ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20230826000899
Original scientific paper

An Adaptive Deep Belief Networkbased Intelligent moving Robot for Navigation Control using Mamdani-Sugeno Fuzzy Inference System

R. SUBHASHINI*, S. GAYATHRI PRIYA, J. RAJALAKSHMI, R. GANDHI RAJ

Abstract: The Intelligent Moving Robots (IMR) are designed to understand instructions and act accordingly in an independent manner. They use sensors that operate with the help of the Internet of Things(IoT) and Deep Learning (DL) for interpreting and navigating the directions following environmental conditions. Recent advancements use the Artificial Neural Network(ANN) and an Adaptive Neuro-Fuzzy Inference System(ANFIS) to model an efficient engineering system. In this work, a hybrid fuzzy inference system, MSFIS(Mamdani-Sugeno Fuzzy Inference System), is proposed along with Adaptive Deep Belief Networks(ADBN) for identifying and tracing the Direction Finding(DF) capability of the IMR. The MSFIS uses parameters in the range 4:4 (4 Input and 4 Output Parameters). The inputs used are Front View (FV), Left View(LV), Right View(RV), and Back View(BV), and the output (4 directions) might depend on the speed of the wheels used in the Robot. Four directions are used at the output for navigation purposes. The results obtained from simulating the experiments confirm that the suggested navigation controller demonstrates superior viability, efficiency, and resilience. Compared with the existing system, the proposed system outperforms well in accuracy and sensitivity, proving it is well efficient in navigating any new environment.

Keywords: adaptive deep belief networks; intelligent moving robots (IMR); internet of things (IoT); mamdani-sugeno fuzzy inference system

1 INTRODUCTION

In recent years, substantial research has been devoted to developing autonomous mobile robots since they can perform dangerous or extremely small tasks [1]. As a result, innovative new technologies were able to be developed. There are numerous industries in which mobile robots have applications, including manufacturing, aerospace engineering, and nuclear engineering, among others. Current robotics research is primarily concerned with developing strategies to ensure that a robot can travel from its point of origin to its point of destination without colliding with either of those locations along the way [2]. Local or global route planning is possible, depending on one's preference. Environmental factors are responsible for this distinction. It is possible for a mobile robot performing local path planning to have only a vague understanding of its surroundings. The mobile Robot can navigate its surroundings with global path planning. While determining its global path, the mobile Robot is keenly aware of its immediate surroundings [3]. Therefore, many sensors collect data from the surrounding environment to aid movement planning. The topic of robot navigation has been the subject of a substantial amount of academic research.

The intelligence present in nature serves as a model for what we refer to as "intelligent systems" (IS). Examples include learning, adapting, and remaining resilient across many problem domains; effectively using resources; converting data into knowledge quickly and efficiently; and reasoning extrapolative [4]. IS provides a dependable method for addressing complex (resource efficiency), scalable problems (workload measurement, decision making). Integration facilitates the development of intelligence in engineering, which results from the accumulation of experience and knowledge. This definition can also be used to refer to a person's intelligence, capacity for learning and memory, and rate of adaptation to new circumstances. Computational intelligence is the capacity of a computer program or algorithm to adapt, learn, remember, reason, and deal effectively with ambiguous or incomplete data. This is an expected and normal part of the process. However, artificial intelligence is indispensable in developing intelligent systems. Most people would agree that there are two distinct types of artificial intelligence. Humanistic artificial intelligence (HAI) seeks to determine how machines can mimic human intelligence [5]. Rationalistic AI (RAI) is the other major subfield that focuses on creating machines that can imitate human intelligence.

HAI's ultimate objective is to create intelligent machines. Computer scientists are interested in discovering methods to instruct computers to perform tasks more suited to humans and currently conducting artificial intelligence research. This is an effort to learn how computers behave so that intelligence can be simulated using that knowledge. Artificial intelligence (AI) researchers attempt to create a machine that can mechanically replicate human intelligence by using computer simulations and other methods (AI). Systems that are more human-like in their approach tend to rely less on AI than rationally intelligent ones. Intelligent systems, a subfield of research that emerged from scientific inquiry, can be defined as Systems based on artificial intelligence. This is best explained by referencing these and other related concepts. Artificial intelligence (AI) is developing a smart system by first comprehending the problem (typically in formal rational terms) and then immediately employing the appropriate reasoning to eliminate the problem [6]. As part of a new initiative to achieve machine intelligence, researchers and developers of intelligent system architectures will conduct their research and development. It is based on the theories of chaos, artificial neural networks, probabilistic reasoning, machine learning, and genetic programming, among other foundational concepts. The nature of this discussion is hypothetical and approximate.

2 LITERATURE REVIEW

Artificial Neural Fuzzy Inference System (ANFIS) is a model that combines conventional neural networks and fuzzy logic. Both of these characteristics are included in the model at issue. Using a technique involving fuzzy-neuro networks and pattern recognition, they created a reactive navigation system for a car-like robot. They proposed employing neuro-fuzzy behavior-based control to guide a robot in the form of a car around stationary obstacles. Using neuro-fuzzy techniques, multiple mobile robots can be controlled. The neural network's findings are then relayed to the fuzzy controller, which enables the mobile Robot to navigate complex environments successfully. The simulation and verification results demonstrate the dependability of the method. A dynamic neural-fuzzy system can be used to navigate mobile robots [7].

A Takagi-Sugeno fuzzy controller processes sensor data to determine the heading of the Robot [8]. Experiments indicate that this technique can assist mobile robots in navigating environments about which they have limited prior knowledge. The neuro-fuzzy guidance systems utilized by mobile robots represent a significant technological advancement. During this experiment, a neural-integrated fuzzy controller was responsible for regulating the mobile Robot's steering angle, as well as its speed and direction. Frequently, mobile robots employ a neuro-fuzzy control system. This article [8] illustrates the construction of a neuro-fuzzy controller by combining a Takagi-Sugeno controller with a radial basis function neural network. This action was taken to achieve the desired outcomes. Calculations based on a neuro-fuzzy model were used to determine the path a mobile robot would take in a dynamic environment in order to avoid obstacles. Neuro-fuzzy methods perform better than GA and Mamdani approaches in terms of parameter tuning, optimizing the available resources etc. This was the conclusion reached after contrasting the two strategies as the beginning of the computational intelligence field (CI). A considerable number of researchers are currently focusing their efforts on CI [9].

Bionics concepts are utilized in computational intelligence, which was initially developed to model intelligent natural phenomena. Bionics is a subfield of engineering science. The ability to simulate intelligence characteristics such as learning and adaptability has led to the birth of a new field of study with the potential to restore ecosystems and redesign technological systems. Nonlinear mapping and optimization are performed with a universal approximator in CI. The foundations of the field can be found in the FL, ANN, and EC regions, as well as a few others. The study of hybrid algorithms has recently increased in the subfield of computational intelligence known as artificial intelligence. In contemporary software, neural networks, evolutionary computation, and fuzzy logic are used less frequently [10]. This is because they are becoming progressively more complicated. The technologies that were mentioned earlier are capable of being combined in a wide variety of different ways. There is no competition between fuzzy logic, neurocomputing, and probabilistic reasoning disciplines. They are a cohesive unit that works very well together. Accumulating evidence supports the claim that combining them yields the best results.

Neuro-fuzzy refers to the widespread use of fuzzy logic and neural network techniques in various consumer products and other types of systems [11]. Combining neural networks, evolutionary computation, and fuzzy logic produces dependable tools that can be created rapidly and efficiently to solve complex problems. Differential equations do not perform well when attempting to model indeterminate or poorly understood systems. In spite of this, fuzzy inference systems can model qualitative aspects of human knowledge and reasoning without the necessity of employing quantitative research methods, use conditional rules that can at best, be described as being hazy in their definition [12]. Because they rely on probabilistic reasoning, fuzzy inference systems can achieve this goal. On the other hand, deep learning methods [13] learn sensitive and valuable representations of features from the original set automatically, in contrast to shallow learning methods, which require the user to select features.

Deep learning methods also learn features from larger data sets. Restricted Boltzmann Machines (RBMs) are the building blocks of the Deep Belief Network (DBN), which is the model of deep learning that has gained the most popularity (RBMs). The Deep Belief Network (DBN) is the deep learning model that has gained the most popularity (RBMs)[14]. DBN can automatically learn information pertinent to the task at hand based on the basic feature set provided. The older method, which involved manually selecting features, has been replaced with this method. Second, because neural networks are so good at nonlinear mapping, this idea ought to be implemented in neural network systems so that it can take advantage of their strengths [15]. It is superior at learning complex relationships in fault diagnosis problems compared to shallow learning models such as ANN and SVM, which have fewer hidden layers. These models are used extensively in machine learning. Even though DBN is beneficial in various settings, the field of diagnosing issues with rotating machinery is just starting to develop at this point. Finding faults in rolling bearings in the time or frequency domain can be difficult due to nonlinear factors such as load, friction, clearance, and stiffness. Because of this, finding the source of the problem can be challenging. This is the case because the vibration signals affected by each nonlinear component are distinct. WPT makes a more in-depth investigation of time-frequency patterns possible [16, 17].

This paper proposes a new intelligent navigational controller for solving navigation problems for mobile robots in a completely or partially unknown environment. A new MANFIS (Fig. 1) (Multiple Adaptive Neuro-Fuzzy Inference System) motion controller has been developed to solve the optimization problem. Finally, Python simulation results are presented to verify the effectiveness of the proposed path planner in various scenarios populated by stationary obstacles.

3 PROPOSED SYSTEM

The proposed system uses a hybrid model called the fuzzy inference system, which uses the Mamdani Fuzzy Inference System (MFIS) and Sugeno Fuzzy Inference System (SFIS). Then the Adaptive Deep Belief Networks (ADBN) [18] are used for identifying and tracing the direction-finding ability of the IMR. Totally four input parameters and four output parameters are used to obtain the correct traceability. The inputs used here are to locate the directions corresponding to Front View (FV), Left

View (LV), Right View (RV), and Back View (BV), and the output might depend on the direction of wheels sensitivity used in the IMR and is used for the navigation-related purpose. Right turn (RT), Left Turn (LT), Moving Forward (MF), and Moving Backward (MB) are the four directions specifically indicating the wheel direction. The central part of the IMR consists of various sensors and actuators in addition to the controller. The controller system is made with two parts, the logic controller and the module controller. Then for processing the information Edge computing is used to get the data from the cloud. The architecture representing the basic IMR is shown in Fig. 1.

The movement control of the IMR is done by performing the kinematic analysis. The various wheel directions will be responsible for the analysis. The analysis is done by considering the wheel position in a two-dimensional plane, and a couple of methods are used for analyzing the wheel's kinematics, which is the position using polar coordinates or cartesian coordinates. We know that mobile robots might not move by slipping on a two-dimensional plane as accurate contact exists between the wheel and the ground, eliminating the lateral slip between the wheel and the two-dimensional plane. Usually, the researchers use cartesian coordinates for modeling.

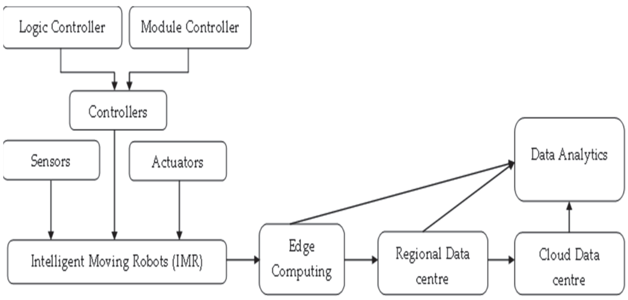

Figure 1 Architecture of the intelligent mobile robot

The basic Robot uses two wheels which are fixed along with one caster wheel, which is driven in a different manner experienced by the steer motion. The driving wheels are driven separately with the help of two various motors for achieving differentiated motion and wheel orientation. The wheels used might have the same diameter as the fixed radius. The distance between the wheels used for driving might get separated by a distance L. The two-dimensional plane indicates the Robot's position considering the coordinates used in cartesian. This is illustrated with the help of the proposed architecture as shown in Fig. 2.

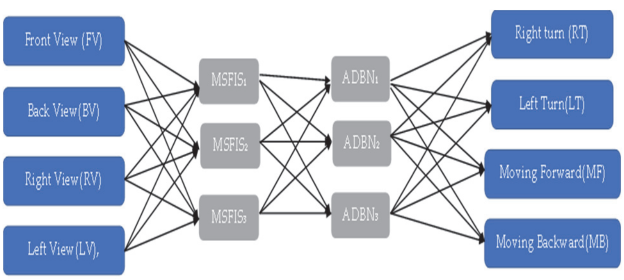

Figure 2 Architecture of the proposed MSFIS-ADBN system

MSFIS (Mamdani-Sugeno Fuzzy Inference System)

The hybrid fuzzy inference system used in this system consists of a neuro fuzzy structure containing Mamdani Fuzzy Inference System (MFIS) [19] and Sugeno Fuzzy Inference System (SFIS) [20]. The two FIS systems are used here in our proposed model for mapping the input and the output data along with the combination of ADBN. This might help compute the membership functions dedicated for FIS. The learning methods used here are ensemble learning methods with backpropagation algorithms, and the various hybrid parameters are represented for indicating the fuzzy membership functions solely. Various learning and Backpropagation methods are used to predict the wheel's velocity in RT, LT, MF, and MB directions for Intelligent mobile robots. The output parameters represent the speed of the wheel. The corresponding parameter representation for input and output is mentioned in Tab. 1.

Table 1 Input and output parameter representation

Sl. No.	Parameters	Assumptions		
	Input	Near	Medium	Far
1	Front View (FV)	$\propto_1$	B_1	Γ_1
2	Left View (LV)	$\propto_2$	B_2	Γ_2
3	Right View (RV)	$\propto_3$	B_3	Γ_3
4	Back View (BV)	$\propto_4$	B_4	Γ_4
	Output			
1	Right turn (RT)	R		
2	Left Turn (LT)	L		
3	Moving Forward (MF)	F		
4	Moving Backward (MB)	B		

Considering the above parameters from Tab. 1, the MSFIS (Mamdani-Sugeno Fuzzy Inference System) uses if-then rules and is defined as follows.

Rule:

IF ($\propto_1$ is FV_i and $\propto_2$ is LV_i and $\propto_3$ is RV_i, and $\propto_4$ is BV_i)

THEN

$$Wheel\ Velocity\ (V) = R\propto_1 + L\propto_2 + F\propto_3 + B\propto_4 \tag{1}$$

Each of the sets R, L, F, and B might be considered as the Fuzzy Membership Sets representing the input variables $\propto_1$, $\propto_2$, $\propto_3$, and $\propto_4$ where i = 1, 2, 3, and 4, whereas R, L, F, and B are the output parameters considered, the linear combination of the input function and ADBN.

The major functions of the MSFIS structure are expressed in the form of mathematical expression as follows.

For Right Turn,

$$\sum_{i=1}^{n} Y_i = \sum_{\alpha=1}^{4} R_j(x) + \sum_{\alpha=1}^{4} \mathrm{P}R_j(x) + \sum_{\alpha=1}^{4} WR_n(x) + \sum_{\alpha=1}^{4} HR_m(x) \tag{2}$$

For Left Turn,

$$\sum_{i=1}^{n} Y_i = \sum_{\alpha=1}^{4} L_j(x) + \sum_{\alpha=1}^{4} \mathrm{P}L_j(x) + \sum_{\alpha=1}^{4} WL_n(x) + \sum_{\alpha=1}^{4} HL_m(x) \tag{3}$$

For Move Forward,

$$\sum_{i=1}^{n} Y_i = \sum_{\alpha=1}^{4} F_j(x) + \sum_{\alpha=1}^{4} \mathrm{P}F_j(x) + \sum_{\alpha=1}^{4} WF_n(x) + \sum_{\alpha=1}^{4} HF_m(x) \tag{4}$$

For Move Backward,

$$\sum_{i=1}^{n} Y_i = \sum_{\alpha=1}^{4} B_j(x) + \sum_{\alpha=1}^{4} \mathrm{P}B_j(x) + \sum_{\alpha=1}^{4} WB_n(x) + \sum_{\alpha=1}^{4} HB_m(x) \tag{5}$$

The various sensors are used to determine the values of all possible directions while specifying the obstacles and the targets. The layer used is the fuzzy layer which includes fuzzification and defuzzification process. The nodes used here in this process might correspond to the calculation scenario of the membership functions typically indicated in fuzzy set. The four inputs produce four corresponding outputs and are represented as there in Eqs. (2), (3), (4) and (5).

For linear consideration of the input and the output, the combination of M-S Fuzzy inference system is implemented, but with certain difficulties in dealing with the multi-parameters. For better evaluation of the input and the output values with certain fuzzy rules commonly the Mamdani model is used as it shows some variation in understandability and legibility. The combination of the sum and the products will be responsible for the following mathematical expression. At least the output value is obtained using the centralized fuzzification and the defuzzification process, which is made equal to the average value of the centroids represented as,

$$\Psi(f_i) = \left(Q(f_i) \times \alpha\right) + \left(Y(f_i) \times \beta\right) \tag{6}$$

Here, the $Q(f_i)$ and $Y(f_i)$ are considered as the weighted parameters by implementing the fuzzy expression $\Psi(f_i)$. Then the corresponding MS of f_1 is expressed as

$$Z_{MS} = \frac{\sum_{x=1}^{n}\left(\mu M(x) + \mu S(x)\right)}{\sum_{x=1}^{n}\left(\mu M(x) + \mu S(x)\right)} + R(x) \tag{7}$$

Adaptive Deep Belief Networks (ADBN)

The adaptive deep belief networks used here are to optimize the IMR flow. Commonly ADBN are used for initiated unsupervised learning. The robotic moves are unpredictable; thus, unsupervised learning is used here. ADBN have the property of adapting the generated changes that occur in the input data in correspondence with the output data. The architecture of the ADBN consists of various layers of neurons, and the corresponding layer contains separate functions indicated within the range. The initial layer is mentioned as the visible layer, and those layers may deal with the generated input data. The rest of the layers are considered hidden layers, and the final layer is called the output layer.

The hidden layers consist of different sub-layers of MSFIS and ADBN. Here the Fuzzification and defuzzification layers are hidden as they can deal with the data hierarchically. The fine-tuning in MSFIS implementation makes the ADBN to understand the identified new features and data patterns for making it more robust and accurate. The restricted Boltzmann machine is a technique used by ADBN for reducing the dimensionality of the data. ADBN could be used for dealing with high-dimensional data. It is considered a generative model that could represent high- and low-dimensional data. Layers of neurons that each carry out a specific function make up ADBNs. The first layer, sometimes referred to as the visible layer, is where the input data are located. Hidden layers, which make up the remaining tiers, hierarchically handle data. The architecture of ADBN is designed to inhibit changes in the data. Pre-training along with fine-tuning are the two phases of ADBN training. Before training, the network's weights are randomly initialized, and limited Boltzmann machines are used to train the network (RBMs). RBMs are generative models that discover a lower-dimensional space in which to represent data. The network can learn crucial data features and patterns during the pre-training phase. The algorithm representing the working of ADBN in the Navigation control of IMR is shown in the following algorithm.

Algorithm 1:

Step 1: Collect the data from sensors on the Robot, such as cameras or LIDARS cans.

Step 2: Pre-process the data to extract Needed features, such as edges in camera images or obstacles in LIDARscans.

Step 3: Split the data into training and Testing sets.

Step 4: Initialize the neural network with a set of input and output layers, and a set of hidden layers with random weights.

Step 5: Train the neural network on the training data using various optimization algorithms.

Step 6: Evaluate the neural network's performance on the validation set.

Step 7: If the validation loss has not improved for a certain number of epochs, increase the complexity of the neural network by adding more hidden layers, increasing the number of neurons in each hidden layer, or using more complex activation functions.

Step 8: Repeat steps 5-7 until the validation loss has converged or the desired level of performance has been achieved.

Step 9: Deploy the trained neural network on the Robot to make navigation decisions based on the sensor data.

This algorithm uses deep neural networks to learn a mapping between sensor data and robot navigation decisions. By adjusting the complexity of the neural network based on the validation loss, the algorithm can adapt the model to the complexity of the task and improve its performance over time. The trained neural network can then be deployed on the Robot to make navigation decisions in real time based on the sensor data it receives.

Back propagation is the method used for training the network at the fine-tuning stage. Normally the backpropagation is a supervised learning technique that might be used to adjust the network weights to minimize the variation between the actual and the predicted result. The stages used for the fine tuning might allow the available network to inhibit the various changes that are needed to process the input data to make it more optimized. This could be mathematically expressed as shown in Eq. (8).

$$\sum_{x=1}^{n} R\mu x_i = \sum_{w=1}^{n}\sum_{y=1}^{n}\left(P\left(w_i x_i\right)+Q\left(w_i x_i\right)\right)+f\left(xi\right) \tag{8}$$

$$\begin{gathered} f_k\left(R_i\right)=\sum_{\alpha=1}^{n}R\left(\alpha\right)+\sum_{\beta=1}^{n}R\left(\beta\right)+\sum_{\gamma=1}^{n}R\left(\gamma\right)f_k\left(L_i\right)= \\ \sum_{\alpha=1}^{n}L\left(\alpha\right)+\sum_{\beta=1}^{n}L\left(\beta\right)+\sum_{\gamma=1}^{n}L\left(\gamma\right)f_k\left(F_i\right)= \\ \sum_{\alpha=1}^{n}F\left(\alpha\right)+\sum_{\beta=1}^{n}F\left(\beta\right)+\sum_{\gamma=1}^{n}F\left(\gamma\right)f_k\left(B_i\right)= \\ \sum_{\alpha=1}^{n}B\left(\alpha\right)+\sum_{\beta=1}^{n}B\left(\beta\right)+\sum_{\gamma=1}^{n}B\left(\gamma\right) \end{gathered} \tag{9}$$

This shows that there is a threshold value available in representing each individual neuron's availability at which the excitation level might depend upon the value, which is fixed as the threshold value. Mathematically it is expressed as,

For the *l*-th neuron,

$$\begin{gathered} \mathrm{T}\left(l\right)=\Sigma\_\left(n=0:l\right)\left(R\left(n\right)\times L\left(n\right)\right)+ \\ +\Sigma\_\left(k=0:l\right)\left(F\left(k\right)+B\left(k\right)\right) \end{gathered} \tag{10}$$

Different activation functions represent the threshold functions in hold up with the nonlinear functions. The network structure distinguishes recursive networks with feedback and unidirectional networks. Whether a network is single-layer or multilayer depends on its architecture. In light of the findings, this technique enables ADBNs to learn to predict output signals based on the succession of input signals and the output signals that correlate to those input signals. To reduce the inaccuracy predicted in the artificial neural networks, the algorithms which are used for learning might choose the weights and the limited threshold values for representing each neuron. The error surface which is found in the neural network might be working in a nonlinear task showing its efficiency in complex design. Modifying the threshold values and the initial weights, which are selected randomly, might be required to select the global minimum.

4 RESULTS AND DISCUSSIONS

The directions and the conditions pointed out by the IMR were simulated on PC using Python version 3.10.5. Generally, the cartesian coordinates are used for simulation purposes. Here the coordinates of the directions and the unified coordinates of the obstacles (both static and dynamic) are controlled by the MSFIS controller. For understanding the position of the coordinates specified to the individual robots, the presently used controller for navigation might be used for calculating the Robot's Robot's distances and focusing angles. But the proposed model uses a navigation controller model that uses two different reaction behaviors. One of the navigations is for reaching the target, and another one is for finding and avoiding obstacles.

Initially, the obstacle is considered too far from the Robot to reach the target device. Then the Robot is placed close to the obstacle, and this will vary the velocity of the moving vehicle to avoid the obstacle, which is available in the directions corresponding to the IMR. This will make the Robot initiate various behaviors corresponding to the reaction and is to activate the IMR between the Robot and the obstacle. The experiments after the simulation will be performed by considering the obstacles at various positions. The proposed navigation controller involves navigation for finding the various reactive behaviors, as shown in Fig. 3 and Fig. 4.

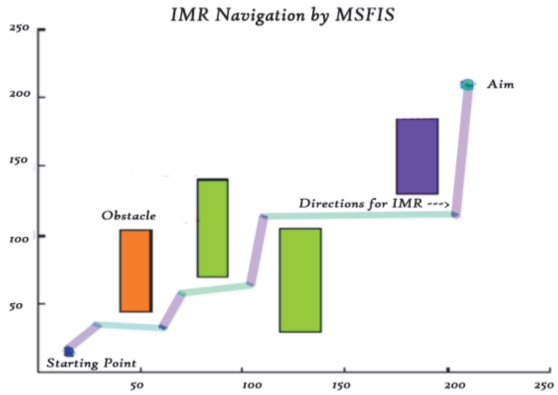

Figure 3 Single robot escaping from the corner end

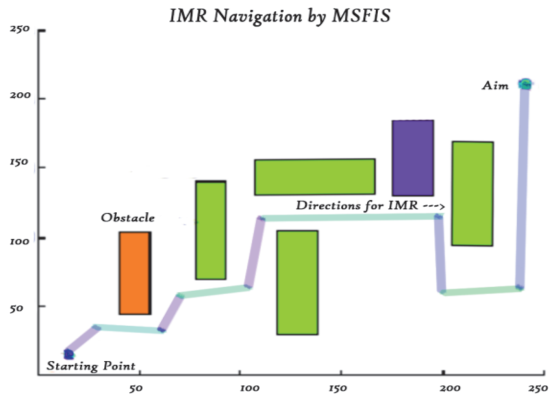

Figure 4 Navigation of single robot using current analysis

The red color in the simulation graph shows the direction of movement of the IMR. This could be verified effectively by setting up the navigation control system using certain authentication techniques. The real-time experiments are conducted using the designed robots as shown in Fig. 5.

The IMR uses 15 infrared sensors and ten ultrasonic sensors placed in the front and back regions of the mobile robot. This could sense the obstacles in the front, back, and side regions. The motion plan of the IMR when using the MSFIS navigation controller will generate very accurate direction findings and obstacle findings. Tab. 2, given below, indicates the direction length used by IMR when searching for targets or obstacles and the error obtained while searching.

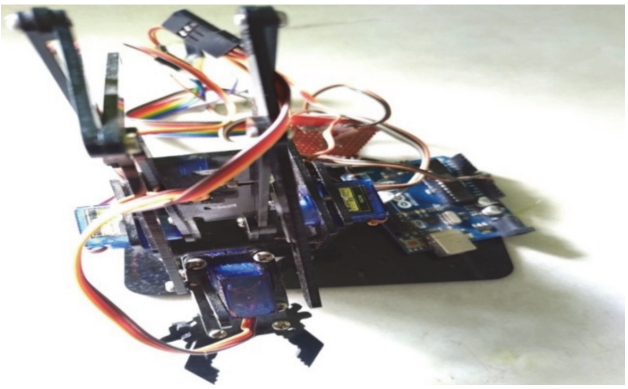

Figure 5 Designed intelligent moving robot

Table 2 IMR path length vs. percentage of error

Sl. No	Direction	Path Length (M)	Percentage of Error
1	Front	1.32	2.12
2	Back	1.12	3.09
3	Left	1.54	2.54
4	Right	1.43	1.96

The variation in the path length is taken for all four directions front, back, left, and right. Graphically the variation identified in the determination of error is given in Fig. 6.

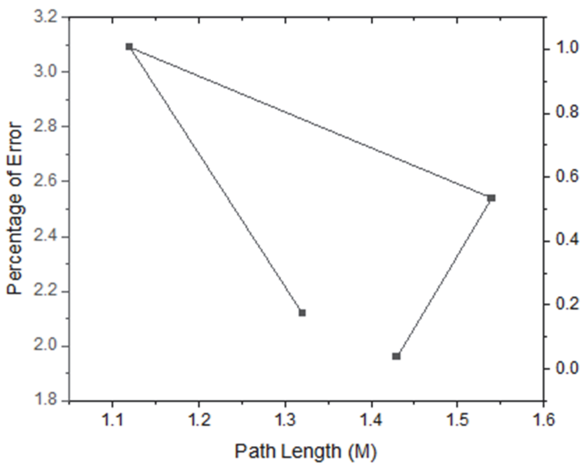

Figure 6 Path length vs. percentage of error

The direction length of the intelligent mobile Robot

Tab. 3, given below, indicates the Wheel Velocity used by IMR when searching for targets or obstacles and the error obtained while moving.

Table 3 IMR wheel velocity vs. percentage of error

Sl. No	Movement	Wheel Velocity / MpH	Percentage of Error
1	Right Move	50	3.33
2	Left Move	46	4.23
3	Forward Move	60	3.36
4	Backward Move	40	1.24

The graphical representation shown in Fig. 7 shows the velocity of the wheel versus the percentage of error obtained in moving the Robot in various directions in accordance with the obstacles.

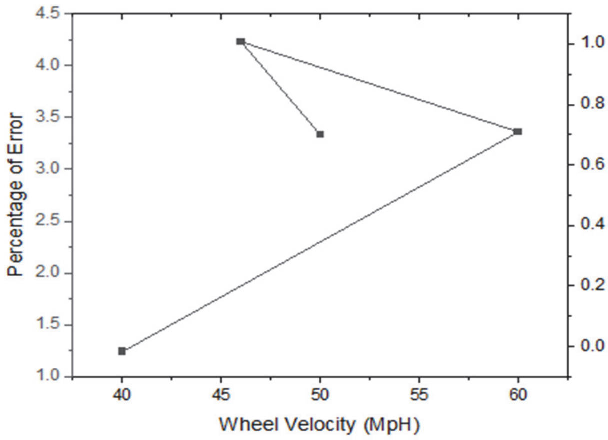

Figure 7 Wheel velocity vs. percentage of error

The MSFIS model is verified by checking the possibility of obstacle prediction and tracking of paths. Primarily the MSFIS is implemented to determine the traffic level of service. Both the training and testing process involved in this proposed model will be extensively used for determining the paths and the level of obstacles present in it.

The implementation of ADBN and MSFIS will resume with specific errors when taking the mean square value. Normally the training process will take about 0.23 seconds and 340 steps for testing as well as the training phase. The training errors for each training step are described in Fig. 8.

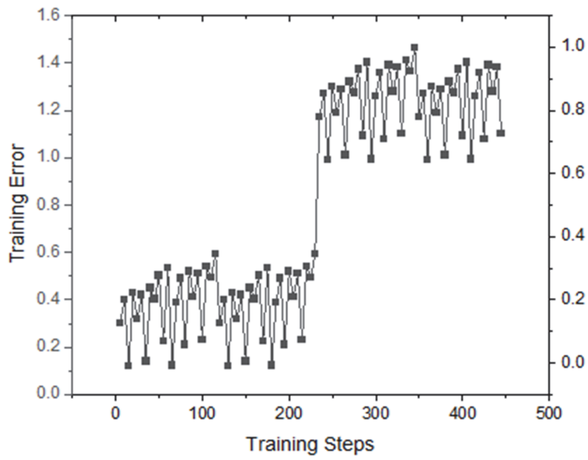

Figure 8 Training steps vs. training errors

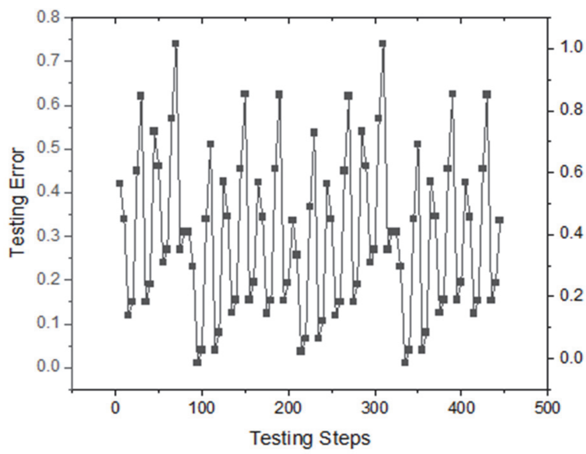

Figure 9 Testing steps vs. testing errors

For testing phase, the modules taken at the real-time rather than simulation is taken into consideration and the errors are more common in the testing phase, as many calibration-related issues exist.

The robots designed (as shown in Fig. 6) will have all testing-related components but induce some error for each operation step as shown in Fig. 9.

Comparison with other Algorithms

The method proposed here in this approach is MSFIS-ADBN and is used to solve the problems related to the local minima and navigation control. Compared with the existing approaches, pathfinding is more accurate and shorter. The planning of directions mentioned by the robots which use the wheel is designed using other artificial neural networks, as shown in Fig. 10.

The intelligent methods implemented for planning and navigation-related issues in IMR using the MSFIS and ADBN will join the repelling effect. This will correspond to the distance of the end target and the obstacles available nearby. The path is obtained to be much shorter than the existing one. This is illustrated in Fig. 11. Considering the existing models for navigation control, the proposed system finds a better way to reach the destination target, and it consumes less time and computational costs with very accurate pointed prediction.

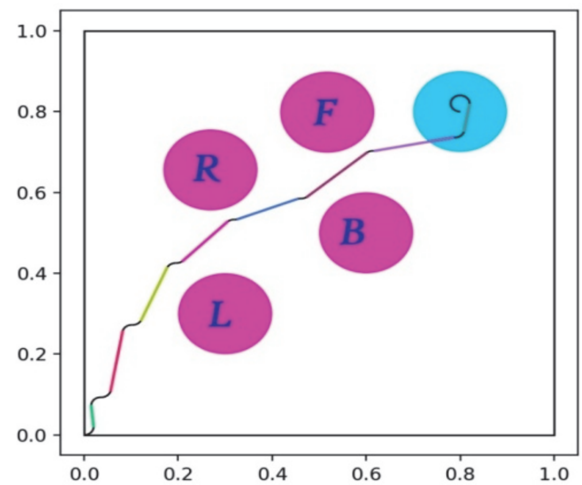

Figure 10 Direction planning of IMR using state of art methods (ANN)

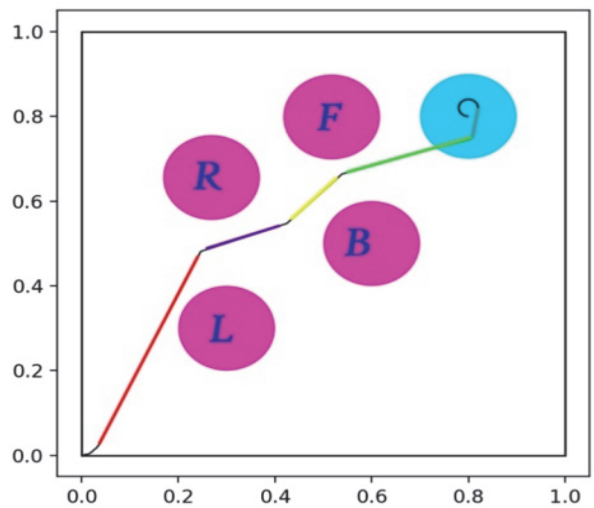

Figure 11 Direction planning and obstacle identification using MSFIS_ADBN

5 CONCLUSION

In this work, an adaptive navigation controller was designed to point in the shortest direction and find obstacles. Here MSFIS (Mamdani-Sugeno Fuzzy Inference System) is proposed along with Adaptive Deep Belief Networks (ADBN) for identifying and tracing the direction finding (DF) capability of the IMR. Four directions are used at the output for navigation purposes. The MSFIS uses parameters in the range 4:4 (4 Input parameters and 4 Output Parameters). The inputs used are Front View (FV), Left View (LV), Right View (RV), and Back View (BV), and the output might depend on the speed of the wheels used in the Robot. The experimentation results made by simulation prove that the proposed navigational controller provides better feasibility, effectiveness, and robustness results. Compared with the existing system, which is designed using other ANN algorithms, the proposed system outperforms well in the form of accuracy and sensitivity, proving it is well efficient in navigating any new environment. In future, advanced algorithms may be employed for the optimization process and for routing the output based on the 4 inputs.

6 REFERENCES

[1] Staczek, P., Jakub, P., Wojciech, D., & Arkadiusz, G. (2021). A digital twin approach for the improvement of an autonomous mobile robots (AMR's) operating environment-A case study. *Sensors*, *21*(23), 7830. https://doi.org/10.3390/s21237830

[2] Mo, Y., Sihan, M., Haoran, G., Zhe, C., Jing, Z., & Dacheng T. (2021). Terra: A smart and sensible digital twin framework for robust robot deployment in challenging environments. *IEEE Internet of Things Journal*, *8*(18), 14039-14050. https://doi.org/10.1109/JIOT.2021.3068736

[3] Sabapathy, S., Surendar, M., Suresh, K., Ananth, T., & Nishanth, R. (2022). Competent and Affordable Rehabilitation Robots for Nervous System Disorders Powered with Dynamic CNN and HMM. *Intelligent Systems for Rehabilitation Engineering*, 57-93. https://doi.org/10.1002/9781119785651.ch3

[4] Lee, A., Andrés, C., & Seng, C. (2023). A Human-centric automated essay scoring and feedback system for the development of ethical reasoning. *Educational Technology & Society*, *26*(1), 147-159. https://doi.org/10.30191/ETS.202301_26(1).0011

[5] Zhang, C., Weidong, Z., Jun, D., Yong ,W., & Xulong, C. (2023). Ethical impact of artificial intelligence in managerial accounting. *International Journal of Accounting Information Systems*, *49*, 100619.

[6] Haider, M., Zhonglai, W., Abdullah, A., Hub, A., Hao, Z., Shaban, U., Rajesh, K., Usman, M., & Pengpeng, Z. (2022). Robust mobile robot navigation in cluttered environments based on hybrid adaptive neuro-fuzzy inference and sensor fusion. *Journal of King Saud University-Computer and Information Sciences*, *34*(10),9060-9070. https://doi.org/10.1016/j.jksuci.2022.08.031

[7] Ayhan, G., Samet, D., & Yasar, B. (2023). Simplified and Smoothed Rapidly-Exploring Random Tree Algorithm for Robot Path Planning. *Technical Gazette*, *30*(3), 891-898. https://doi.org/10.17559/TV-20221015080721

[8] Nizar, H. & Mustafa, W. (2023). An Adaptive Neuro-Fuzzy Based on a Fractionalorder Proportional Integral Derivative Design for a Two-Legged Robot With an Improved Swarm Algorithm. *Engineering Review*, *43*(1), 1-19. https://doi.org/10.30765/er.1916

[9] Petar, P., Denis, K., Alen, S., & Tomislav, P. (2023). Prototyping and Integration of Educational Low-Cost Mobile Robot Platform. *Tehnički Glasnik*, *17*(2), 179-184. https://doi.org/10.31803/tg-0220714131724

[10] Deng, W., Hailong, L., Junjie, X., Huimin, Z., & Yingjie, S. (2020). An improved quantum-inspired differential evolution algorithm for deep belief network. *IEEE Transactions on Instrumentation and Measurement*, *69*(10), 7319-7327.

[11] Joseph, A., Christian, A., & Festus, O. (2022). Design And Simulation of a Mobile Robot Platform for Navigation and Obstacle Detection. *Engineering Review*, *42*(1), 56-65. https://doi.org/10.30765/er.1644

[12] Zhang, P. & Xinyuan, C. (2021). Internal leakage diagnosis of a hydraulic cylinder based on optimization DBN using the CEEMDAN technique. *Shock and Vibration*, 1-10.

[13] Jin, Z., Deqiang, H., & Zexian, W. (2022). Intelligent fault diagnosis of train axle box bearing based on parameter optimization VMD and improved DBN. *Engineering Applications of Artificial Intelligence*, *110*, 104713.

[14] Tao, J., Chengjin, Q., Weixing, L., & Chengliang, L. (2019). Intelligent fault diagnosis of diesel engines via extreme gradient boosting and high-accuracy time-frequency information of vibration signals. *Sensors*, *19*(15), 3280.

[15] Sandi, B., Nikola, A., Zlatan C., & Mario S. (2022). Prediction of Robot Grasp Robustness using Artificial Intelligence Algorithms. *Technical Gazette*, *29*(1), 101-108. https://doi.org/10.17559/TV-20210204092154

[16] Wang, L., Zhiwen, L., Hongrui, C., & Xin, Z. (2022). Subband averaging kurtogram with dual-tree complex wavelet packet transform for rotating machinery fault diagnosis. *Mechanical Systems and Signal Processing*, *142*, 106755.

[17] Norbert, B., Gabor, K., & Aron, B. (2023). Implementation of Trajectory Planning Algorithms for Track Serving Mobile Robot in ROS 2 Ecosystem. *Technical Gazette*, *30*(4), 1020-1028. https://doi.org/10.17559/TV-20220823131848

[18] Kannan, S., Prabakaran. D., Dhenesh Kumar, S., & Sivaram, S. (2023). A Deep Learning-Based Convolution Neural Networks to Forecast Wind Energy. *International Conference on Recent Trends in Electronics and Communication (ICRTEC)*, 1-6. https://doi.org 10.1109/ICRTEC56977.2023.10111917

[19] Jiang, Z., Xu, J., Li, H., & Huang, Q. (2020). Stable Parking Control of a Robot Astronaut in a Space Station Based on Human Dynamics. *IEEE Transactions on Robotics*, *36*(2), 399-413. https://doi.org 10.1109/TRO.2019.2936302

[20] Zhang, B., Okutsu, M., Ochiai, R., Tayama, M., & Lim, H. O. (2023). Research on Design and Motion Control of a Considerate Guide Mobile Robot for Visually Impaired People. *IEEE Access*, *11*, 62820-62828. https://doi.org 10.1109/ACCESS.2023.3288152

Contact information:

Dr. R. SUBHASHINI, Associate Professor
(Corresponding Author)
Department of Information Technology,
Sona College of Technology, Salem
E-mail: subhashini.it@sonatech.ac.in

S. GAYATHRI PRIYA, Assistant Professor
Department of ECE,
R.M.D. Engineering College
E-mail: sgp.ece@rmd.ac.in

Dr. J. RAJALAKSHMI, Associate Professor
Department of Biomedical Engineering,
Velalar College of Engineering & Technology, Erode - 12
E-mail: rajivcet21@yahoo.com

Dr. R. GANDHI RAJ, Assistant Professor
Department of Electrical and Electronics Engineering,
University College of Engineering,
BIT Campus, Anna University, Tiruchirappalli
E-mail: gandhirajeee@gmail.com