International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Design and Implementation ofAdvanced Deep Q-Network for Robot Motion Control

Dr. Mrunali Sonwalkar
Assistant Professor,
College of Engineering, Ambajogai,
Dist. Beed – 431517, Maharashtra, India

***Abstract*—The mobile robots are increasing day by day because of their numerous applications. The motion control of these robots is very important to finish a task perfectly. The motion control of robots is to identify a time series of control inputs that will result in the required output or motion. Controlling inputs are typically motor currents, although they can be converted into torque or velocity for control design. A required trajectory, which consists of x and y coordinates, often describes the desired motion. In this research, for robot motion control, the advanced Deep Q-Network (DQN) controller is designed. The performance of the DQN controller in robot motion control is evaluated with the conventional controllers of Proportional-Integral-Derivative (PID) and Neuro-Fuzzy Controller (NFC). The performance of the controllers is compared using error metrics like Integral Time Absolute Error (ITAE), Integral Squared Error (ISE), and Integral Absolute Error (IAE). The two-wheeled mobile robot is used to implement the controller designs. The experimental outcome of the motion control of the two-wheeled mobile robot shows the excellence of the DQN controller with an IAE error of 8.45 in x-coordinate tracking and 5.33 in y-coordinate tracking.**

Keywords—Robot Motion, Deep Q Network, Error Metrics, Kinematic Equation, Fuzzy Controller

I. INTRODUCTION

In recent years, mobile robots have been everywhere. They carry out various activities, such as rescue, exploration, material transfer, and education [1]. Leg robots are harder to construct and control because they have more degrees of freedom (DOF). Wheel robots use less energy than leg robots, which rely on ground contact and external driving power. They also have simpler mechanical structures and dynamics. Static stability, which is attained by robots with three or more wheels, makes dynamics easier to understand. The four-wheel configuration is also widely used; it is the most common on vehicles. This is because, at high speeds, a larger supporting polygon gives greater stability. However, if the terrain is completely flat, having more than three wheels limits the mechanism and requires stabilization. This article addresses two-wheeled statically unstable robots. Because they contain two coaxial wheels attached to a middle body and a center of mass over the wheel axles, these robots actively steady themselves to avoid falling over. Compared to other types of mobile robots, those with two wheels have various advantages. While leg robots are simple to control, two-wheeled robots remain more difficult than statically stable wheeled robots. Controlling the movement of robots is the main focus of the research. Motion planning is the function of identifying the most practical and optimal route for the robot to take from its starting point to its target state while avoiding obstacles and meeting constraints [2].

Motion control is the process of directing the robot and the end-effector's speed, position, and orientation in response to commands to carry out motion [3]. Based on whether they presume a linear or nonlinear relationship between the system's inputs and outputs, motion control algorithms are categorized as linear or nonlinear. The simplicity of linear algorithms makes them appealing, but they come at the cost of potential stability and performance issues. Although nonlinear methods are more complicated and flexible, they may be more resource-and adjustment-intensive. Some of the current research on robot control systems is discussed below.

Research [4] shows that a mobile robot with a Non-Uniform Rational B-Splines (NURBS) trajectory and a variable parameter PID controller can reach the target time-varying velocity. It all starts with configuring the nonlinear kinematic error model for the robot. Then, taking the required angular velocity into account, we linearize it to provide an equation for the linear error. As the robot's speed changes, a variable-parameter PID controller makes sure it stays close to the NURBS path with little room for error. To eliminate kinematic errors, controller coefficients are determined through testing and simulation. A platform robot is created to demonstrate the proposed controller. The proposed controller's operation is illustrated through simulations and testing outcomes. The mobile robot's control system is defined in research [5], which employs a DC motor and two wheels. This system uses several controllers to regulate the speed of the DC motor. Initially, it was anticipated that Adaptive Neuro-Fuzzy Inference System (ANFIS), Fuzzy, or PID controllers would be best suited to controlling the speed of the DC motor. The study examines several strategies based on the settling time and speed of the DC motor's output signal. By pursuing this path, ANFIS outperforms PID and Fuzzy Controllers in terms of settling time and output signal overshoots.

A study [6] explains the Dynamic Window Approach (DWA) technique for mobile robots, which uses fuzzy logic

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

controllers (FLC) to dynamically avoid obstacles. One well-known navigation method is DWA. Finding the ideal weights for DWA's objective function is a difficulty for ensuring collision-free navigation to the destination. The study's main contribution is an AI system for efficiently optimizing the weights of the dynamic window's objective functions. This employs a fuzzy logic method to make the system more flexible to changing circumstances and proceed more swiftly toward the goal. Regardless of how dynamic the scenario was, the proposed enhanced adaptive controller decreased DWA failure rates. The work [7] suggests a PID with time-varying parameters to accurately follow the path of a Mecanum-Wheeled Robot (MWR). The first stage is to develop and linearize a robot's nonlinear kinematic error model around the work area. The next step is to utilize this linear model as the foundation for creating a PID controller with variables that can change over time. To keep the robot on the planned course with minimal variation, the control parameter coefficients are estimated through trial and error. To demonstrate the proposed controller, a platform MWR was constructed and manufactured. To demonstrate that the proposed controller works and is accurate, simulation and testing results are presented. The paper [8] offers a novel hybrid control system that integrates model-referenced adaptive and neural network (NN)-based kinematic controllers. The NNs are employed to dynamically set controller parameters in real time. The kinematic controller is adaptively calibrated, ensuring rapid convergence to the expected trajectory. The model-referenced adaptive controller maintains its expected tracking performance even when the model or parameters are changed. The Lyapunov stability technique is employed to derive adaptive gains that provide asymptotically stable error dynamics. Using a range of performance studies, the recommended controller's efficiency is compared to that of PID, kinematic, and dynamic controllers. Research on the effects of parameter uncertainties and slip disturbances on controller performance reveals that the proposed controller surpasses the alternatives in terms of convergence speed and tracking accuracy.

The research aims to design an advanced controller for accurate robot motion. The article is organized as follows: Section I gives the application of mobile robots and recent work done on robot motion control. Section II derives the mathematical modelling of the 2-WMR. Section III discusses the operation of PID, NFC, and DQN controllers. Section IV details the outcomes of the three controllers in robot X and Y coordinate motion control. Section V concludes the research with future directions.

II. MATHEMATICAL MODELING OF ROBOT

The research employs a2-WMR for motion control. The robot consists of one free front wheel and two drive wheels on the same axis and it is shown in Figure 1. The front wheel is for balancing purposes; the motion of the robot fully depends on the two drive wheels. Both orientation and translation are performed using two actuators that operate independently on the two driving wheels.

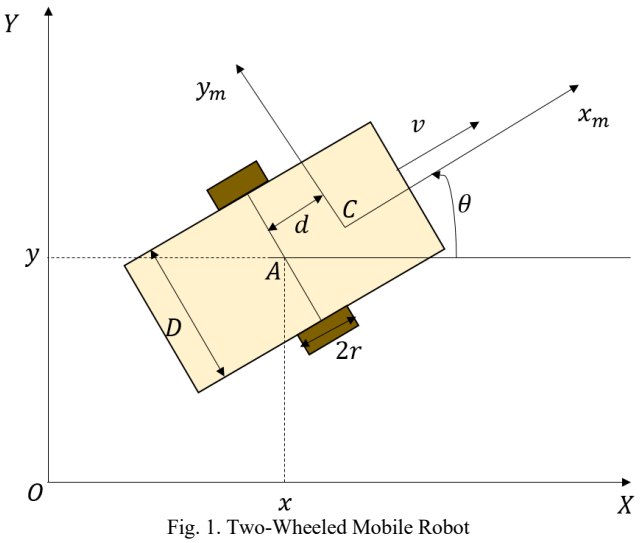

Fig. 1. Two-Wheeled Mobile Robot

In the global frame $\{X, O, Y\}$, the mobile robot's location can be identified by the position of the mass center (C) or the center of the gear (A), as well as the angle between the robot's local frame $\{x_m, C, y_m\}$and the global frame. The following are the kinematic equations for a mobile robot with two wheels:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}cos(\theta) & 0\\ sin(\theta) & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} \quad [1]$$

$$\begin{bmatrix}v\\ \omega\end{bmatrix} = \begin{bmatrix}\frac{r}{D} & \frac{r}{D}\\ -\frac{r}{D} & \end{bmatrix}\begin{bmatrix}v_R\\ v_L\end{bmatrix} \quad [2]$$

In this framework, x and y are the mobile robot's centercoordinates, θ is the vehicle's orientation, v and ωdenotes the linear and angular velocities, v_R and v_L are the right and left wheel's velocity, and D is the mobile robot's base length. By adding the first Equations (1 and 2), we obtain:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}rcos(\theta) & rcos(\theta)\\ rsin(\theta) & rsin(\theta)\\ \frac{r}{D} & -\frac{r}{D}\end{bmatrix}\begin{bmatrix}v_R\\ v_L\end{bmatrix} \quad [3]$$

The velocities of the right and left wheels, v_R and v_L, are the kinematic model inputs for mobile robots. The following constraint prevents the wheels from sliding laterally.

$$\dot{x}sin\theta + \dot{y}cos\theta - d\dot{\theta} = 0 \quad [4]$$

III. CONTROLLER DESIGN

For precise control of the 2-WMR motion in X and Y coordinates, advanced controllers are required. To perform this task, we employ three controllers: PID, NFC, and DQN. The block diagram of robot motion control using the advanced controllers is given in Figure 2.The operation of these three controllers is detailed in this section.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

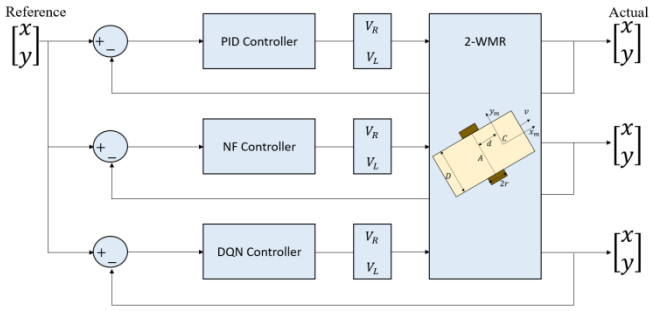

Fig. 2. Block diagram of 2-WMR motion control

A. PID

PID controllers are mostly employed in industrial systems and feature a control loop feedback system [9]. The PID controller continually computes the error score as the difference between the expected and observed values.It corrects errors by adjusting control variables provided to the actuator. $K_p, K_i, and K_d$ correspond to the proportional, integral, and derivative controllers, respectively.

$$C(t) = K_p e(t) + K_I \int_0^\tau e(\tau)d(\tau) + K_D \frac{de(t)}{dt} \quad [5]$$

In Equation (5), $C(t)$ represents the Control signal, $e(t)$ denotes the Error signal, $K_p, K_i,$ and K_d denotes the Proportional, Integral and Derivative gain

B. Neuro-Fuzzy

The design and successful implementation of NFC can be seen in a variety of industrial applications that use Artificial Neural Networks (ANN) to accomplish the functions of an FLC [10]. While the ANN can learn, generalize, and adapt, the FLC can draw inferences based on expert knowledge. By combining FLC and ANN capabilities, the NFC provides a smart control system. NFCs have the characteristics of a nonlinear controller, which can learn, adapt, and make inferences [11]. The NFC controller can be designed without the use of models. Because of these capabilities, the NFC works well with other standard controllers. As a result, non-linear systems with parameter variation and uncertainty typically use NFCs [12]. Consider the following Fuzzy Rules of the Sugeno type:

$$R^j: If X_1, A_1^j \dots X_n, A_n^j then y = f_i = a_0^j + a_1^j X_1 + a_2^j X_2 + a_n^j X_n \quad [6]$$

The input and output variables are X_i and y, respectively. The coefficients of the linear function $f_i = (x_1, x_2, \dots, x_n)$ are indicated as $A_i^j \epsilon R$. The NFC takes two inputs: error (e) and change in error (Δe). For each input, one of five membership functions (MF) was chosen. The second layer is where the MFs are implemented. Each artificial neuron now has an activation function rather than anMF. This layer generates the following output.

$$y_j^2 = exp\left|\frac{(x_i - m_{ij})}{2(\sigma_{ij})^2}\right|^2 \quad [7]$$

Where,σ and $m$indicates the MF'sparameters that need to be changed. Thesecond layer'sith cell receives X_i as input.

The third layer of NFC, similar to FLC, consists of a rule base for establishing Fuzzy Rules.

$$y_k^3 = \prod_j w_{jk}^3 x_j^3 \quad [8]$$

Where,x_j^3 represents the input for the j^{th}cell in the third layer. As shown below, the system's output is specified using Mamdani Fuzzy logic.

$$y_0^4 = \frac{\sum_k w_{k0}^4 y_k^3}{\sum_k y_k^3} \quad [9]$$

The Fuzzy rulesaccuracy is determined in the fourth layer. Layer 5 handles the firing strength of rules. The firing level of normalized rules is multiplied by the linear function in this layer. For the connected test system, this layer produces the required output values.

$$y_0^5 = (p_k x_1^1 + q_k x_2^1 + r_k) \quad [10]$$

The following is the output of layer 6:

$$y_0^6 = V_{nfcs}^* \quad [11]$$

The squared error E, which reduces the tracking error e, is calculated by modifying input and output parameters using the back-propagation approach.

$$E = \frac{1}{2} e^2 \quad [12]$$

The parameters that need adjustment could be updated by the Equation (13):

$$\vartheta(k) = \vartheta(k-1) + \left(-\eta \frac{\partial E(k)}{\partial \vartheta(k)}\right) \quad [13]$$

The parameter ϑ can be adapted, with the learning rate represented by η. The partial derivative is computed using the chain rule. Equation (14) generates the derivative chain that leads to the NFC's output.

$$\delta^1 = \frac{\partial E}{\partial e} \frac{\partial e}{\partial V_{nfcs}^*} \frac{\partial V_{nfcs}^*}{\partial y^6} \quad [14]$$

Where,δ^1 represents the local gradient. Updating the result parameters of the NFC used in the interconnected test system was carried out as follows:

$$\frac{\partial E}{\partial p_k} = \delta^1 \frac{x_j^4}{\sum_j x_j^4} x_1^1, \frac{\partial E}{\partial q_k} = \delta^1 \frac{x_j^4}{\sum_j x_j^4} x_2^1, \frac{\partial E}{\partial r_k} = \delta^1 \frac{x_j^4}{\sum_j x_j^4} \quad [15]$$

The simulation model was employed to train the parameters for the membership layer in the NFC structure. Throughout the simulation studies, the outcome parameters were trained using a back-propagation learning approach.

C. DQN

DQN, a famous model-free, online, off-policy technique, was one of the first RL algorithms studied [13]. To forecast future reward returns, DQN trained a critic for approximating the action value function$Q(s, a|\theta_Q)$. Thus, the approach is based on two spaces, one continuous and one discrete [14]. Consequently, a ϵ-greedy method is utilized to choose the action.

$$a = \begin{cases} \underset{a' \epsilon A}{argmax} Q(s, a'|\theta_Q) if b \le \epsilon \\ random(a' \in A) if b \le \epsilon \end{cases} \quad [16]$$

where A represents the action space, $\epsilon \in [0, 1]$ representsthe exploration rate, and $b \in [0, 1]$representsthe random number. Exploration is often assigned a maximumscoreof 1 at the initial training

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

procedure, and as learning improves, a decay function is used to reduce thescore. This guarantees that the agent initially concentrates on examining the environment before shifting to utilizing the optimal actions.Training runs smoothly due to two factors. This technique stores the history of each step (a, s, s', r, d)as a transition within the memory buffer R. This includes the action, state, new state, reward, and episode-ending flag. The networks are trained by sampling a mini-batch of experiences $M \in R$ from memory. Overfitting may occur in the neural network if it is trained with only the most recent data. To ensure stability, a large replay buffer containing data from a wide range for each variable is required. The amount of memory available limits the number of data points that can be saved. Therefore, new information should only be recorded if it significantly differs from the existing data. It is also recommended to randomly sample batches from the experience buffer for optimal computational performance and to prevent overfitting. DQN learning is referred to as an off-policy scheme because it is not based on previous experiences.Furthermore, to improve learning and optimization stability, the DQN technique employs the concept of the critic's target network. The network, $Q'(s, a|\theta'_Q)$ has asimilar topology and parameterization as $Q(s, a|\theta_Q)$ .Consequently, the target value operation is defined in Equation (17).

$$y = r + \gamma(1-d)\max_{a' \in A} Q'(s', a'|\theta'_Q) \quad [17]$$

The target value function is determined by the same parameters acquired during training, resulting in unstable error reduction. The target network serves this purpose by sharing the same specifications as the critic network but updates more slowly. Therefore, Polyak averaging updates the target network once for every main network:

$$\theta'_Q = \tau\theta_Q + (1-\tau)\theta_{Q'}, \tau \in [0, 1] \quad [18]$$

IV. RESULT AND DISCUSSION

The research focuses on controlling the motion of a mobile robot using advanced controller technologies. For this purpose, a 2-WMR is used in experiments. The controller is deployed to precisely position the robot in exact coordinates using motion control of both wheels. The DQN controller is proposed for controlling the motion of the right and left wheels. To evaluate the performance of the controller, NFC and PID controllers are also employed. All three controllers yield excellent results in the motion control of the robot. For mathematical evaluation, error metrics are chosen and compared. MATLAB software is used to simulate the experiment.

Firstly, the evaluation of controllers on the X-coordinate motion is conducted. The PID controllers track the given trajectory and produce an ITAE of 107.88, IAE of 19.65, and ISE of 366.87. NFC achieves an ITAE of 65.61, IAE of 10.32, and ISE of 256.03. Finally, the proposed DQN achieves the minimum error compared to the other controllers, with ITAE, IAE, and ISE values of 48.24, 8.45, and 193.69, respectively. The error metrics attained by each controller for robot motion control in the X-coordinate are shown in Figure 3.

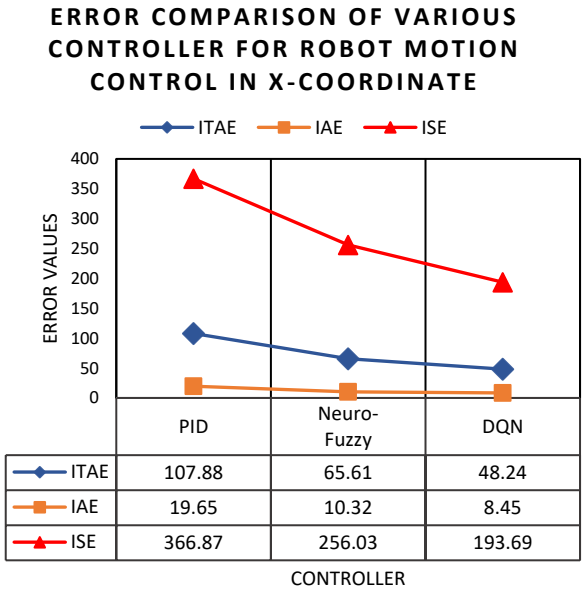

Fig. 3. Error plot of three controllers on X-coordinate motion control

Secondly, the evaluation of controllers on the Y-coordinate motion is conducted. Similar to the X-coordinate results, DQN performs well in Y-coordinate motion control. The ITAE, IAE, and ISE produced by the PID controllers are 95.42, 17.05, and 321.67, respectively. NFC achieves an ITAE of 58.21, IAE of 8.21, and ISE of 239.56. Finally, the proposed DQN achieves ITAE, IAE, and ISE values of 39.85, 5.33, and 176.26, respectively. The error metrics attained by each controller in the Y-coordinate robot motion control are given in Figure 4.

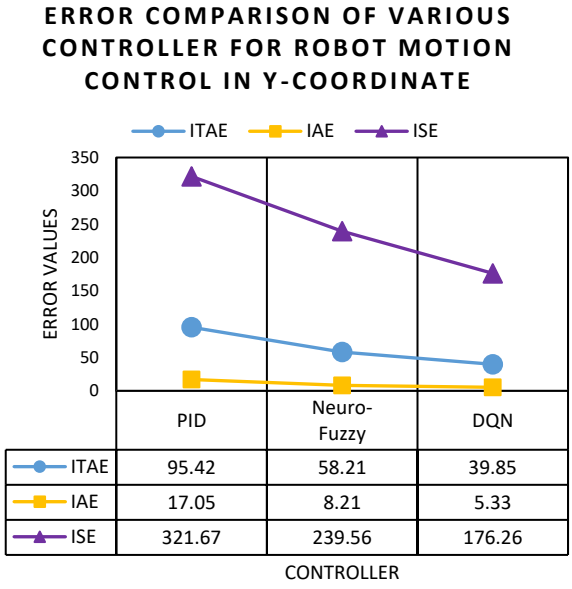

Fig. 4. Error plot of three controllers on Y-coordinate motion control

V. CONCLUSION

Robot motion control is an important area of research where the quality of work depends on precise control of robot motion. To achieve this, the mathematical model of the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

robot is derived. Next, the DQN controller is designed for robot motion control. The controller is responsible for tracking the x and y coordinates by providing control signals to the left and right wheels. The DQN controller is compared with PID and NFC controllers. All three controllers are tested by providing a reference trajectory as input, which contains the X and Y coordinates. The DQN effectively tracks the given input, yielding minimal ITAE, IAE, and ISE values of 48.24, 8.45, and 193.69 in x-coordinate tracking, and 39.85, 5.33, and 176.26 in y-coordinate tracking, respectively. These results demonstrate the promising potential of DQN in robot motion control applications. In the future, higher DOF robots will be considered, and controllers will be designed for their motion control.

REFERENCES

[1] Sa, Baskaran, Ramesh Kumar Tb, and Karrthik R. Sc. "Applications and future scope of robotics-a review." *International Journal of Robotics and Autonomous Systems* 3, no. 1 (2018): 12-26.

[2] Madridano, Ángel, Abdulla Al-Kaff, David Martín, and Arturo De La Escalera. "Trajectory planning for multi-robot systems: Methods and applications." *Expert Systems with Applications* 173 (2021): 114660.

[3] Zhang, Liwen, Song Zhao, Xinzhao Zhou, Xueshan Jing, Yu Zhou, Yan Wang, Yantong Zhu et al. "A magnetic-driven multi-motion robot with position/orientation sensing capability." *Research* 6 (2023): 0177.

[4] Thai, Nguyen Hong, Trinh Thi Khanh Ly, Hoang Thien, and Le Quoc Dzung. "Trajectory tracking control for differential-drive mobile robot by a variable parameter PID controller." *International Journal of Mechanical Engineering and Robotics Research* 11, no. 8 (2022): 614-621.

[5] Khan, Huma, Shahida Khatoon, Prerna Gaur, and Salman Ahmad Khan. "Speed control comparison of wheeled mobile robot by ANFIS, Fuzzy and PID controllers." *International Journal of Information Technology* 14, no. 4 (2022): 1893-1899.

[6] Abubakr, Omer Ali, Mohammad A. Jaradat, and Mamoun F. Abdel-Hafez. "Intelligent optimization of adaptive dynamic window approach for mobile robot motion control using fuzzy logic." *IEEE Access* 10 (2022): 119368-119378.

[7] Thai, Nguyen Hong, and Trinh Thi Khanh Ly. "Trajectory tracking control for mecanum wheel mobile robot by time-varying parameter PID controller." *Bulletin of Electrical Engineering and Informatics* 11, no. 4 (2022): 1902-1910.

[8] Hassan, Najva, and Abdul Saleem. "Neural network-based adaptive controller for trajectory tracking of wheeled mobile robots." *IEEE Access* 10 (2022): 13582-13597.

[9] Rojas, José David, Orlando Arrieta, and Ramon Vilanova. *Industrial PID controller tuning*. Springer International Publishing, 2021.

[10] Uddin, M. Nasir, and Hao Wen. "Development of a self-tuned neuro-fuzzy controller for induction motor drives." *IEEE Transactions on industry applications* 43, no. 4 (2007): 1108-1116.

[11] Mishra, Divyendu Kumar, Aby Thomas, Jinsa Kuruvilla, P. Kalyanasundaram, K. Ramalingeswara Prasad, and Anandakumar Haldorai. "Design of mobile robot navigation controller using neuro-fuzzy logic system." *Computers and Electrical Engineering* 101 (2022): 108044.

[12] Talpur, Noureen, Said Jadid Abdulkadir, Hitham Alhussian, Mohd Hilmi Hasan, Norshakirah Aziz, and Alwi Bamhdi. "Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: A systematic survey." *Artificial intelligence review* 56, no. 2 (2023): 865-913.

[13] Gholizadeh, Nastaran, Nazli Kazemi, and Petr Musilek. "A comparative study of reinforcement learning algorithms for distribution network reconfiguration with deep Q-learning-based action sampling." *IEEE Access* 11 (2023): 13714-13723.

[14] Marchesini, Enrico, and Alessandro Farinelli. "Discrete deep reinforcement learning for mapless navigation." In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10688-10694. IEEE, 2020.

[15] Bejani, Mohammad Mahdi, and Mehdi Ghatee. "A systematic review on overfitting control in shallow and deep neural networks." *Artificial Intelligence Review* 54, no. 8 (2021): 6391-6438.