Intelligent Method for Avoiding Obstacles in Unknown Environment for Swarm Robots Using ANFIS and GWO Technique

Abhisek Das[1*], Sourav Parida[1], Rati Ranjan Dash[2]
[1]Reseach Scholar, [2]Professor
Department of Mechanical Engineering, OUTR, Bhubaneswar, Odisha, India

***Corresponding Author**
E-Mail Id: *abhisek.das.96@gmail.com*

ABSTRACT
In this paper, An Intelligent method using ANFIS designed in MATLAB® for path planning of a mobile robot in an unknown environment avoiding static obstacles is proposed. The robots are equipped with a set of three IR sensor to detect obstacles in detection range i.e. left, front and right respectively. Their membership functions are generated depending upon the distance of detection. The effectiveness of the controller is verified in multiple environments having random static obstacles generated in Coppeliasim® Robotics Simulator and this work is extended for swarm robots using GWO algorithm.

Keywords: *ANFIS, Path planning, Swarm Robots, GWO Algorithm, Coppeliasim®*

INTRODUCTION

These days, mobile robotics mechanism is one of the quickly expanding fields of technological exploration. Because of their capacities, mobile robots can substitution people in many fields. Applications incorporate checking, planetary investigation, watching, crisis salvage activities, observation, petrochemical applications, modern computerization, development, diversion, gallery guides, individual administrations, intercession in outrageous conditions, transportation, clinical consideration just as numerous other modern and nonindustrial applications. The vast majority of these are as of now accessible available market.

Mobile robots can travel independently (in an industrial plant, research facility, or on the surface of a planet) without the assistance of external human administrators. When a robot is self-sufficient, it can determine the steps to be performed to complete a task with the assistance of a perceptual system. It also requires a cognition unit or a control framework to organise all of the robot's subsystems.

Cognition is responsible of examining sensor input information and taking appropriate actions to achieve the mobile robot's goals. It is in charge of the control system scheme. Path planning algorithms, information theory, and artificial intelligence are all needed for navigation. This article deals with mobile robots and how a mobile robot can move in the real world to reach its targets without the assistance of humans. To comprehend the foundation, it should be noted that in a mobile robot, a few mechanical regions and fields must be noticed and incorporated for the robot's proper activity, including the locomotion system and kinematics, perception system (sensors), localization system, swarm optimization, and navigation system.

According to logic interaction and technical rules, the design of swarm robot is elastic, extensible and forming robust cooperative behaviors for the coordination of large numbers of robots. The loss of individuals are the ability to cope with strength .When the leader is absent then the strength of animals are low.

In general animals are extensible by local sensing and communication. Flexibility is the capacity to adapt the cope with an expansive range of various conditions and assignments. Control of multiple-specialist frameworks, with an emphasis on advanced swarm mechanics, has been under increasing degrees of investigation for a variety of applications. Autonomous multi-robot frameworks have numerous applications, including map building of mysterious places, space mining, revelation and exploration of specific zones of the planet. Other feasible jobs for multirobot systems include reconnaissance and rescue using swarms of autonomous mobile robots. Mobile robot swarms outfitted with sensors not just permit a lot of information to be gathered in known and obscure conditions yet additionally wipe out dangers to people in risky activity regions. The exploration field of swarm robotics often produces inspiration from flocking behaviour in nature.

For common velocity, we are generally use Flocking behaviour of the motion of a population. We studied swarming agent control by taking advantage of flocking behaviors robustness. To characterize swarm behavior in a multi-agent system, three simple principles were used: cohesion, separation, and alignment. Different types of navigation strategies are used by autonomous robots.

We can separated motion planning approaches for autonomous robots into two types. The global navigation strategy, which presupposes that the robot obtains an environment map before calculating its path. Local navigation, also known as On-line mode for path planning, is a mode in which the robot determines its position and orientation and controls its mobility using externally attached sensors. The robot can move autonomously utilising sensor data and react to unknown environmental conditions.

In this outline, we can't name every one of them, so we apologize for this deficiency. Nonetheless, we might want to bring up the work did in before contemplates on the grounds that they offer an excursion through the universe of mobile robots (underwater robots, flying octopus, earth exploration robots, etc.), with technical design to mechanical design .

LITERATURE SURVEY

The principles of mobile robotics include mobility, perception, cognition, and navigation. Understanding the mechanism and kinematics, dynamics, and control theory help to address movement problems. The main area of robot signal analysis and specialised disciplines is the computer version and sensor technology.

Swarm intelligence and swarm robotics are approaches based on the mechanisms of natural swarms' animals such as wolves, birds, bees, ants, fish, and humans. One of the most well-known examples of biological self-organized behaviour is that of social insects. According to the character and bounded link between the swarm animals which are able to participate for effective developmental feats like caring for each other's for young also individual, maintain leadership quality for healthy group. The swarm robots are able to carry out their target in a group, so they can never play their individual role, e.g., wolf together for food, but leader of the team guide the path for food and also good security against predators through the so-called "safety through numbers" principle [1-4].

This paper is making plans for collision-unfastened navigation of self-sustaining cell robots to a hybrid GPS-ANFIS primarily based totally technique. The navigation route of the robotic is primarily based totally at the GPS- controller closer to the static goal. From the two GPS modules ,it makes use of the coordinates obtained through rims of the longitudinal axis of the robotic the collectively with the coordinates of the goal to divert it from the present day direction creating a positive attitude closer to the goal .Dynamic navigation is demanding for mobile agents running in an unstructured and crowded environments. Both easy trajectory making plans and most useful direction making plans are primarily based totally on green navigation. For direction making plans is following to the technique of offering direction for the robotic (via intermediate points) to transport from the initial region to the goal region, whilst optimizing over designed heuristics and warding off obstacles [5-9].

Various algorithms has been delivered to address the troubles in direction-making plans for instance ANFIS-logics particle-swarm optimization (PSO) , ant-colony optimization (ACO), Grey wolf optimization(GWO),Whale optimization algorithm(WOA) etc [10-11]. In this paper, for navigation of swarms robots is designed by a hybrid technique using bio insect algorithm. Many researcher have presented the ANFIS logic for control the robot and good path planning for different scenarios of obstacles .So we have discussed the WOA technique (Whale optimization algorithm) to optimization mathematical formula to best path planning and design of structure [12-18].

This paper is organized as follows: in next section Obstacles avoidance of swarms robot is proposed. Then system workflow and pseudo code of system work flow is presented. In the next chapter the steering angle calculation using ANFIS proposed. Then in new section GWO technique is presented with ANFIS (ANFIS-GWO). Then the next section explained the result and simulation of this work[19-20]. The last section is presented the conclusion of the paper.

OBSTACLES AVODANCE

The robotics is designed according to the obstacles avoidance behavior with the orientation between the robotic, target position and obstacles. The robot acts by using steering angle in an unknown environment according to its situation. The static obstacles can found by the help of front, left and right of the robot and the goal position. We are determining the steering angle between the robot and the obstacles and the distance to the target position by using a steering function in the ANFIS controller as shown in Table 1.

SYSTEM WORK FLOW

Here the system workflow is designed in Algorithm 1 [Figure 1]. Three IR sensors are calculated the left, front and right distance between the robot and obstacles. The steering angle is calculated by Steering function through ANFIS controller by using input data from sensor and GPS module(i.e. front GPS module and rear GPS module).The coordinates of two GPS module and goal point are define as front GPS(Fx, Fy), Rear GPS(Rx, Ry), and Goal point(Gx, Gy). The steering angle γ is calculated between η and ψ (Here both ψ and η are the scalar product.), where η=(Rx-Gx, Ry-Gy) and ψ=(Rx-Fx, Ry-Fy) ;

$$\gamma = \cos^{-1}\left(\frac{\eta.\psi}{|\eta||\psi|}\right) \quad (1)$$

Table1: Obstacles Behavior of the robot

Sl. No.	Obstacle Position	Target Position	Steering Angle	Steering Direction
1	Left	Near	<45°	Clockwise
2	Left	Medium	=45°	Clockwise
3	Left	Far	>45°	Clockwise
4	Right	Near	<45°	Counterclockwise
5	Right	Medium	=45°	Counterclockwise
6	Right	Far	>45°	Counterclockwise
7	Front	Near	<90°	CW and CCW
8	Front	Medium	=90°	CW and CCW
9	Front	Far	>90°	CW and CCW
10	Left and Front	Near	<90°	Clockwise
11	Left and Front	Medium	=90°	Clockwise
12	Left and Front	Far	>90°	Clockwise
13	Right and Front	Near	<90°	Counterclockwise
14	Right and Front	Medium	=90°	Counterclockwise
15	Right and Front	Far	>90°	Counterclockwise
16	Left, Right and Front	Near	<180°	Clockwise
17	Left, Right and Front	Medium	=180°	CW and CCW
18	Left, Right and Front	Far	>180°	Counterclockwise

Algorithm 1 (Pseudocode for obstacles avoidance).

Start
Read GPS data, Target location
Compute Target distance (D)
Assign $D_{max} = D$
Calculate η and ψ
Set the weights η and ψ based on D
Set steering angle to 0°
Set threshold distance = 20
Set iteration = 0
While (the robot is not at the target (i.e., $D \neq 0$) is True)
Read the left, front & right IR sensors
Obtain the minimum value from above array
If ((minimum<= threshold distance i.e., Obstacle is detected) Then
Send Senor data to the ANFIS
Compute ANFIS output g
Calculate mean distance ($M = D_{max}/2$)
Send η, ψ, D, M and g to the Steering Function
Update Steering Angle (θ)
Orient and head towards target

Else, No Obstacle is detected
Read GPS data
Calculate D, η, ψ
Feed to Steering Function
Update Steering Angle (θ)
Orient and head towards target
End if
Update iteration (iteration + 1)
End while, Robot at Target location

STEERING ANGLE CALCULATION

In this paper, when the robot is facing the obstacles then the steering angle of robot is calculated by ANFIS-based controller in Figure.2. In this Figure, the ANFIS controller consists of an ANFIS module and a steering function. There are three inputs, (i.e.) l, f, and r, and single output is defined by g in ANFIS controller using the first-order (TSK) fuzzy logic system. The sensor is calculated the distance between robot and obstacles, so 'Left distance', 'Front distance', and 'Right distance' is denoted by l, f, r and output of the steering angle of the robot is denoted by g. The ANFIS controller is determining the new steering angle of the robot for distance i.e. less than the minimum threshold distance , The Gbell shaped membership function is used to identified the crisp value of linguistic terms .we use the IF-THEN rules in ANFIS controller to calculated the value.

$$s_i = a_i l + b_i f + c_i r + d_i \tag{2}$$

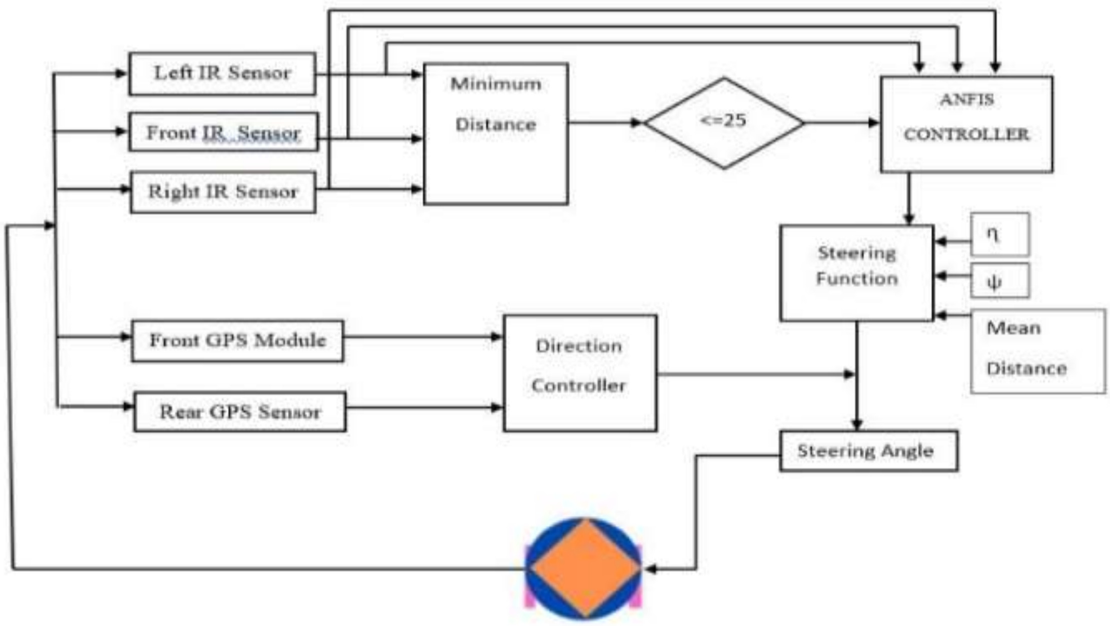

***Fig 1**: Schematic diagram of System Work flow*

ANFIS controller consists five layers. The first layer is defined by input data with bell shaped membership function based on belongingness to fuzzy logic,

$$p_{i,1} = \alpha_{A_i}(l) \quad \text{for } i = 1,2,3 \qquad (3)$$
$$p_{i,1} = \alpha_{B_{i-4}}(f) \quad \text{for } i = 4,5,6 \qquad (4)$$
$$p_{i,1} = \alpha_{C_{i-7}}(r) \quad \text{for } i = 7,8,9 \qquad (5)$$

Where l,f and r are crisp value inputs of node i, , α_{A_i} , $\alpha_{B_{i-4}}$ and $\alpha_{C_{i-7}}$ are bell-shaped membership functions of linguistic parameters .Membership degree of each nodes are calculated by using membership function as shown in Figure 3 below;

$$\alpha_{A_i} = \frac{1}{1+\left|\frac{l-c_i}{a_i}\right|^{2b_i}} \qquad (6)$$
$$\alpha_{B_{i-4}} = \frac{1}{1+\left|\frac{f-c_i}{a_i}\right|^{2b_i}} \qquad (7)$$
$$\alpha_{C_{i-7}} = \frac{1}{1+\left|\frac{r-c_i}{a_i}\right|^{2b_i}} \qquad (8)$$

Where $\{a_i, b_i, c_i\}$ define the parameters of the membership functions in the IF-THEN fuzzy rules.

In the second layer of each rule nodes are applied in the AND operator to calculate the result of the antecedent for each fuzzy rule. Fuzzy rule uses to calculate the firing strength form of output function of each fuzzy rules.

So outputs of second layer is calculated the degrees of products corresponding from the previous layer:

$$p_{2,n} = w_n = \alpha_{A_i}(l) \times \alpha_{B_j}(f) \times \alpha_{C_k}(r) \qquad (9)$$

For n = 1,2,--------,8,9 ; i=j=k = 1,2,3;
The third layer of average nodes are specified to the ratio of each ith rule's firing strength to the sum of all rules' firing strength. Hence, the nodes of this layer are determined by the normalized firing strength:

$$p_{3,n} = \overline{w}_i = \frac{w_i}{\sum_{n=1}^{9} w_n} \qquad (10)$$

For n = 1,2, - - - ,8,9;.
In the fourth layer is defined by result of each nodes, the function of each node is calculated the contribution of each i^{th} rule's to the total output:

$$p_{4,i} = \overline{w}_i s_i = \overline{w}_i(a_i l + b_i f + c_i r + d_i) \qquad (11)$$

The fifth layer is also defined of each nodes. The final out put (g) is determined by using results of each nodes with defuzzification process . so

$$p_{5,i} = \sum_{i=1}^{8} \overline{w}_i f_i = \frac{\sum_{n=1}^{8} w_i s_i}{\sum_{n=1}^{8} w_i} \quad (12)$$

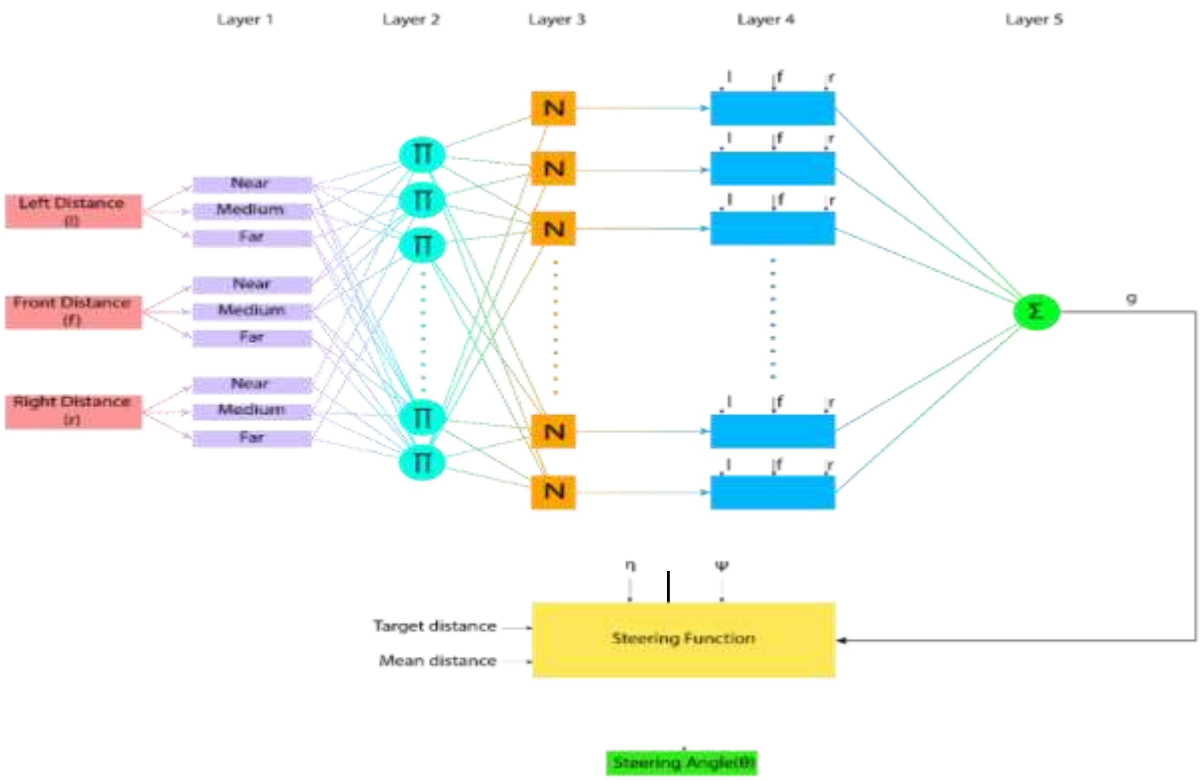

Fig.2: *ANFIS controller with steering function*

The Steering function is used for adjusting the steering angle by using ANFIS controller. The ANFIS output g is then manipulated by a steering function which yields the desired steering angle. The value of Difference is given by the steering function and is calculated using the target distance, the mean distance between the robot and the target, and weighting parameters:

$$Diff = \begin{cases} -\frac{w_n D}{M} \\ 0 \\ \frac{w_f D}{M} \end{cases} \quad (13)$$

Here D and M are the distance and mean distance between the robot and the target, respectively. w_n and w_f are defined as weights for 'near' and 'far' target, respectively. Here, D and M are obtained from:

$$D = \sqrt{(X_T - X_R)^2 + (Y_T - Y_R)^2} \quad (14)$$

$$M = \frac{D_{max}}{2} \quad (15)$$

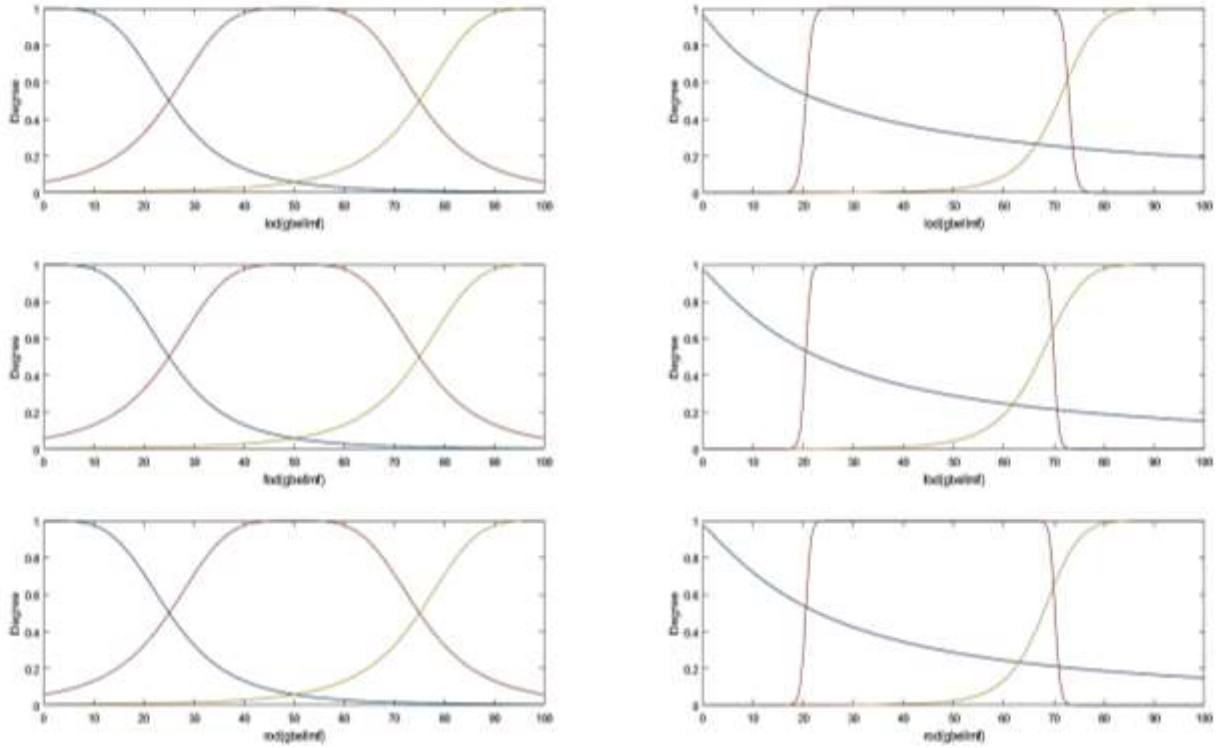

Fig 3: *Membership functions of ANFIS Controller*

Training data of the ANFIS controller

To get the best result of ANFIS controller, the training data is required. The value of linguistic parameters (first layer) and consequent parameters (penultimate layer) are explored during the training. The most favorable value found minimize the error function (LSE). The training data is created in Python3 for obstacle avoidance. The data contained 10,000 records each randomly generated for robot navigation as shown in Figure 4 below. Surface plot of ANFIS proper aspect ratio is shown in Figure 5.

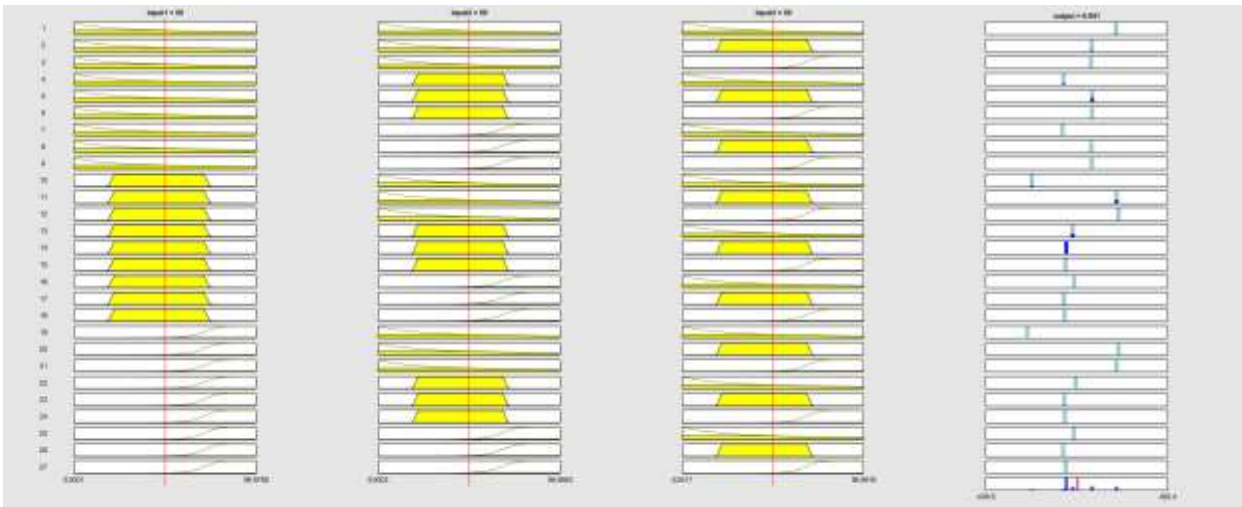

Fig.4: *Rule – View of the AFIS controller*

Algorithm 2 (Pseudocode of generating the training data).

Build a 2D Matrix Training_data of dimension 7500 x 4
Fill Training_data with random values between 0 and 100 (Training_data [i][1]: left obstacle distance, Training_data [i][2]: front obstacle distance, Training_data [i][3] right obstacle distance, Training_data [i][4]: steering angle) except the last column which is set to 0.
Set Threshold distance = 20

While (i <= 7500 is True)
if ((Training_data [i][1] <= Threshold distance) and (Training_data [i][2] > Threshold distance) and (Training_data [i][3] > Threshold distance)) then Training_data [i][4] = 45
elif ((Training_data [i][1] > Threshold distance) and (Training_data [i][2] <= Threshold distance) and (Training_data [i][3]> Threshold distance)) then Training_data [i][4] = 90
elif ((Training_data [i][1] > Threshold distance) and (Training_data [i][2]> Threshold distance) and (Training_data [i][3] <= Threshold distance)) then Training_data [i][4] = -45
elif ((Training_data [i][1] <= Threshold distance) and (Training_data [i][2] <= Threshold distance) and (Training_data [i][3]> Threshold distance)) then Training_data [i][4] = 90
else if ((Training_data [i][1] > Threshold distance) and (Training_data [i][2] <= Threshold distance) and (Training_data [i][3] <= Threshold distance)) then Training_data [i][4] = -90
elif ((Training_data [i][1] <= Threshold distance) and (Training_data [i][2] <= Threshold distance) and (Training_data [i][3] <= Threshold distance)) then Training_data [i][4] = 180
else Training_data [i][4] = 0;
end if
end while

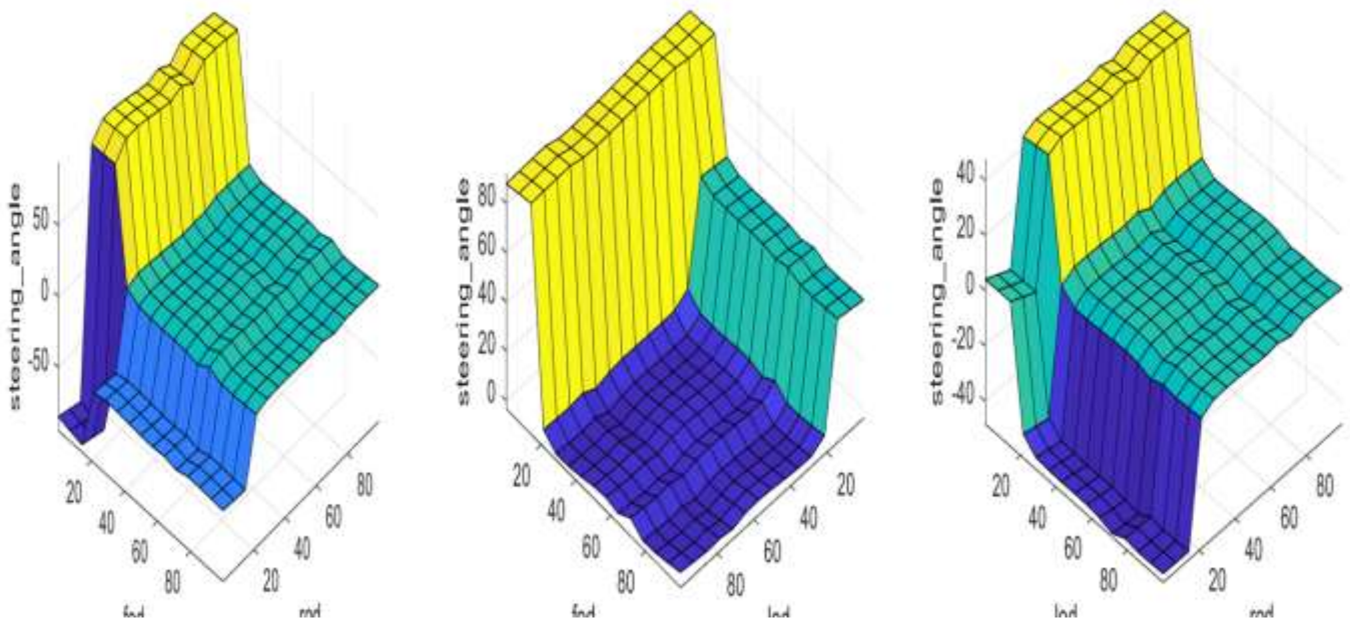

Fig 5: *Surface plot of ANFIS proper aspect ratio*

GREY WOLF OPTIMIZER

The approach of swarm intelligence and swarm robotics are based on the mechanism of the natural swarms animals such as wolfs , birds, bees , ants , fish and also human. Social insects provide one of the best-known examples of biological self-organized behaviour. According to the character and bounded link between the swarm animals ,which are able to participated for effective developmental feats like caring for each other's for young also individual , maintain leadership quality for healthy group.

The swarm robots are able to carry out their target in a group ,so they can never play their individual role, e.g., wolf together for food ,but leader of the team guide the path for food and also good security against predators through the so-called "safety through numbers" principle. Grey wolves have hierarchical social ties since they typically live in packs. In a grey wolf pack, the social rankings go from highest to lowest as follows α, β,δ and ω. The α wolf is the pack leader at the very top of the social structure. The α wolf is mostly in charge of making decisions on predation and other actions.

The α wolf can rule every other wolf except the β wolf at the second level of social order, where it obeys and helps the δ and ω wolfs. The third level of the social structure is occupied by the δ wolf, who is subservient to the wolves and. The dominant wolf has influence over the other wolves. The ω wolves make up the lowest social level and are subservient to other wolves at lower social levels.

Grey wolves surround their victim while hunting. The following is the mathematical representation of these behaviours:

$$\vec{D} = \left|\vec{C}.\vec{X}_{\mathrm{p}}(t) - \vec{X}(t)\right| \tag{16}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{17}$$

$$\vec{X} = 2\vec{a}.\vec{r}_1 - \vec{a} \tag{18}$$

$$\vec{C} = 2\vec{r}_2 \tag{19}$$

Here Eqs. (16) And (17), $\vec{X}(t)$ and Xp are the position vector of the grey wolf and prey. Here t is known as current iteration number. Both A and C are known as coefficient vectors. During the process, the convergence factor is decreasing linearly from 2 to 0. The random vectors (r1 and r2) have a range of [0,1].

$$\left.\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - \vec{A}_1.\left(\vec{D}_\alpha\right) \\ \vec{X}_2 &= \vec{X}_\beta - \vec{A}_2.\left(\vec{D}_\beta\right) \\ \vec{X}_3 &= \vec{X}_\delta - \vec{A}_3.\left(\vec{D}_\delta\right) \end{aligned}\right\} \tag{20}$$

$$\left.\begin{aligned} \vec{D}_\alpha &= \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}\right| \\ \vec{D}_\delta &= \left|\vec{C}_3.\vec{X}_\delta - \vec{X}\right| \\ \vec{D}_\beta &= \left|\vec{C}_2.\vec{X}_\beta - \vec{X}\right| \end{aligned}\right\} \tag{21}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{22}$$

Where $\vec{D}_{\alpha}$, $\vec{D}_{\beta}$and $\vec{D}_{\delta}$ ignify the distances between the current candidate grey wolf and the three wolves(α, β, and δ) respectively. Here $\vec{X}_{\alpha}$, $\vec{X}_{\beta}$ and $\vec{X}_{\delta}$ represent the current position vectors of the three wolves(α, β, and δ), respectively. The grey wolves disperse to different locations to look for prey when |||A |||> 1. The grey wolves all hunt for prey in the same locations when |||A ||| <1. When the end criterion is met, the GWO algorithm is finally stopped. The C vector has random values in the range [0, 2], as can be seen in Eq. (19). In order to stochastically emphasise (||| C |||> 1) or deemphasize (||| C |||< 1) the effect of the prey in defining the distance in Eq., this component gives the prey random weights (16).

Attack prey (exploitation): as already said that after the prey stops moving when the grey wolves end the hunt by attacking the victim. We lower the value of a to mathematically represent the process of approaching the prey. Notably, lowering an also reduces the fluctuation range of$\vec{A}$. The next position of a search agent can be any position between the present position and the position of the prey when the random values of $\vec{A}$ are in the range [-1,1].

Each grey wolf represents a robot. Because a robot's movement is constrained by its maximum speed, it can only move in order to update its position based on its initial position. In contrast to the original GWO, which had to take into account both position updates in the search region and actual robot movements, the robot in this technique needs to advance to the best location or target incrementally as shown in Figure 6(a),(b),(c). The GWO algorithm must be modified to account for the specifics of mobile robot movement in order to be employed in the application of mobile robot target finding as shown in Figure 7.

SIMULATION AND RESULT:-
In this part, obstacle avoidance technique is put to test in a virtual simulation environment. A pioneer 3DX wheel robot is designed in the Coppeliasim® simulator. Here the robot consists two powered wheels (i.e. left and right) and one free wheel (front) and two front and rear mounted GPS modules. It is manipulated in the Coppeliasim® and MATLAB® environment sliding through source code and ANFIS toolbox in MATLAB.
The Navigation of swarm robots is shown in Figure 8.

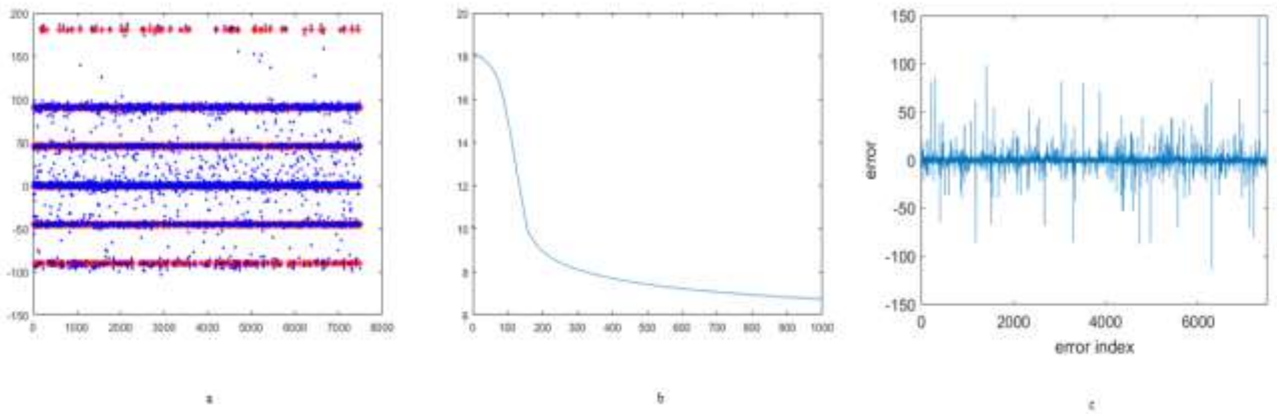

Fig 6: (a) Training target, (b) Epoch vs Training error , (c) Training target vs ANFIS output

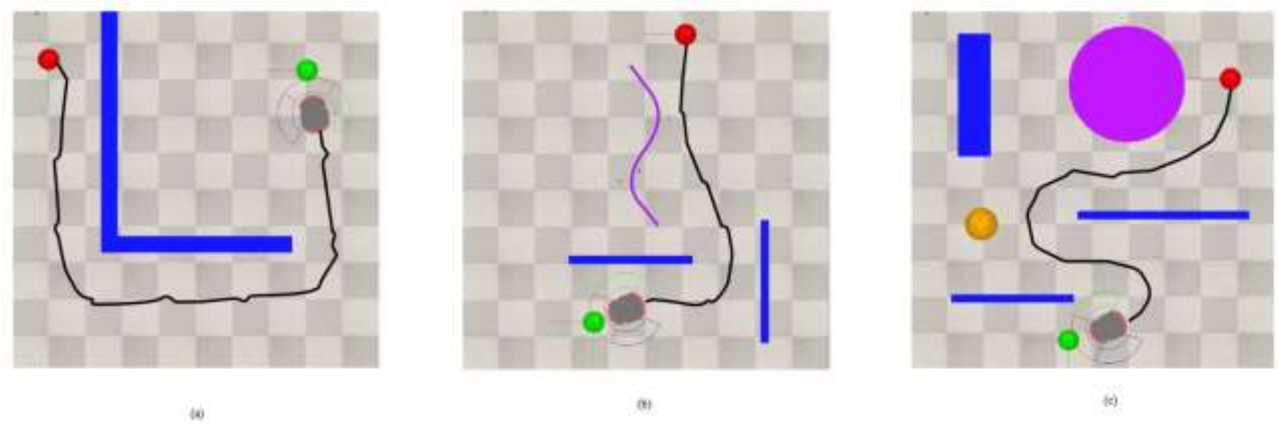

Fig 7: Navigation of Single Robot

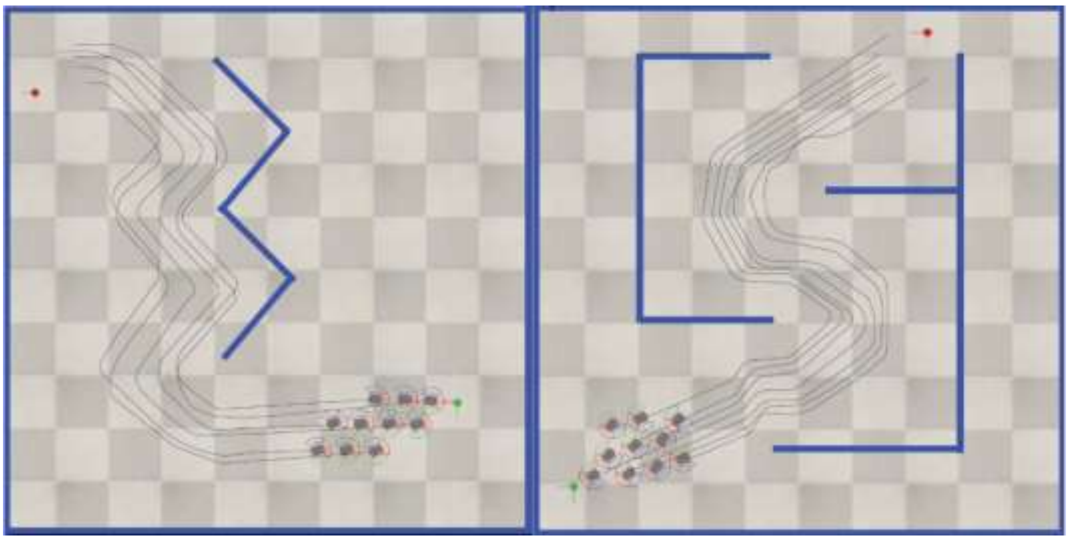

Fig 8: Navigation of swarm robots

CONCLUSION

In this present work, an efficient technique for mobile robot navigation in unknown environment is proposed. The obstacles used in these environments are static in nature, the robot control system uses these environment data to estimate the optimal collision free path

The used technique is based on ANFIS (Adaptive Network based Fuzzy Inference System) which is a kind of Takagi-Sugano fuzzy inference system. The ANFIS controller is paired with a steering function which takes in obstacle distance as input parameters and computes required steering angle needed to avoid collision.

The robot used in simulation is equipped with three IR(Infra-red) sensors oriented at left, right and front direction to detect obstacle positions relative to the robot. In further work we have integrated the prepared ANFIS controller with GWO (Wolf Optimization Algorithm) to

control a swarm of robots in similar environments while achieving desired goal. The validity of the proposed technique is then tested virtually in Coppeliasim® Robotics simulator.
Future work will involve navigation of robots in the presence of long-distance objectives with dynamic obstacles and implementation of more bio inspired algorithm for a robust path planning technique with collision avoidance.

REFERENCES

1. E. Olcay, K. Gabrich, B. Lohmann,” Optimal control of a swarming multiagent system through guidance of a leader-agent”, IFAC-PapersOnLine 52 (20) (2019) 1–6.
2. E. Olcay, B. Lohmann, M.R. Akella, “An information-driven algorithm in flocking systems for an improved obstacle avoidance”, in: 45th Annual Conference of the IEEE Industrial Electronics Society, IECON, Vol. 1, IEEE, 2019, pp. 298–304.
3. Mauro S. Innocente, Paolo Grasso,” Self-organising swarms of firefighting drones: Harnessing the powerof collective intelligence in decentralised multi-robot systemsMauro, Journal of Computational Science 3480–101, (2019).
4. P.K. Das and P.K. Jena, Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators, Applied Soft Computing Journal (2020).
5. S. Mutti , G. Nicola , M. Beschi , N. Pedrocchi , L. Molinari Tosatti,” Towards optimal task positioning in multi-robot cells, using nested meta-heuristic swarm algorithms”,2021.
6. Zhonghua Miao, Jinwei Yu, Jinchen Ji, Jin Zhou,” Multi-objective region reaching control for a swarm of robots, 2019.
7. Zhi Feng, Guoqiang Hu, Yajuan Sun, Jeffrey Soon,”An overview of collaborative robotic manipulation in multi-robot systems”, 2020.
8. E. Olcay, K. Gabrich, B. Lohmann,” Collective navigation of a multi-robot system in an unknown environment.”, 2020.
9. Gabrielle Hedrick, Nicholas Ohi, and Yu Gu,” Terrain-Aware Path Planning and Map Update for Mars Sample Return Mission “, 2020.
10. Andreja Malus, Dominik Kozjek, Rok Vrabi,” Real-time order dispatching for a fleet of autonomous mobile robots using multi-agent reinforcement learning”, 2020.
11. Hamid Taghavifar, Subhash Rakheja, Giulio Reina,” A novel optimal path-planning and following algorithm for wheeled robots on deformable terrains”, 2020.
12. E. Olcay, B. Lohmann, M.R. Akella, “An information-driven algorithm in flocking systems for an improved obstacle avoidance”, in: 45th Annual Conference of the IEEE Industrial Electronics Society, IECON, Vol. 1, IEEE, 2019, pp. 298–304.
13. Anuj Agrawal, Aniket Gupta, Joyraj Bhowmick, Anurag Singh, Raghava Nallanthighal” ,2020
14. Mohammad Samadi Gharajeh , Hossein B. Jond ,” An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system”,2021
15. Hongwei Tang , Wei Sun , Anping Lin , Min Xue , Xing Zhang ,” A GWO-based multi-robot cooperation method for target searching in unknown environments”,j.eswa.2021.115795.
16. Mohammad Samadi Gharajeh , Hossein B. Jond,” Hybrid Global Positioning System-Adaptive Neuro-Fuzzy Inference System based autonomous mobile robot navigation”, December, 2020
17. ZIMING ZHANG, MENG WU, AND PENG CHEN,” Fuzzy Controller for Autonomous Vehicle Based on Rough Sets”,2019 “Imu.”

https://cdnshop.adafruit.com/datasheets/LSM9DS0.pdf., 2019.
18. .“Neato.”http://www.robotpark.com/image/data/PRO/91353/neato-xv-11-datasheet.pdf, 2019.
19. “Vl53lox.”https://www.st.com/resource/en/datasheet/vl53l0x.pdf. 2019.
20. “Poluluencoderkit.”https://www.pololu.com/product/3542/specs. 2019.