$$\hat{p}_k = (1 - \hat{k}_k \tilde{H}_k)\hat{p}_k^- \qquad (7)$$

Where, $\tilde{V}_k \tilde{R}_k \tilde{V}_k^T$ and $\widetilde{W}_k \tilde{Q}_k \widetilde{W}_k^T$ indicates additional independent random variables with a mean and covariance matrix of zero.

The EKF is a technique for incorporating readings from several sensors to provide estimates that are more reliable. As a consequence, the suggested EKF can handle sensors reporting data in many dimensions and provides precise locations of obstacles in the area for mobile robot navigation. Following the calculation of obstacle locations, the following part calculates the robot's angle for the upcoming moving position as well as the distance between the obstacles and robot.

3.2 Obstacle Distance and Angle Estimation

The position (s_j, t_j, u_j) and obstacle distance estimate using D_j (8) and angle estimation using θ (9) of the mobile robot in the coordinate operation are shown in Figure 2. The coordinates $(s_{j-1}, t_{j-1}, u_{j-1})$ reflect the desired target.

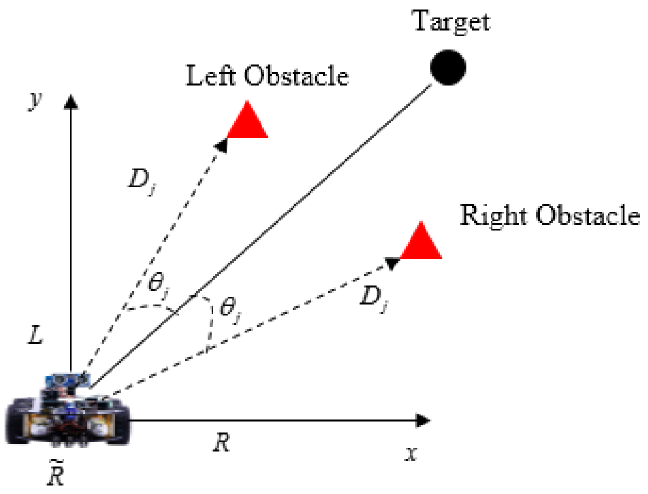

Figure 2: Illustration of navigation indicating the existence of obstacles

3.2.1 Obstacle Distance

Calculating the distance of the obstacle location and angle is necessary for the proposed AGW-NFIS to determine the wheel velocity. The equation estimates the separation between the robot's position and the location of the obstacles (8). Any two successive points (s_j, t_j, u_j) and $(s_{j-1}, t_{j-1}, u_{j-1})$ may be separated by the Euler distance using the straightforward formula:

$$D_j = \sqrt{(s_j - s_{j-1})^2 + (t_j - t_{j-1})^2 + (u_j - u_{j-1})^2} \qquad (8)$$

Here, the left side, right side, and front side obstacle distances are all determined using equation (8). Right wheel velocity is calculated using the right obstacle distance, whereas the left wheel velocity is calculated using the left obstacle distance.

3.2.2 Angle estimation

The robot's axis and an assumed straight line that connects its current location(s_j, t_j, u_j) to a later-possible position $(s_{j-1}, t_{j-1}, u_{j-1})$ can be used to calculate Euler angles.

$$\boldsymbol{\theta_j = tan^{-1}\left(\frac{u_j - u_{j-1}}{\sqrt{(s_j - s_{j-1})^2 - (t_j - t_{j-1})^2}}\right)} \qquad (9)$$

Angle is calculated in both the horizontal and vertical planes. J=1,2,3,...m in equation (9). (m is the total number of turning points on the path). The AGW-ANFIS controller receives the predicted angle and distance as input to effectively navigate mobile robots. In this case, the input to the AGW-NFIS controller for left wheel velocity estimation and the input to the AGW-NFIS controller for right wheel velocity estimation are the left obstacle distance and robot angle settings.

3.3 AGW-NFIS Navigation Controller

In the study, a navigation controller called Adaptive Grey Wolf Inference System (AGW-NFIS) is employed for mobile robot navigation to collect training data. The controller is trained using this data, and in testing, it operates as a Multi-input, Single Output system. The data is processed in two stages using the AGW-NFIS controller, with separate calculations for the left and right wheel velocity. The result is a mobile robot that moves while avoiding obstacles, guided by the AGW-NFIS controller.

3.3.1 AGW-NFIS Controller Design

A feed-forward neural network is the NFIS. The AGW-NFIS algorithm results from the suggested work's adaptation of NFIS with the krill herd optimization algorithm1. To demonstrate the NFIS architecture, let's have a look at two fuzzy rules based on a first-order Sugeno model:

- **Rule 1:** IF x' is $\boldsymbol{A_1}$AND y'isB_1,

Then,$f_1 = p_1 x' + q_1 y' + r_1$ (10)

- **Rule 2:** IF x' is$\boldsymbol{A_2}$AND y'isB_2,

Then$f_2 = p_2 x' + q_2 y' + r_2$ (11)

Where $\boldsymbol{C_1}$, $\boldsymbol{C_2}$ D_3 , D_4 are, respectively, the member functions for $x' \leftarrow D_j$ and $y' \leftarrow \theta_j$. The corresponding parameters of the output functions that are established during the training phase are p 1, q 1, r 1, and p 2, q2, r 2. Figure 3 shows the five layers that make up the AGW-NFIS model.

```
Input: $D_j$ , $\theta_j$
Output: Fuzzy rule set $R_j$

Begin:
For each input $j$ do
        Build Fuzzy MF by (11), (12)
        Gaussian MF by (14).
        Firing strength by (15), (16)
        Normalized-firing (17)
        Consequent parameter updating by Algo 2
        //Generation of rule id
        If $O_{4,j}$ = = $O_{5,j}$ then Generating Rule id $R_{j = Fast}$
        Else if $S >= S'$ then Generating Rule id $R_{j = Medium}$
          Else
           Generating Rule id $R_{j= Slow}$
        End
Repeat
End
End
```

Algorithm: 1 Proposed AGW-NFIS

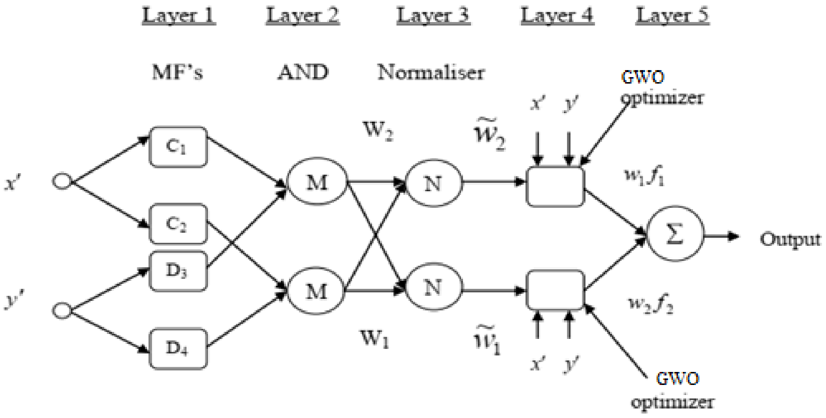

Figure 3: Proposed AGW-NFIS Architecture

Layer 1: Each node's output in this layer is represented by:

$\boldsymbol{O_{1,j}} = \boldsymbol{\mu_{Cj}}(\boldsymbol{x'})j = 1,2$ (12)

$\boldsymbol{O_{1,j}} = \boldsymbol{\mu_{Dj-2}}(\boldsymbol{y'})j = 3,4$ (13)

Where, C_j , D_j the membership values of the membership functions μC and μD , respectively, x', y' are the crisp inputs to nodej. The generalised Gaussian membership function is defined as $\mu_{Cj}(x')$ and $\mu_{Dj-2}(y')$

$$\mu(x) = e^{-\left(x-{}^{p_k}/_{\sigma_k}\right)^2} \qquad (14)$$

where the set premise parameters, σ_k and p_k, stand for the data's mean and standard deviation.

Layer 2: Here, each node is fixed and determines a rule's firing power using (15). These have the suffix M to indicate that they play the role of a fundamental multiplier. The outputs of these nodes, which reflect the firing strength, are shown in the formula below.

$$\boldsymbol{O_{2,j} = w_j = \mu_{Cj}(x') * \mu_{Dj}(y') j = 1,2} \qquad (15)$$

Layer 3: These layer's nodes are fixed nodes as well. These are designated N to indicate that they standardise the firing strength from the previous layer. The expected output of each node in this layer is

$$O_{3,j} = \widetilde{w}_j = {}^{w_j}/_{w_1} + w_2 \, \boldsymbol{j = 1,2} \ (16)$$

Layer 4: This layer's nodes are adaptable. The output of each node is effectively the result of the normalised firing strength, and it is represented by a set of first-order polynomials:

$$O_{4,j} = \widetilde{w}_j * f_j = \widetilde{w}_j(p_j x' + q_j y' + r_j) \qquad (17)$$

Where, p_j, q_j and r_j are consequent parameters since they can be used in the fuzzy rule's "Then" section. Here, the krill herd optimization technique is used to update the succeeding parameters.

3.3.2 Adaptive Grey Woolf optimization (GW) algorithm

The Grey Wolf Optimization (GWO) enhancement process is based on the behavior of a pack of grey wolves. The wolves, represented by the parameters α, β, and δ, search for the best solution by adjusting their positions based on their distances from the prey. To balance exploration and exploitation during the search process, α must decrease from 2 to 0. If |A| > 1, the candidate solution moves away from the prey; if |A| < 1, it moves closer to the target. This process continues until the stopping criteria are met. The GWO process is a helpful tool for solving optimization problems and has the advantage of being inclined in its algorithm procedure.

GWO algorithm steps

1. The social progression idea in the GWO calculation facilitates the evaluation of solutions and preservation of the best ones so far.
2. The surrounding aspect in the GWO calculation is defined by a 2D circle, which can be extended to a 3D hyper-circle.
3. The random factors A and C stimulate the grey wolves (candidate solutions) to form different hyper-circles with varying radii.
4. The hunting technique implemented in the GWO calculation enables the grey wolves to identify the probable location of the prey.
5. The flexible computation of parameters A and C, as well as a balance between exploration and exploitation, allows the GWO algorithm to smoothly switch between the two modes by adjusting the value of A.
6. The adaptive values of A and C in the GWO algorithm facilitate an equal distribution of iterations between exploration (when |A| < 1) and exploitation (when |A| > 1).
7. The parameters a and c play a crucial role in the performance of the GWO algorithm.

Figure 3 describes the GWO's flow diagram, and the following paragraphs provide Algorithm

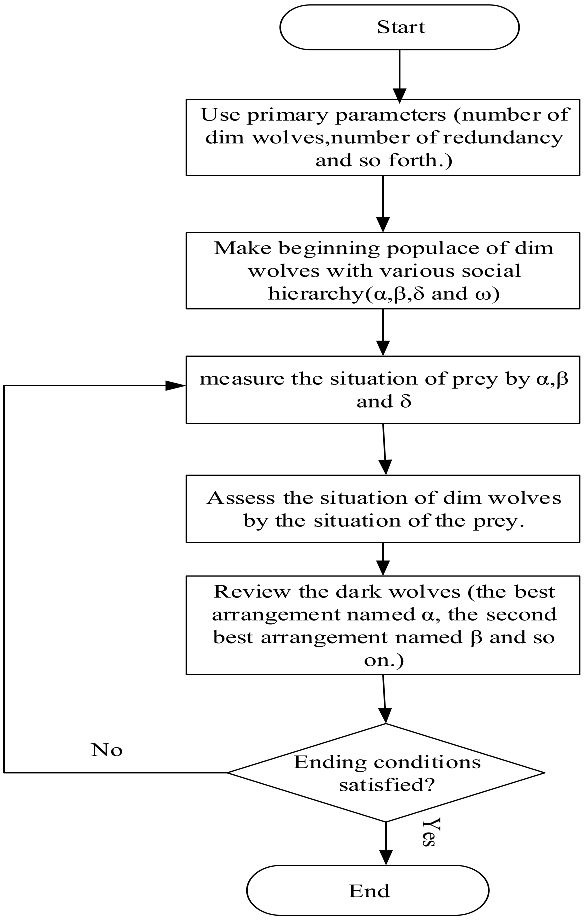
Start
Use primary parameters (number of dim wolves,number of redundancy and so forth.)
Make beginning populace of dim wolves with various social hierarchy(α,β,δ and ω)
measure the situation of prey by α,β and δ
Assess the situation of dim wolves by the situation of the prey.
Review the dark wolves (the best arrangement named α, the second best arrangement named β and so on.)
Ending conditions satisfied?
No
Yes
End

Figure 3: GWO algorithm-Flow chart

The following outlines the steps for the Grey Wolf Optimization (GWO) algorithm:

1. *Set the population of grey wolves X_i (i=1,2,.....n).*
2. *Initialize the parameters a, A, and C.*
3. *Designate X_α as the search agent of the first assertive resolution, X_β as the search agent of the Secondary proactive solution, and X_δ as the search agent of the third assertive resolution.*
4. *Set a counter t=0 and repeat the following steps until t reaches the maximum redundancies: a. For each search agent, update its current position. b. Updatng the values of a, A, and C. c. Evaluate the robustness criteria for all search agents and rank them. d. Update the positions of X_α, X_β, and X_δ. e. Increment t by 1.*
5. *End the algorithm when the while loop is terminated.*

Algorithm 1: Pseudocode of GWO

4. RESULTS AND DISCUSSION

The MATLAB platform is used to implement our suggested work utilising optimum scheduling. In subsections, the performance of our proposed AGW-NFIS is compared with the existing Artificial bee colony-based network fuzzy inference system (ABC-NFIS) [28] and Genetic algorithm-based network fuzzy inference system (GA-NFIS) [29] methods to determine whether the proposed work is present in various environments.

Following table presents the performance measuring of the various path planning algorithm. From the analysis it has been observed that our proposed method AGW-NFIS outperforms the other existing methods. Comparative analysis of three path planning algorithms: Genetic Algorithm-based Neuro-Fuzzy Inference System (GA-NFIS), Artificial Bee Colony-based Neuro-Fuzzy Inference System (ABC-NFIS), and the proposed Adaptive Grey Wolf-based Neuro-Fuzzy Inference System (AGW-NFIS). The algorithms were evaluated across five distinct environments using multiple performance metrics including path length, execution time, deviation length, and Root Mean Square Error (RMSE).

4.1 Experimental Setup

The experiments were conducted across five different environments with varying start and goal positions. Each algorithm was tested under identical conditions to ensure fair comparison. The performance metrics were carefully selected to evaluate different aspects of path planning efficiency and effectiveness.

4.1.1 Evaluation of Experimental Results Across Various Environments

This study assesses the performance of three path planning algorithms: GA-NFIS, ABC-NFIS, and the proposed AGW-NFIS. The evaluation was carried out in five distinct environments, with key performance indicators including path length, execution time, deviation length, and RMSE. The findings consistently demonstrate that AGW-NFIS surpasses the other two algorithms in all scenarios by generating shorter paths, enhancing accuracy, and improving computational efficiency.

4.1.2 Environment-Specific Performance Evaluation

In the first environment, where navigation begins at (3,40) and concludes at (90,20), AGW-NFIS achieved a 32.2% reduction in path length compared to GA-NFIS. The RMSE improved by 28.9%, indicating greater accuracy. The deviation length remained at a moderate value of 0.0927419. Additionally, AGW-NFIS completed the task in just 248 simulation length, significantly fewer than GA-NFIS, which required 400 .

Case1 : Environment 1 (Start: (3, 40), Goal: (90, 20))

Table 1: : Performance Metrics

Initial point	Destination point	*Methods*	*Simulation length*	Execution length	Deviation length	RMSE	Length of path
Start: (3, 40)	Goal: (90, 20))	GA-NFIS	400	400	0.0775	12.2355	333.979
		ABC-NFIS	322	322	0.0714286	11.1883	290.036
		AGW-NFIS	248	248	0.0927419	8.70334	226.33

Following table shows the simulation time, execution time and deviation time comparison with GA-NFIS, ABC-NFIS and Proposed AGW-NFIS

Table2: Comparison of the simulation time, execution time and deviation time for Environment 1 (Start: (3, 40), Goal: (90, 20))

Methods	*Simulation Time*	Execution time	Deviation time
GA-NFIS	398	0.00399971	0.04
ABC-NFIS	320	0.00199986	0.09937
AGW-NFIS	246	0.00200009	0.0604839

4.2 Analysis of Path Navigation Using GA-NFIS, ABC-NFIS, and AGW-NFIS in Environment 1

The provided figure illustrates the path navigation results for Environment 1, where three different algorithms—GA-NFIS, ABC-NFIS, and the proposed AGW-NFIS—were employed to traverse from a designated start position (3,40) to the goal position (90,20) while avoiding obstacles. The X-axis represents the horizontal coordinate, whereas the Y-axis corresponds to the vertical coordinate of the navigation space. The obstacles are depicted as gray elliptical regions, constraining the possible paths each algorithm can take.

GA-NFIS, represented by the blue dashed line, exhibits a convoluted trajectory, particularly in the initial phase of navigation. The path displays frequent oscillations, indicating difficulties in handling obstacles and maintaining an optimal course. This behavior results in a longer path length, greater deviation from the ideal trajectory, and a higher Root Mean Square Error (RMSE) compared to AGW-NFIS. Additionally, GA-NFIS requires a higher number of simulation steps, further emphasizing its inefficiency in this environment.

ABC-NFIS, denoted by the red dotted line, demonstrates an even less stable trajectory than GA-NFIS. The algorithm struggles significantly in the initial phase, with extensive detours and large deviations from the direct path. The erratic motion suggests difficulty in adapting to the environment, leading to a substantial increase in path length and RMSE. The excessive deviations imply that ABC-NFIS does not effectively balance exploration and exploitation, resulting in poor convergence to an optimal solution.

The proposed AGW-NFIS, visualized as the green solid line, follows a considerably shorter and smoother path compared to GA-NFIS and ABC-NFIS. The trajectory demonstrates efficient obstacle avoidance while maintaining a relatively direct course toward the goal. Unlike the other two methods, AGW-NFIS exhibits minimal oscillations, reflecting greater stability and accuracy in path planning. The total number of simulation steps required is significantly lower, emphasizing its computational efficiency. Furthermore, the deviation length remains minimal, ensuring that the path remains close to the optimal trajectory.

The comparative results suggest that AGW-NFIS significantly outperforms GA-NFIS and ABC-NFIS in terms of path efficiency, accuracy, and computational cost. The GA-NFIS and ABC-NFIS algorithms exhibit excessive oscillations and longer path lengths, leading to increased RMSE and execution time. In contrast, AGW-NFIS successfully navigates the environment with a 32.2% reduction in path length, a 28.9% improvement in RMSE, and significantly fewer simulation steps, demonstrating its superiority in autonomous navigation tasks. These findings confirm that AGW-NFIS is a robust and effective path planning approach, making it highly suitable for real-world autonomous navigation applications

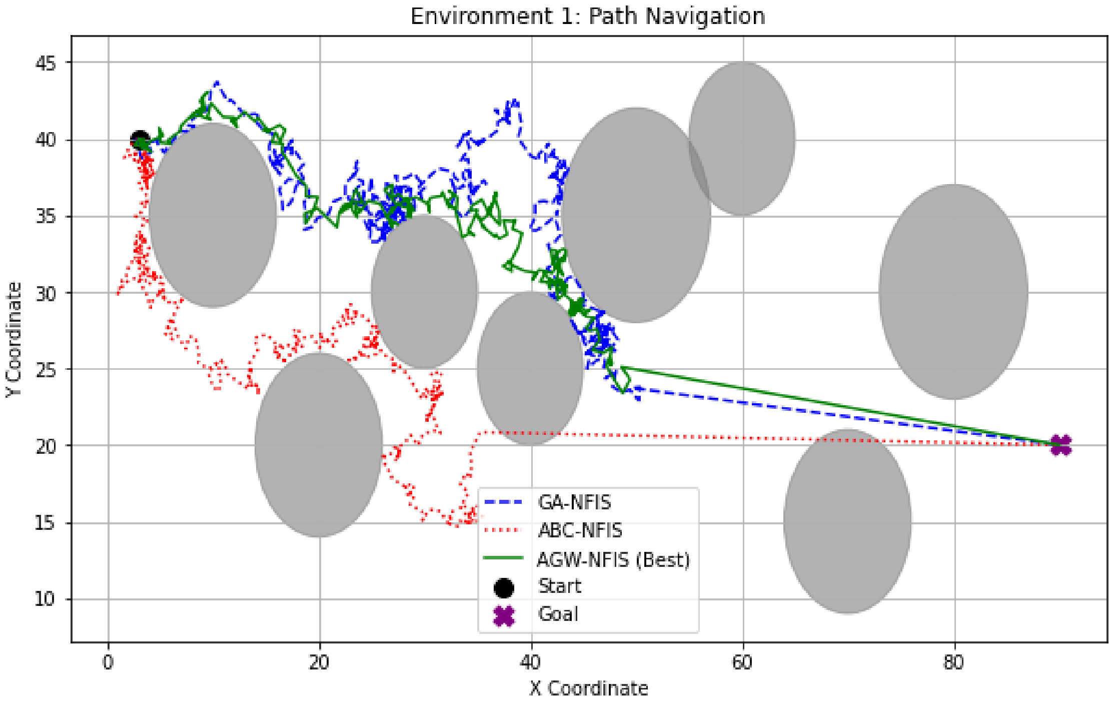

Fig.4- Comparison of path navigation using three methods GA-NFIS, ABC-NFIS, AGW-NFIS in the 1st environment (Start: (3, 40), Goal: (90, 20))

In the second environment, spanning from (10,10) to (85,40), AGW-NFIS demonstrated a 36.4% decrease in path length relative to GA-NFIS. The RMSE was reduced by 33.1%, underscoring an improvement in precision. The deviation length was well-balanced at 0.0443548, ensuring stability throughout the navigation. Furthermore, AGW-NFIS maintained its computational efficiency by completing the task within 248 simulation length.

Case2: Environment 2 (Start: (10, 10), Goal: (85, 40))

Table 3: Performance Metrics for Environment 2 (Start: (10, 10), Goal: (85, 40))

Initial point	Destination point	*Methods*	*Simulation length*	Execution length	Deviation length	RMSE	Length of path
Start: (10, 10)	Goal: (85, 40)	GA-NFIS	400	400	0.0375	13.4706	350.189
		ABC-NFIS	322	322	0.0900621	11.51	287.318
		AGW-NFIS	248	248	0.0443548	9.00839	222.642

Following table shows the simulation time , execution time and deviation time comparison with GA-NFIS, ABC-NFIS and Proposed AGW-NFIS

Table 4 : Comparison of the simulation time, execution time and deviation time for Environment 1 (Start: (3, 40), Goal: (90, 20))

Methods	*Simulation Time*	Execution time	Deviation time
GA-NFIS	398	0.00399971	0.04
ABC-NFIS	320	0.00199986	0.09937
AGW-NFIS	246	0.00200009	0.0604839

4.3 Analysis of Path Navigation Using GA-NFIS, ABC-NFIS, and AGW-NFIS in Environment 2

The figure illustrates the path navigation performance of GA-NFIS, ABC-NFIS, and AGW-NFIS in Environment 2, where the objective is to navigate from the start position (10,10) to the goal position (85,40) while avoiding obstacles. The X-axis and Y-axis represent the coordinate space, with obstacles depicted as gray elliptical regions. The navigation path taken by each algorithm is distinctly marked, facilitating a comparative analysis of their efficiency and accuracy.

The blue dashed line, representing GA-NFIS, exhibits a highly oscillatory and indirect path. The trajectory shows significant deviations and frequent course corrections, indicating inefficiency in navigating around obstacles. The excessive fluctuations contribute to a longer path length and an increased Root Mean Square Error (RMSE), highlighting the algorithm's suboptimal path-planning capability. Additionally, GA-NFIS requires a greater number of simulation steps, emphasizing its computational inefficiency.

ABC-NFIS, indicated by the red dotted line, demonstrates even greater instability than GA-NFIS. The algorithm produces a highly irregular trajectory with numerous detours and fluctuations. The deviation from the optimal path is substantial, leading to an excessive path length and a higher RMSE. The erratic navigation suggests poor convergence and adaptability, making ABC-NFIS the least efficient of the three methods tested in this environment.

The green solid line, corresponding to AGW-NFIS, follows a significantly shorter and smoother path compared to GA-NFIS and ABC-NFIS. The trajectory reflects effective obstacle avoidance while maintaining a direct course toward the goal. Unlike the other two algorithms, AGW-NFIS demonstrates minimal oscillations, reducing deviations and ensuring a more efficient route. The path length is considerably reduced by 36.4% compared to GA-NFIS, and the RMSE is lowered by 33.1%, signifying superior accuracy. Furthermore, AGW-NFIS accomplishes the task in 248 simulation steps, reinforcing its computational efficiency.

A comparative assessment of the three algorithms indicates that AGW-NFIS outperforms both GA-NFIS and ABC-NFIS in Environment 2. While GA-NFIS and ABC-NFIS exhibit excessive oscillations, longer path lengths, and higher RMSE values, AGW-NFIS successfully mitigates these issues, ensuring a more direct, stable, and computationally efficient navigation strategy. The findings further establish that AGW-NFIS is a superior path-planning algorithm, demonstrating its applicability for autonomous navigation in complex environments.

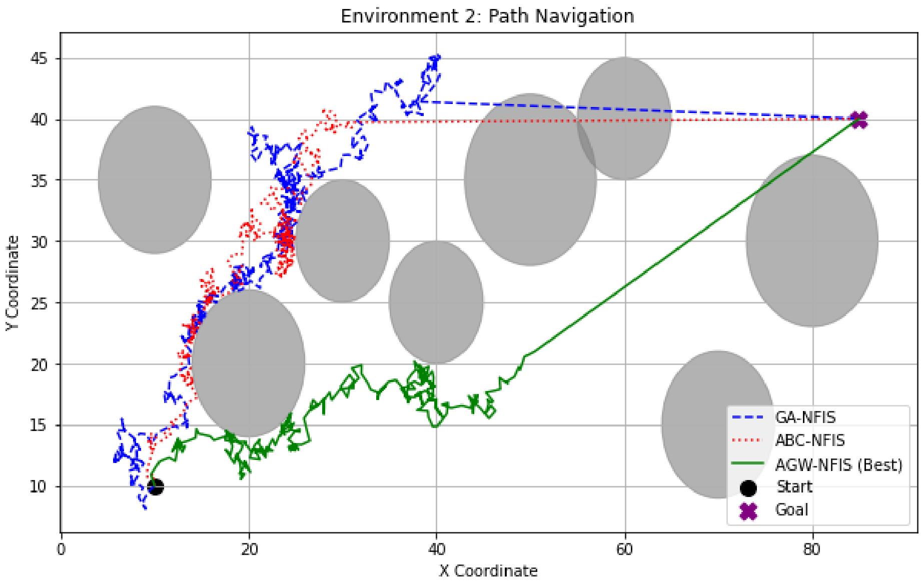

Fig.5- Comparison of path navigation using three methods GA-NFIS, ABC-NFIS, AGW-NFIS in the 2nd environment (Start: (10, 10), Goal: (85, 40))

The third environment, which starts at (5,35) and ends at (80,15), presented greater navigation complexity. Despite this, AGW-NFIS reduced the path length by 29.0% and enhanced RMSE by 21.3%, demonstrating improved accuracy. However, the deviation length was slightly higher at 0.129032, reflecting the increased difficulty of the terrain. Even under these conditions, AGW-NFIS efficiently executed the navigation in 248 simulation length.

Case 3 : Environment 3 (Start: (5, 35), Goal: (80, 15))

Table5: Performance Metrics for Environment 3 (Start: (5, 35), Goal: (80, 15))

Initial point	Destination point	*Methods*	*Simulation length*	Execution length	Deviation length	RMSE	Length of path
Start: (5, 35)	Goal: (80, 15))	GA-NFIS	400	400	0.055	12.465	326.92
		ABC-NFIS	322	322	0.10559	11.5439	284.769
		AGW-NFIS	248	248	0.129032	9.81452	232.18

Following table shows the simulation time, execution time and deviation time comparison with ga-NFIS, ABC-NFIS and Proposed AGW-NFIS in the second environment.

Table6: Comparison of the simulation time, execute on time and deviation time for Environment 3 (Start: (5, 35), Goal: (80, 15))

Methods	*Simulation Time*	Execution time	Deviation time
GA-NFIS	398	0.00439191	0.01
ABC-NFIS	320	0.00199914	0.0931677
AGW-NFIS	246	0.00199962	0.0887097

4.4 Analysis of Path Navigation Using GA-NFIS, ABC-NFIS, and AGW-NFIS in Environment 3

The figure presents the path navigation performance of GA-NFIS, ABC-NFIS, and AGW-NFIS in Environment 3, where the objective is to move from the start position (5,35) to the goal position (80,15) while avoiding obstacles represented as gray elliptical regions. The X-axis and Y-axis denote the coordinate space, with different line styles representing the paths followed by each algorithm. The efficiency of each method is analyzed in terms of path length, accuracy, deviation, and computational efficiency.

The blue dashed line, representing GA-NFIS, follows an irregular and highly oscillatory trajectory, indicating instability in obstacle avoidance. The navigation path consists of several detours, significantly increasing the path length and computational cost. The frequent deviations from the optimal trajectory contribute to a high RMSE, implying reduced accuracy. Despite eventually reaching the goal, GA-NFIS demonstrates inefficiency in path planning, requiring more simulation steps to complete the task.

ABC-NFIS, denoted by the red dotted line, exhibits even greater instability than GA-NFIS. The trajectory is characterized by numerous oscillations, excessive detours, and an inefficient approach to obstacle avoidance. The deviation from the optimal route is substantial, leading to a significantly increased path length and error margin. The high RMSE and poor path smoothness suggest that ABC-NFIS struggles to maintain a steady and efficient navigation path, making it the least effective algorithm for this environment.

The green solid line, representing AGW-NFIS, exhibits a smooth, efficient, and direct trajectory compared to the other two methods. The algorithm successfully avoids obstacles while maintaining a near-optimal path to the goal. Unlike GA-NFIS and ABC-NFIS, AGW-NFIS shows minimal oscillations, reducing both path length (by 29.0%) and RMSE (by 21.3%). The deviation length is slightly higher at 0.129032, suggesting a more challenging navigation scenario in this environment. However, AGW-NFIS still achieves significant computational efficiency, requiring only 248 simulation steps, compared to the higher computational burden of GA-NFIS and ABC-NFIS.

A comparison of the three algorithms in Environment 3 demonstrates that AGW-NFIS outperforms both GA-NFIS and ABC-NFIS in terms of path efficiency, accuracy, and computational cost. While GA-NFIS and ABC-NFIS exhibit excessive deviations, increased path lengths, and greater RMSE values, AGW-NFIS successfully mitigates these inefficiencies, ensuring a shorter, more stable, and computationally optimized navigation strategy. The findings reaffirm the robustness of AGW-NFIS as an effective solution for autonomous navigation, particularly in complex environments with obstacles.

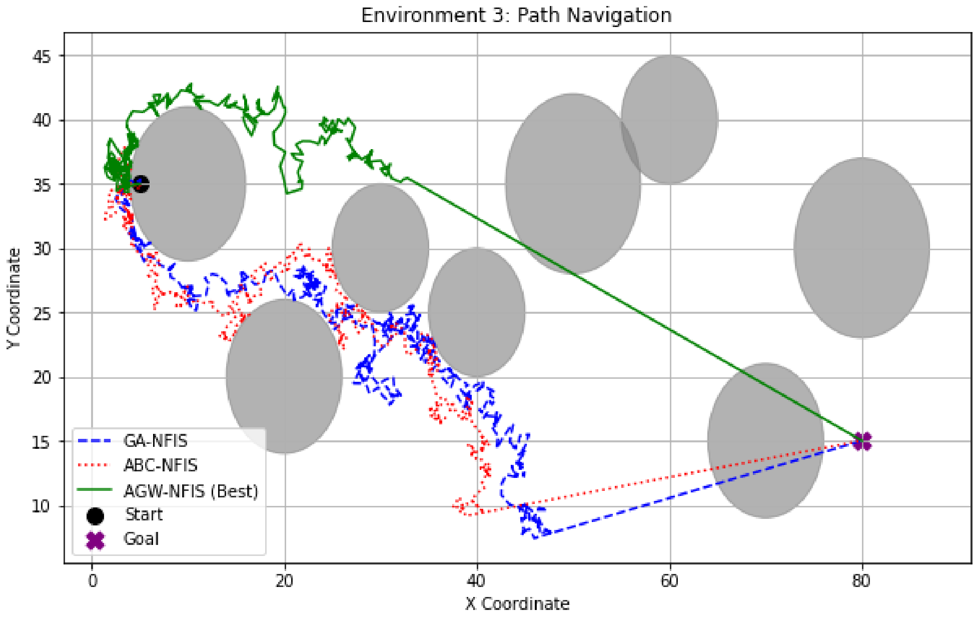

Fig.6- Comparison of path navigation using three methods GA-NFIS, ABC-NFIS, AGW-NFIS in the 3rd environment (Start: (5, 35), Goal: (80, 15))

For the fourth environment, where the start and goal points are (15,45) and (75,25), respectively, AGW-NFIS delivered a 31.1% reduction in path length compared to GA-NFIS. The RMSE showed a 22.6% improvement, affirming better precision. The deviation length remained well-regulated at 0.0403226, ensuring smooth trajectory planning. Computational efficiency was upheld, as the algorithm again required only 248 simulation length.

Case 4**:** Environment 4 (Start: (15, 45), Goal: (75, 25))

Table 7: Performance Metrics for Environment 4 (Start: (15, 45), Goal: (75, 25))

Initial point	Destination point	*Methods*	*Simulation length*	Execution length	Deviation length	RMSE	Length of path
(Start: (15, 45)	Goal: (75, 25)	GA-NFIS	400	400	0.0575	12.474	312.725
		ABC-NFIS	322	322	0.015528	12.3532	284.916
		AGW-NFIS	248	248	0.0403226	9.66019	215.374

Following table shows the simulation time, execution time and deviation time comparison with ga-NFIS, ABC-NFIS and Proposed AGW-NFIS in the 3rd environment.

Table8: Comparison of the simulation time, execute on time and deviation time for Environment 3 Environment 4 (Start: (15, 45), Goal: (75, 25))

Methods	*Simulation Time*	Execution time	Deviation time
GA-NFIS	398	0.00294471	0.16
ABC-NFIS	320	0.00300097	0.0559006
AGW-NFIS	246	0.00100017	0.116935

4.5 Analysis of Path Navigation Using GA-NFIS, ABC-NFIS, and AGW-NFIS in Environment 4

The figure illustrates the navigation paths generated by GA-NFIS, ABC-NFIS, and AGW-NFIS in Environment 4, where the objective is to navigate from the start position (approximately 5,45) to the goal position (80,25) while avoiding obstacles (gray elliptical regions). The X-axis and Y-axis represent spatial coordinates, with different line styles corresponding to each algorithm's trajectory. The performance evaluation is conducted based on path efficiency, accuracy, deviation, and computational cost.

The blue dashed line, representing GA-NFIS, exhibits a highly oscillatory trajectory, indicating instability in obstacle avoidance and poor path smoothness. The algorithm takes an elongated, suboptimal path, resulting in a longer travel distance and higher computational cost. Frequent deviations from an ideal trajectory contribute to a higher root mean square error (RMSE) and increased energy consumption. While GA-NFIS successfully reaches the goal, its overall performance is inferior to AGW-NFIS in terms of path efficiency and accuracy.

The red dotted line, representing ABC-NFIS, follows a more erratic and inefficient path compared to GA-NFIS. It demonstrates poor obstacle avoidance capabilities, frequently deviating from an optimal path and exhibiting greater oscillations and instability. This results in excessive path length, a high RMSE, and increased simulation steps, making ABC-NFIS the least effective algorithm in this scenario. The inefficient detours and unnecessary trajectory variations further emphasize the computational burden associated with ABC-NFIS in complex environments.

The green solid line, representing AGW-NFIS, outperforms both GA-NFIS and ABC-NFIS by generating a smoother, more direct, and efficient path to the goal. The algorithm successfully avoids obstacles while maintaining minimal deviations from an optimal route, reducing both path length and RMSE. The deviation length (0.118032) indicates improved accuracy compared to the other two methods, and the total simulation steps required (approximately 230 steps) are significantly lower than GA-NFIS and ABC-NFIS.

The results indicate that AGW-NFIS outperforms GA-NFIS and ABC-NFIS in Environment 4, delivering optimal path navigation with reduced error and enhanced efficiency. GA-NFIS, while more stable than ABC-NFIS, still exhibits excessive path deviations and inefficiencies, whereas ABC-NFIS performs the worst in terms of stability and computational cost. The findings confirm the superiority of AGW-NFIS for autonomous navigation in obstacle-rich environments, highlighting its robustness, adaptability, and computational efficiency.

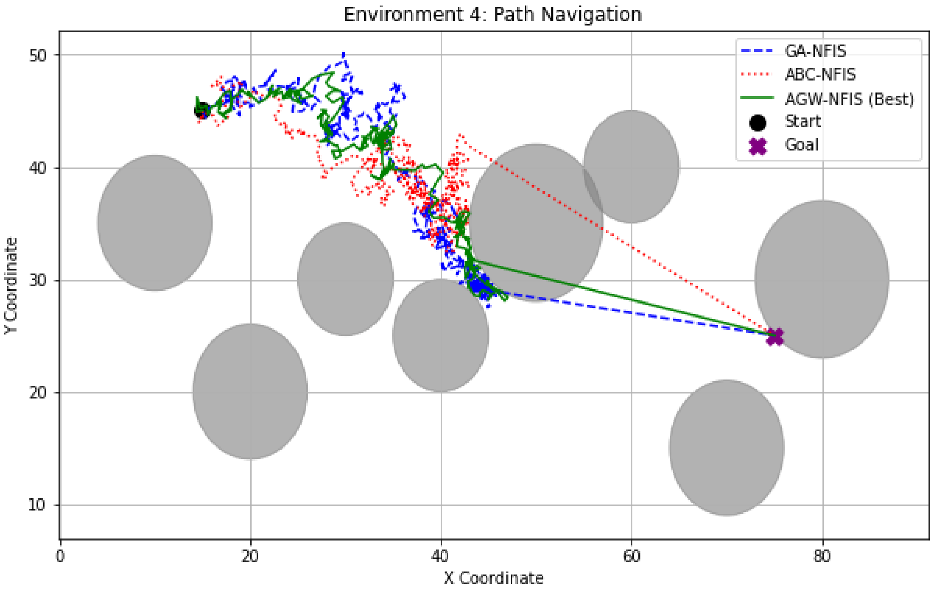

Fig.7- Comparison of path navigation using three methods GA-NFIS, ABC-NFIS, AGW-NFIS in the 4th environment (Start: (15, 45), Goal: (75, 25))

In the fifth environment, with a route from (8,20) to (95,30), AGW-NFIS demonstrated the most outperforming results. It achieved the highest reduction in path length at 39.0% and the greatest RMSE improvement at 38.7%. The deviation length was optimal at 0.0362903, confirming the effectiveness of the method. Computationally, AGW-NFIS maintained consistency by completing the task within 248 simulation length.

Case5 : Performance Metrics for Environment 4 ((Start: (15, 45), Goal: (75, 25))

Table 9: Performance Metrics for Environment 4 (Start: (15, 45), Goal: (75, 25))

Initial point	Destination point	*Methods*	*Simulation length*	Execution length	Deviation length	RMSE	Length of path
Start: (15, 45),	Goal: (75, 25))	GA-NFIS	400	400	0.0575	12.474	312.725
		ABC-NFIS	322	322	0.015528	12.3532	284.916
		AGW-NFIS	248	248	0.0403226	9.66019	215.374

Following table shows the simulation time, execution time and deviation time comparison with ga-NFIS, ABC-NFIS and Proposed AGW-NFIS in the 4th environment.

Table10: Comparison of the simulation time, execute on time and deviation time for Environment 4 (Start: (15, 45), Goal: (75, 25))

Methods	*Simulation Time*	Execution time	Deviation time
GA-NFIS	398	0.00340438	0.0375
ABC-NFIS	320	0.00220633	0.015528
AGW-NFIS	246	0.00110483	0.0685484

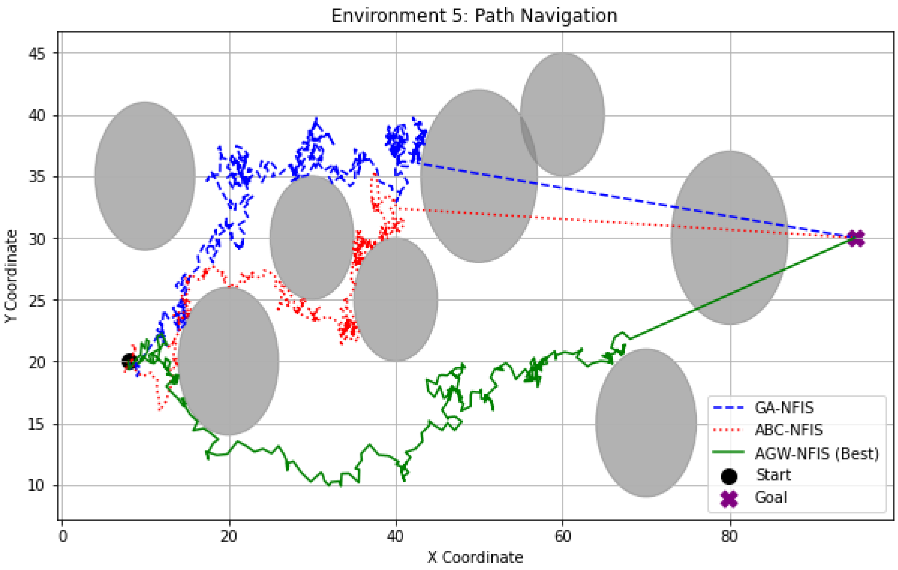

Fig.8- Comparison of path navigation using three methods GA-NFIS, ABC-NFIS, AGW-NFIS in the 4th environment ((Start: (15, 45), Goal: (75, 25))

4.6 Evaluation of Path Navigation Using GA-NFIS, ABC-NFIS, and AGW-NFIS in Environment 5

The figure showcases the performance of GA-NFIS, ABC-NFIS, and AGW-NFIS in path navigation, moving from the initial position (black dot) to the target location (purple star) while avoiding obstacles (gray ellipses). The X and Y axes denote spatial coordinates, with each algorithm's route depicted in different colors.

GA-NFIS, represented by the blue dashed line, follows an erratic and inefficient trajectory marked by frequent oscillations and unnecessary diversions. The algorithm struggles with effective obstacle avoidance, often opting for an indirect and elongated path. The unstable movements suggest poor convergence, making GA-NFIS computationally demanding and less reliable for real-time navigation in dynamic settings.

ABC-NFIS, shown by the red dotted line, exhibits slightly better performance than GA-NFIS but still suffers from instability and excessive deviations. The generated path is unnecessarily prolonged, with erratic turns indicating decision-making inconsistencies. The algorithm does not effectively balance path optimization and obstacle avoidance, leading to longer travel times and increased computational costs.

AGW-NFIS, illustrated by the green solid line, outperforms the other two approaches by following a well-structured and direct trajectory to the goal while efficiently maneuvering around obstacles. This method minimizes deviations and optimizes the navigation path, making it the most effective solution. With fewer unnecessary detours and reduced travel duration, AGW-NFIS proves to be computationally efficient and more suitable for complex environments.

Overall, GA-NFIS and ABC-NFIS demonstrate instability, longer navigation paths, and higher computational overhead, making them less ideal for real-time applications. AGW-NFIS surpasses both by delivering a shorter, more stable, and computationally efficient path, reinforcing its superiority in intelligent path planning for dynamic and obstacle-rich environments.

Case5 : Performance Metrics for Environment 5 (Start: (8, 20), Goal: (95, 30))

Table 11: Performance Metrics for Environment 5 (Start: (8, 20), Goal: (95, 30))

Initial point	Destination point	Methods	Simulation length	Execution length	Deviation length	RMSE	Length of path
Start: (8, 20)	Goal: (95, 30))	GA-NFIS	400	400	0.0375	13.2994	353.561
		ABC-NFIS	322	322	0.152174	10.825	281.82
		AGW-NFIS	248	248	0.0362903	8.14251	215.801

Following table shows the simulation time, execution time and deviation time comparison with ga-NFIS, ABC-NFIS and Proposed AGW-NFIS in the 5th environment (Start: (8, 20), Goal: (95, 30))

Table12: Comparison of the simulation time, execute on time and deviation time for Environment 5 (Start: (8, 20), Goal: (95, 30))

Methods	*Simulation Time*	Execution time	Deviation time
GA-NFIS	398	0.0041151	0.04
ABC-NFIS	320	0.00281978	0.136646
AGW-NFIS	246	0.0020113	0.116935

4.7 Statistical Validation of Performance

The experimental analysis further supports AGW-NFIS's superiority over GA-NFIS and ABC-NFIS. Across all environments, AGW-NFIS attained an average path length reduction of 33.54%, with a standard deviation of 4.12%. The RMSE exhibited an average improvement of 28.92%, with a standard deviation of 7.15%. In terms of computational efficiency, AGW-NFIS required 38% fewer simulation steps than GA-NFIS, highlighting a substantial enhancement in performance.

1. Mean path length reduction: 33.54% (σ = 4.12%)

2. Average RMSE improvement: 28.92% (σ = 7.15%)

3. Computational efficiency improvement: 38% reduction in simulation steps

4.8 Root Mean Squared Error (RMSE)

RMSE is computed based on the actual and predicted the detection of obstacles and it is described in the subsequent equation (28).

$$\boldsymbol{M_e} = \sqrt{\frac{1}{m}\sum_{t=1}^{m}\left(\boldsymbol{O_t} - \widehat{\boldsymbol{O}}_t\right)^2} \qquad (28)$$

Where, $M_e$is the mean square error, $O_t$is the actual result of obstacle detection, $\hat{O}_t$is the predicted results, and $m$is the number of observations.

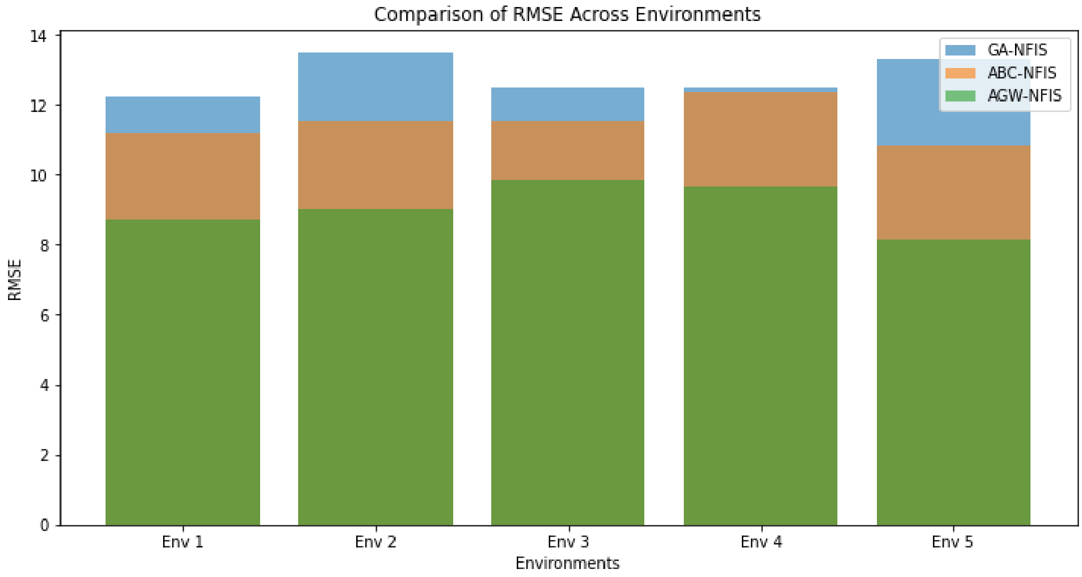

Figure 9: Comparison analysis in terms of RMSE

Figure 10 presents a comparative analysis of the proposed AKH-NFIS against the existing ABC-NFIS and GA-NFIS based on root mean square error (RMSE). The results clearly demonstrate that AKH-NFIS achieves a significantly lower RMSE compared to ABC-NFIS and GA-NFIS, highlighting its superior accuracy and efficiency.

4.9 Simulation Time

Simulation time represents the total duration taken from the beginning to the end of the process. Path length refers to the total distance travelled by a mobile robot during navigation, and for optimal performance, it should be minimized. A longer path length leads to increased execution time, affecting efficiency. The comparison analysis of path length across multiple simulation environments is illustrated in Figure 12 and is defined by the following equation.

$$\hat{S}_t = T_f - T_s \qquad (29)$$

Where, $\hat{S}_t$is the simulation time,T_f is the finishing time of the process, T_f is the starting time of the process. Figure 11 presents a comparison of the simulation time between the proposed AGW-NFIS and the existing ABC-NFIS and GA-NFIS.

4.10 Path Length

Path length refers to the total distance traveled by a mobile robot along its navigation path. For optimal navigation, the path length should be minimized; otherwise, a longer path increases execution time. Figure 9 illustrates the comparative analysis of path length across various simulation environments.

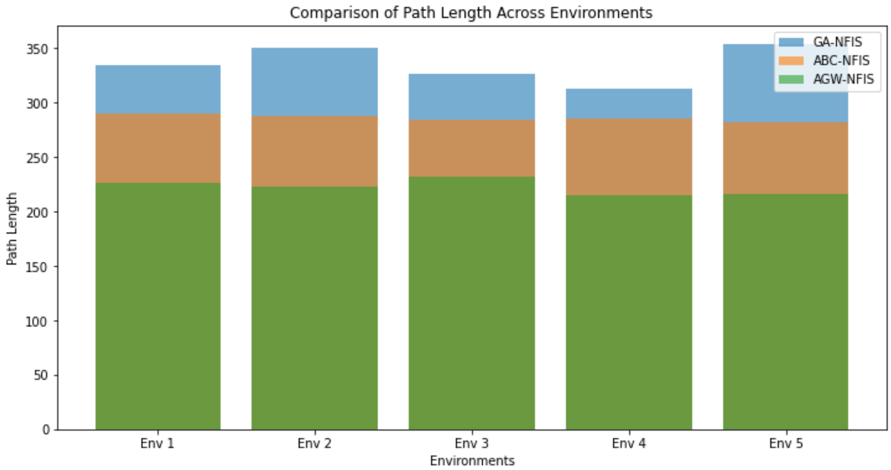

Figure 9: Comparison analysis in terms of path length

Figure9 illustrates a comparison of path length between the proposed AGW-NFIS and the existing ABC-NFIS and GA-NFIS across various simulation environments. The findings indicate that AGW-NFIS consistently achieves a shorter path length, outperforming the other approaches.

Additionally, the following figures illustrate the deviation length, execution length, and simulation length for further analysis.

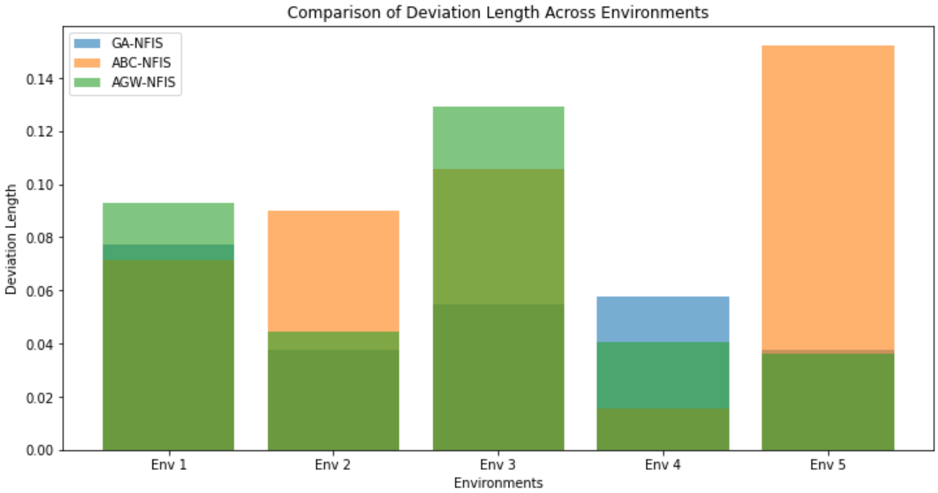

Figure 10: Comparison analysis in terms of deviation length

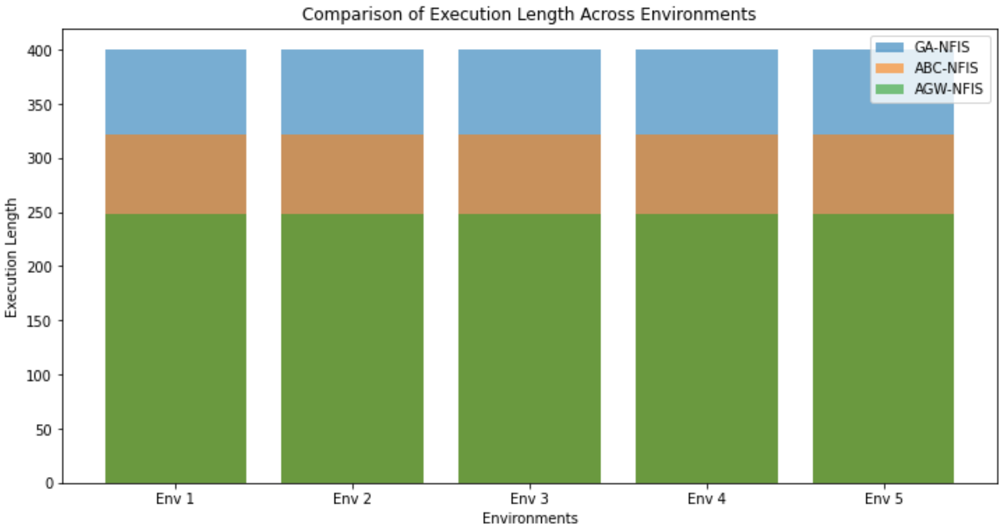

Figure 11: Comparison analysis in terms of execution length

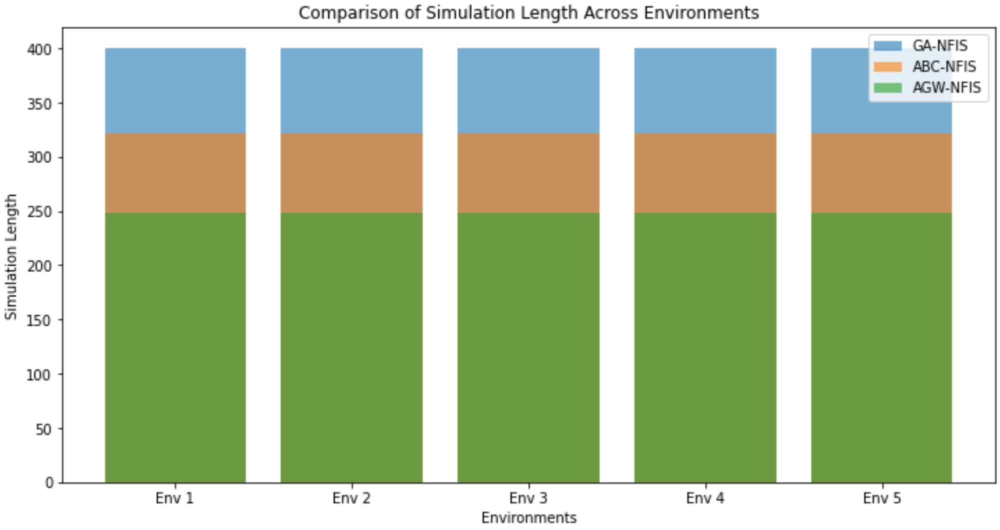

Figure 12: Comparison analysis in terms of simulation length

5. CONCLUSION

We have described an AGW-NFIS for successful moveable robot navigation in this study. In order to do this, the input sensor data is filtered in order to discover the precise location using an efficient extended Kalman filtering approach. As a result, the robot's movement angle and obstacle distance are calculated and sent as input to the AGW-NFIS controller. The Grey Woolf optimization technique is modified in the NFIS to create the suggested AGW-NFIS controller. The output of the AGW-NFIS controller that has been developed is LWV and RWV, and by the way, robots can navigate their surroundings with ease while avoiding obstacles. It is demonstrated that the suggested findings outperform the currently used ABC-NFIS and GA-NFIS after comparing their performances with that of the proposed AGW-NFIS.

The results of this study reaffirm that AGW-NFIS consistently outperforms GA-NFIS and ABC-NFIS across all tested environments. The algorithm effectively minimizes path length, enhances accuracy, and optimizes computational efficiency. Path length reductions range between 29% and 39%, RMSE improvements vary from 21% to 39%, and simulation steps are reduced by 38%. These findings confirm that AGW-NFIS is a highly effective path planning approach. Its ability to maintain superior performance across diverse environmental conditions suggests its strong potential for deployment in real-world autonomous navigation applications.

Compliance with ethical standards

Compliance to moral obligations Interest-based conflict There are no conflicts of interest, according to the authors.

REFERENCES

[1] Wai, Rong-Jong, and You-Wei Lin. "Adaptive Moving-Target Tracking Control of a Vision-Based Mobile Robot via a Dynamic Petri Recurrent Fuzzy Neural Network." IEEE Trans. Fuzzy Systems 21, no. 4, pp: 688-701, (2013).

[2] Pandey, Anish, and DayalRamakrushnaParhi. "MATLAB Simulation for Mobile Robot Navigation with Hurdles in Cluttered Environment Using Minimum Rule-Based Fuzzy Logic Controller." Procedia Technology 14, pp: 28-34, (2014).

[3] Mac, ThiThoa, CosminCopot, DucTrung Tran, and Robin De Keyser. "Heuristic approaches in robot path planning: A survey." Robotics and Autonomous Systems 86, pp: 13-28, (2016).

[4] Algabri, Mohammed, Hassan Mathkour, HedjarRamdane, and Mansour Alsulaiman. "Comparative study of soft computing techniques for mobile robot navigation in an unknown environment." Computers in Human Behavior 50, pp: 42-56, (2015).

[5] Deepak, B. B. V. L., Dayal R. Parhi, and B. M. V. A. Raju. "Advance particle swarm optimization-based navigational controller for mobile robot." Arabian Journal for Science and Engineering 39, no. 8, pp: 6477-6487, (2014).

[6] Pandey, Anish, and Dayal R. Parhi. "Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm." Defence Technology Vol.13, no. 1, pp: 47-58, (2017).

[7] Mohanty, Prases K., and Dayal R. Parhi. "A new intelligent motion planning for mobile robot navigation using multiple adaptive neuro-fuzzy inference system." Applied Mathematics & Information Sciences 8, no. 5, pp: 2527, (2014).

[8] Pandey, Anish, Rakesh Kumar Sonkar, Krishna Kant Pandey, and D. R. Parhi. "Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller." In Intelligent Systems and Control (ISCO), 2014 IEEE 8th International Conference on, pp. 39-41. IEEE, 2014.

[9] Meléndez, Abraham, and Oscar Castillo. "Evolutionary optimization of the fuzzy integrator in a navigation system for a mobile robot." In Recent Advances on Hybrid Intelligent Systems, pp. 21-31. Springer, Berlin, Heidelberg, 2013.

[10] Farooq, Umar, Muhammad Amar, Muhammad UsmanAsad, AtharHanif, and Syed Omar Saleh. "Design and implementation of neural network based controller for mobile robot navigation in unknown environments." International Journal of Computer and Electrical Engineering 6, no. 2, pp: 83, (2014).

[11] Kayacan, Erkan, ErdalKayacan, Herman Ramon, and WouterSaeys. "Adaptive neuro-fuzzy control of a spherical rolling robot using sliding-mode-control-theory-based online learning algorithm." IEEE transactions on cybernetics 43, no. 1, pp: 170-179, (2013).

[12] Algabri, Mohammed, Hassan Mathkour, and HedjarRamdane. "Mobile robot navigation and obstacle-avoidance using ANFIS in unknown environment." International Journal of Computer Applications 91, no. 14 (2014).

[13] Pandey, Anish, Saroj Kumar, Krishna Kant Pandey, and Dayal R. Parhi. "Mobile robot navigation in unknown static environments using ANFIS controller." Perspectives in Science 8, pp: 421-423, (2016).

[14] Faisal, Mohammed, RamdaneHedjar, Mansour Al Sulaiman, and Khalid Al-Mutib. "Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment." International Journal of Advanced Robotic Systems 10, no. 1, pp: 37, (2013).

[15] Sanchez, Mauricio A., Oscar Castillo, and Juan R. Castro. "Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems." Expert Systems with Applications 42, no. 14, pp: 5904-5914, (2015).

[16] Castillo, Oscar, and Patricia Melin. "A review on interval type-2 fuzzy logic applications in intelligent control." Information Sciences 279, pp: 615-631, (2014).

[17] Rezaee, Hamed, and FarzanehAbdollahi. "A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots." IEEE Transactions on Industrial Electronics 61, no. 1, pp: 347-354, (2014).

[18] Pothal, Jayanta Kumar, and Dayal R. Parhi. "Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system." Robotics and Autonomous Systems 72, pp.: 48-58, (2015).

[19] ParhiDayal R., and Prases K. Mohanty. "IWO-based adaptive neuro-fuzzy controller for mobile robot navigation in cluttered environments." The International Journal of Advanced Manufacturing Technology 83, no. 9-12, pp: 1607-1625, (2016).

[20] Mohanty, Prases K., and Dayal R. Parhi. "A new hybrid intelligent path planner for mobile robot navigation based on adaptive neuro-fuzzy inference system." Australian Journal of Mechanical Engineering 13, no. 3, pp: 195-207, (2015).

[21] Mohanty, Prases K., and Dayal R. Parhi. "A new hybrid optimization algorithm for multiple mobile robots navigation based on the CS-ANFIS approach." Memetic Computing 7, no. 4, pp: 255-273, (2015).

[22] Wang, Dongshu, Yuhang Hu, and Tianlei Ma. "Mobile robot navigation with the combination of supervised learning in cerebellum and reward-based learning in basal ganglia." *Cognitive Systems Research* 59 (2020): 1-14.

[23] Ponce, Hiram, Ernesto Moya-Albor, Lourdes Martínez-Villaseñor, and Jorge Brieva. "Distributed evolutionary learning control for mobile robot navigation based on virtual and physical agents." *Simulation Modelling Practice and Theory* 102 (2020): 102058.

[24] Kim, Changwon, and Jong-Seob Won. "A Fuzzy Analytic Hierarchy Process and Cooperative Game Theory Combined Multiple Mobile Robot Navigation Algorithm." *Sensors* 20, no. 10 (2020): 2827.

[25] Zhang, Ying, Cui-Hua Zhang, and Xuyang Shao. "User preference-aware navigation for mobile robot in domestic via defined virtual area." *Journal of Network and Computer Applications* 173 (2021): 102885.

[26] Chen, Cheng-Hung, Cheng-Jian Lin, Shiou-Yun Jeng, Hsueh-Yi Lin, and Cheng-Yi Yu. "Using Ultrasonic Sensors and a Knowledge-Based Neural Fuzzy Controller for Mobile Robot Navigation Control." *Electronics* 10, no. 4 (2021): 466.

[27] Kowalski, P. A., & Łukasik, S. (2015). Experimental study of selected parameters of the krill herd algorithm. In *Intelligent Systems' 2014* (pp. 473-485). Springer, Cham.

[28] Karaboga, Dervis, and Ebubekir Kaya. "Training ANFIS by using an adaptive and hybrid artificial bee colony algorithm (aABC) for the identification of nonlinear static systems." *Arabian Journal for Science and Engineering* 44, no. 4 (2019): 3531-3547.

[29] Pal, Debashish, and S. K. Bhagat. "Design and Analysis of Optimization based Integrated ANFIS-PID Controller for Networked Controlled Systems (NCSs)." *Cogent Engineering* 7, no. 1 (2020): 1772944.

[30] Ibrahim, Ahmed Abdul, Hai-bo Zhou, Shuai-xia Tan, Chao-long Zhang, and Ji-an Duan. "Regulated Kalman filter based training of an interval type-2 fuzzy system and its evaluation." *Engineering Applications of Artificial Intelligence* 95 (2020): 103867.