

Integration of Q-Learning and PID Controller for Mobile Robots Trajectory Tracking in Unknown Environments



Almojtaba Munaf^{*}, Ahmed Rahman Jasim Almusawi^{}

Department of Mechatronics Engineering, University of Baghdad, Baghdad 10071, Iraq

Corresponding Author Email: almojtaba.manaf1702a@kecbu.uobaghdad.edu.iq

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.570410>

ABSTRACT

Received: 22 April 2024

Revised: 19 July 2024

Accepted: 5 August 2024

Available online: 27 August 2024

Keywords:

Q-learning, path planning, differential drive, mobile robot, trajectory tracking, reinforcement learning (RL), robotics, PID controller

In the realm of autonomous robotics, navigating differential drive mobile robots through unknown environments poses significant challenges due to their complex nonholonomic constraints. This issue is particularly acute in applications requiring precise trajectory tracking and effective obstacle avoidance without prior knowledge of the surroundings. Traditional navigation systems often struggle with these demands, leading to inefficiencies and potential safety risks. To address this problem, our studies propose an algorithm that integrates machine learning and control concepts, especially through the synergistic software of a Q-learning set of rules and a (PID) controller. This technique leverages the adaptability of Q-learning pathfinding and the precision of PID control for actual-time trajectory adjustment, aiming to beautify the robotics' navigation skills. Our comprehensive technique includes growing a country-area version that integrates Q-values with the dynamics of differential power robots, employing Bellman's equation for iterative coverage refinement. This version enables the robotics' capacity to dynamically adapt its navigation techniques in reaction to instant environmental feedback, thereby optimizing efficiency and protection in actual time. The effects of our full-size simulations exhibit a marked improvement in trajectory-tracking accuracy and impediment-avoidance competencies. These findings underscore the capability of combining machine learning algorithms with traditional methods to increase autonomous navigation technology in robotic systems. Our effects, derived from full-size simulations, suggest that the integration of Q-learning with PID controller markedly improves trajectory tracking accuracy, reduces tour times to targets, and complements the robotics' ability to navigate round barriers. This incorporated method demonstrates a tremendous advantage over conventional navigation systems, providing a sturdy way to the challenges of autonomous robot navigation in unpredictable environments.

1. INTRODUCTION

Recently, there was a developing consciousness on using machine learning (ML) strategies within the international of robotics because of the substantial importance it represents in the medical studies discipline. The trajectory monitoring procedure for robots has continually represented widespread importance, in particular in the modern-day era of rapid improvements. This is because of its big applications in numerous fields, together with agriculture, industry, surveillance, and plenty of others [1]. The wheeled mobile robotic is a complex mechanical system that doesn't observe conventional movement constraints, making it nonlinear, underactuated, and free of waft. These characteristics pose considerable demanding situations for controlling its movement. Among those challenges, ensuring that the robot accurately follows a predetermined course has emerged as a crucial vicinity of recognition in studies [2]. Wheeled mobile robots are very popular in commercial environments and customer spaces due to their simple shape, ease of modeling and control, flexible movement, and cost effectiveness in

every process and production. These robots can perform various duties, such as moving and sorting items, acting as medical robots, and providing mobility assistance to visually impaired individuals. The maximum, but not unusual, capabilities of the design are two powered wheels and one passive wheel, called robotic two-wheelers [3, 4].

Trajectory tracking control for a mobile robot includes making sure that the robot's cutting-edge position and orientation converge in the direction of a predetermined reference route. This direction can both be predefined or generated dynamically, which includes following the trajectory of a transferring virtual goal. The primary intention is to manual the cellular robotic accurately alongside the desired trajectory [5]. Traditional methods for trajectory tracking, such as PID controllers and path planning algorithms like A* and Dijkstra, rely on static models and predefined rules. PID controllers work by minimizing the error between the desired and actual positions through a fixed set of control parameters. While effective in stable and predictable environments, they lack the flexibility to adapt to dynamic and changing conditions. This can lead to difficulties when

encountering unexpected obstacles or variations in the environment, resulting in suboptimal performance and potential safety risks. The limitations of these traditional approaches highlight the need for more adaptive and responsive solutions. In dynamic and unpredictable settings, relying solely on predefined models and rules can result in inefficiencies and safety hazards. This is where advanced ML techniques, particularly RL, offer significant advantages. ML enables mobile robots to learn from their interactions with the environment, continuously refining their navigation strategies based on real-time feedback.

ML enables cell robots to examine from reveal in, enhancing their navigation through complicated environments. It encompasses Supervised Learning for prediction, unsupervised learning for pattern discovery, and RL for selection-making through trial and mistake, which is in particularly important for adaptive trajectory tracking. For precise navigation, ML uses strategies like Bayesian Filters for role prediction and Monte Carlo Methods for probabilistic route estimation. In comparison, conventional methods like PID controllers and path planning algorithms (e.g., A* and Dijkstra) provide dependent navigation solutions, relying on predefined fashions and policies. Combining ML's adaptability with conventional techniques' reliability equips mobile robots with both the dynamic Q-learning wanted for unsure environments and the consistency required for unique trajectory monitoring. This integrated approach offers sophisticated navigational capabilities, balancing innovation with balance [6].

Q-learning rule set is a type of reinforcement study and follows a Markov selection system. In this technique, the mobile robot starts out without any previous experience. It takes many moves based on strategy and study, getting exclusive rewards for each move, and striving to choose the move that yields the best reward. By interacting with its environment, the mobile robot collects feedback and uses this data to improve its choices, eventually achieving the gold standard average strategy [7].

In this paper, we introduce an advanced method that utilizes Q-mastering for pathfinding and eventually enhances navigation in a differential drive cell robotic through the mixing of PID controller. This two-step technique first applies Q-mastering to enable autonomous course identity, that specialize in overcoming trajectory monitoring and impediment avoidance challenges. We then increase this with PID control to refine the robotics' motion, aiming for smoother and more unique navigation. Our technique is tested through comparative simulations that examine the trajectory tracking performance of our Q-studying and PID hybrid model against that of a traditional fuzzy good judgment controller. The consequences spotlight the superiority of our method in unknown environments, showcasing greater performance and adaptability. This concise method not only advances independent navigation technologies but additionally underscores the capacity of mixing system to gaining knowledge of with control theory to address complex navigation demanding situations in unpredictable settings. The organization of the paper is succinctly laid out to ensure clarity and coherence: after reviewing pertinent literature in Section 2, we expound on our Q-learning based methodology and system design in Sections 3 and 4. Section 5 is dedicated to presenting the results and discussion, illustrating the significant advantages of our approach. Finally, Section 6 wraps up the paper with conclusions and pointers toward

future research avenues, emphasizing Q-learning's pivotal role in the enhancement of autonomous robotic navigation.

2. RELATED WORK

Current research in the field of mobile robot navigation focuses on employing ML techniques since they provide good performance enhancement. Introduced a navigation control algorithm for mobile robots based on Q-learning which divides the environment into discrete state spaces and maps actions to states while designing reward functions. The algorithm emphasizes the importance of intelligent abilities for robot navigation in unknown environments. Through computer simulations, they validate the feasibility of the algorithm, demonstrating its potential for effective robot navigation [8].

Method developed to enhance autonomous navigation for mobile robots in unpredictable settings by employing a dual Q-learning-based RL strategy for dynamic PID controller gain adjustments. It methodically incorporates active learning to efficiently navigate state and action domains. Through sizable simulation assessments on terrestrial, aerial, and aquatic robots, the method demonstrates marked superiority over conventional manage mechanisms, highlighting its robustness and adaptableness in varied environments [9]. Emphasizes the utility of reinforcement gaining knowledge of to remedy the difficulty of function manage in mobile robots. It provides a newly evolved reinforcement getting to know set of rules, particularly designed for this venture. The findings show the set of rule's success in allowing the robot to determine its position autonomously, the usage of a technique of trial and blunders knowledgeable by using remarks such as rewards and penalties [10]. Explored the deployment of RL in controlling the positioning of mobile robots. Utilizing the Q-learning algorithm, the framework is designed to refine movements through rewards derived from environmental interactions. The findings suggest that the RL framework surpasses conventional control methods, demonstrating superior efficiency in navigating to detailed targets. These consequences underscore the effectiveness of RL in improving the positional manage of cellular robots, indicating its promising applicability within the development of self sufficient navigation systems [11]. Brought an revolutionary method for dealing with a robot waiter in restaurants through a Q-getting to know-based totally adaptive PID controller. This technique overcomes the shortcomings of traditional PID controllers by using dynamically adjusting the manipulate settings to healthy changing conditions inside the system. With the implementation of dual-line sensors, the robot can navigate efficaciously in a restaurant placing. Both simulations and actual-international experiments display that this adaptive PID controller surpasses traditional PID controllers in terms of responsiveness, precision, and overall balance [12]. Focused on the utility of Neural Networks and Hierarchical Reinforcement Learning (HRL) to facilitate self reliant direction planning for cell robots. These methods triumph over the challenges faced via modern-day robotic systems, appreciably growing their ability to evolve to evolving environments and enhancing the speed of achieving optimal direction answers [13]. Concentrated at the approach for making plans paths for more than one robots, this studies introduces an innovative technique that makes use of Deep Q-Networks (DQN). The advised approach leverages DQN to

broaden a coverage community that buddies environmental situations with respective movements [14]. introduced a modified Q-learning algorithm for mobile robot path planning, incorporating a motivation model to influence reward values. Through simulations, the work demonstrates that the modified algorithm produces multiple safe path variations in diverse obstacle scenarios. Changes in motivation model variables impact reward achievement and subsequently influence Q-value updates. Despite slightly longer computation times compared to traditional methods, the modified algorithm improves path planning efficiency and safety. Overall, the approach enhances mobile robot navigation. by generating diverse and safe paths.

Our work integrates Q-learning with PID control for mobile robot navigation, focusing on trajectory tracking and obstacle avoidance, and combining ML adaptability with PID precision. In contrast, the study [15] specifically employs double Q-learning and backstepping to achieve precise trajectory tracking in complex environments, emphasizing adaptive control for enhanced accuracy. The main difference lies in our method's broad application scope versus the comparative study's focus on precision through advanced learning algorithms [16].

3. METHODOLOGY

In this work, the Q-learning algorithm is employed to develop and train a mathematical representation of a differential drive mobile robot, enabling it to navigate a path within an unfamiliar setting.

3.1 Q-learning algorithm

Q-learning is a ML method that allows a model to progressively enhance its performance by taking optimal actions iteratively. It falls within RL, where favorable actions receive rewards, and unfavorable actions incur penalties. The training process follows a state-action-reward- state action framework, guiding the model to make optimal decisions. Notably, there is no explicit model of the environment to direct the RL process. The agent, the AI component within the environment, autonomously learns and predicts its surroundings through iterative interactions. In the iterative process of Q-learning models, multiple components collaborate to train the model effectively. The agent learns through exploration of the environment, continually Adjusting the model based on ongoing exploration activities. Figure 1 shows various components of Q-learning which includes:

- Agents: The agent is the entity that acts and operates within an environment.
- States: A state is a variable that signifies the current location of an agent within an environment.
- Actions: Actions represent the operations or activities that the agent performs in a specific state within an environment.
- Rewards: In RL, rewards constitute a fundamental concept where the agent receives either a positive or negative response based on actions.
- Episodes: An episode occurs when an agent reaches a point where it can no longer take new actions and consequently terminates.
- Q-values: The Q-value is a metric employed to quantify the utility or effectiveness of an action taken at a specific state within a RL context.

•Bellman's equation: Bellman's equation is a recursive formula for optimal decision-making. In the context of Q-learning, this equation is utilized to calculate the value of a given state and assess its relative position, aiding in the optimization of decision-making processes.

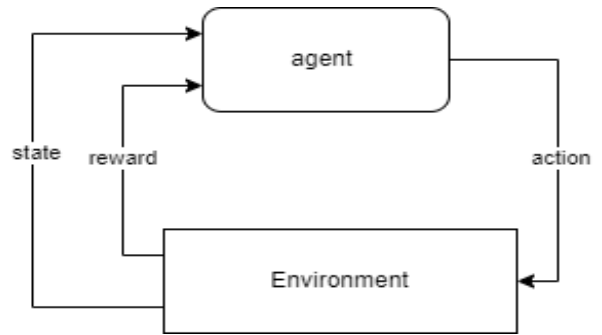


Figure 1. RL agent-environment interaction loop

3.2 The QL-based path planning method

The basic concept behind the QL-based path planning method involves the Q-learning algorithm. When reviewing the Q value of a case action pair, this technique includes the Q value of the subsequent case action pair generated by the policy under evaluation, as opposed to the Q value of the next case action pair committed to the current policy. In the context of path planning for a mobile robot, the algorithm involves randomly sampling the environment and generating paths through multiple sampling. Throughout this process, the interaction between the behavior policy and the goal policy continues to iterate until the optimal path is obtained. The learning process of the QL algorithm is explained in the Algorithm 1.

Algorithm 1. Q-learning algorithm [17]

- 1 Initialize $Q_{n \times m}(s, a) = 0$ (n state and m actions)
 - 2 Repeat
 - 3 Using ϵ -greedy to select a from present state s
 - 4 Take action a , get r, s'
 - 5 Update $Q(s, a)$ by (1)
 - 6 $s \leftarrow s'$
 - 7 Until s is a destination
-

The updating process described by Eq. (1) unfolds as follows:

$$Q(s, a) = Q(s, a) + \alpha * (r + \gamma \max_{a'} (Q(s', a') - Q(s, a))) \quad (1)$$

where, s is a state, a : is action, r is the reward that received a reinforcement signal after s is executed, s' is next state, γ ($0 \leq \gamma < 1$) is the discount factor, and α ($0 \leq \alpha < 1$) is the learning rate. Various methods have been employed to address the issues.

3.3 Differential drive mobile robot

The mobile robot platform depicted in Figure 2 features two driving wheels arranged in parallel, accompanied by one passive wheel strategically positioned to maintain the static stability of the robot. The common radius of the two driving

wheels is denoted as "r," and the separation between the two wheels is indicated by "L." To navigate the robot from one location to another, knowledge of both the position and orientation is essential. It should know the position and orientation to move the robot from one place to another [18]. The Pioneer 3DX mobile robot (Differential Drive) is designed with specific physical characteristics conducive to research and educational uses in robotics. It measures 44.5 cm in length, 39.3 cm in width, and stands 23.7 cm tall. The robot's lightweight design is reflected in its total weight of approximately 9 kg. It uses lidar sensor (lidar lite v3 model) which helps the robot to detect obstacles and avoid it.

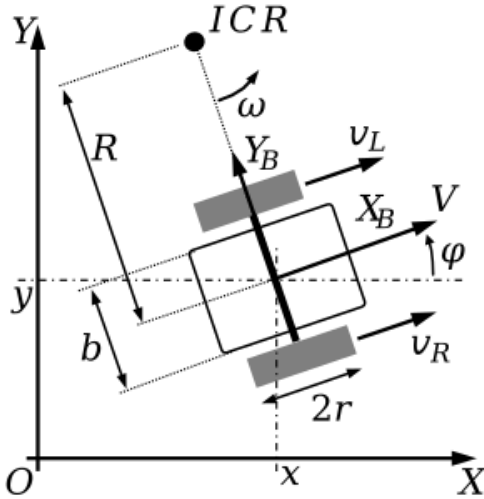


Figure 2. Schematic of differential drive mobile robot kinematics [19]

By changing the speeds of the two wheels, the paths taken by the robot can be changed. Given that the rate of rotation (ω) around the instantaneous center of curvature (ICC) must remain constant for both wheels, the following equation can be written as:

$$Vr = \omega \left(R + \frac{l}{2} \right) \quad (2)$$

$$Vl = \omega \left(R - \frac{l}{2} \right) \quad (3)$$

where, Vr and Vl represent the velocities of the right and left wheels along the ground, respectively. R is defined as the signed distance from ICC to the midpoint between the wheels, and ICC is identified as the Instantaneous Center of Curvature. At any given moment, R and ω can be solved as follows:

$$R = \frac{l}{2} \left(\frac{Vl + Vr}{Vr - Vl} \right) \quad (4)$$

$$\omega = \frac{(Vr + Vl)}{l} \quad (5)$$

when $Vl = Vr$ the robot's motion will be linear, proceeding in a straight line.

If $Vl = -Vr$ then $R = 0$ results in rotation around the center of the wheel axis - rotation in place occurs. If $Vl = 0$ rotates around the left wheel, with $R = -l/2$. The same principle applies if $Vr = 0$, $R = l/2$.

3.4 Kinematic modeling of differential drive robotic systems

In Figure 1, the robot's current position is represented by the coordinates (x, y) , and its orientation is indicated by an angle θ relative to the X-axis. The robot's center is considered to be the midpoint of the wheel axle. By adjusting the control parameters Vl and Vr , which are likely the velocities of the left and right wheels respectively, the robot can be directed to different positions and orientations. Knowing the velocities Vl and Vr , and using Eq. (3), the location of ICC can be determined.

$$ICC = [x - R \sin(\theta), y + R \cos(\theta)] \quad (6)$$

where, ICC is the center of rotation. And at time $t + \delta t$ the robot's pose will be:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \delta t) & -\sin(\omega \delta t) & 0 \\ \sin(\omega \delta t) & \cos(\omega \delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICCx \\ y - ICCy \\ \theta \end{bmatrix} + \begin{bmatrix} ICCx \\ ICCy \\ \omega \delta t \end{bmatrix} \quad (7)$$

This equation simply describes the motion of a robot rotating a distance R about its ICC with an angular velocity of ω (ICC) [20].

3.4.1 State space model of differential drive mobile robot

Figure 3 shows the complete block diagram of the system illustrates the desired movement of the robot in a specific direction (θ) with a velocity (V). The velocities of the left and right wheels, Vl and Vr , are input into the kinematic equation to determine the robot's linear and angular velocities. By integrating the angular velocity, the resulting orientation is fed back into the system as input, creating a closed-loop system that enhances stability. The state space model of this system is described as follows:

State variables

x : Position of the robot along the x-axis.

y : Position of the robot along the y-axis.

θ : Orientation of the robot.

The state vector is defined as:

$$X = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (8)$$

The control input vector is:

$$U = \begin{bmatrix} Vr \\ Vl \end{bmatrix} \quad (9)$$

Derive the state equations based on the robot's kinematics:

-Linear velocity v :

$$v = \frac{r}{2} (vr + vl) \quad (10)$$

-Angular velocity ω :

$$\omega = \frac{r}{L} (vr - vl) \quad (11)$$

where,

r is the radius of the wheels.

L is the distance between the wheels (wheelbase).

Use the linear and angular velocities to update the state variables over time:

Time derivatives of state variables:

$$\dot{x} = v \cos(\theta) \quad (12)$$

$$\dot{y} = v \sin(\theta) \quad (13)$$

$$\dot{\theta} = \omega \quad (14)$$

These equations represent how the robot's position and orientation change over time based on its wheel velocities.

$$X' = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{R}{2}(vr + vl) \cos \theta \\ \frac{R}{2}(vr + vl) \sin \theta \\ \frac{R}{L}(vr - vl) \end{bmatrix} \quad (15)$$

$$= \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2} & \frac{r}{2} \\ \frac{r}{L} & -\frac{r}{L} \end{bmatrix} \begin{bmatrix} vr \\ vl \end{bmatrix}$$

where, U is the control input vector and represents the velocities of the right and left wheels [21].

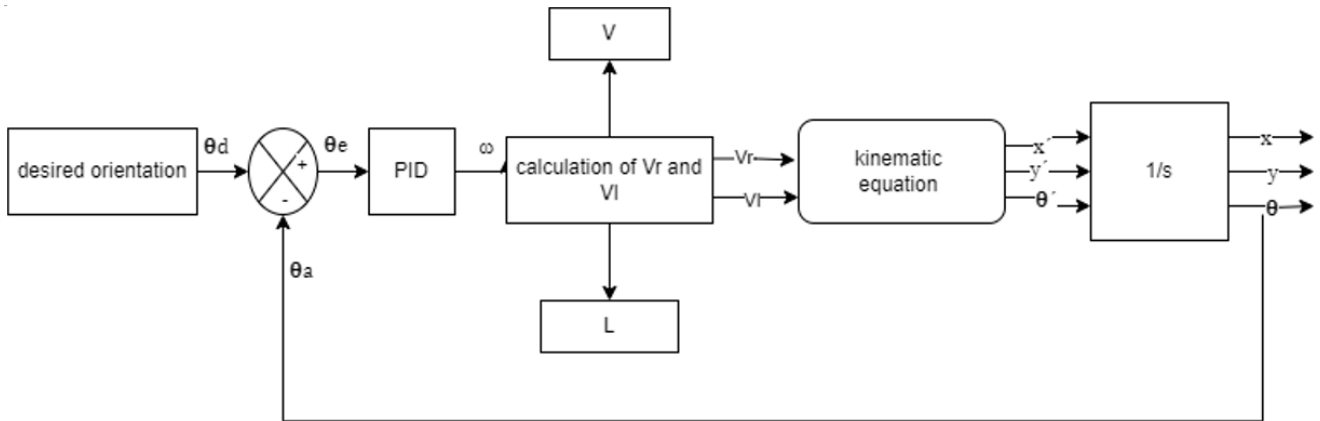


Figure 3. Control system overview for a differential drive robot

Linking to Q-learning:

In the Q-learning algorithm, the robot learns to navigate its environment by interacting with it and receiving feedback in the form of rewards. The state space model plays a critical role in this process by providing a structured representation of the robot's state and the effects of its actions.

Q-learning process:

State Representation

-The current state of the robot is represented by the state vector $X=[x,y,\theta]$.

-The environment is discretized into a grid or continuous state space, where each state represents a unique combination of x , y , and θ .

Action selection

At each time step, the robot selects an action based on its current state. Actions correspond to changes in the control input vector $U=[Vr,Vl]$.

The Q-learning algorithm uses an ϵ -greedy policy to balance exploration (trying new actions) and exploitation (selecting the best-known actions).

State transition

When an action is executed, the state space model equations are used to compute the new state X' .

The transition to the new state is governed by the control inputs and the robot's dynamics, as described by the state equations.

Reward calculation

The robot receives a reward based on the new state X'' . The reward function is designed to encourage desirable behaviors, such as moving towards the goal or avoiding obstacles.

3.4.2 Design of PID controller: Definition of the control variable U

The PID controller in your differential drive robot's design is a feedback mechanism that adjusts the control variable $u(t)$ to minimize the orientation error $e(t)$ between the desired angle $\theta_{desired}$ and the actual angle $\theta_{actual}(t)$. The controller is comprised of three terms:

-Proportional Term (K_p): This term produces an output that is proportional to the current error. The constant K_p determines the reaction to the current error; a higher K_p results in a larger response to errors.

-Integral Term (K_i): This term accounts for past errors by integrating (summing over time) the error value. The constant K_i scales the contribution of these accumulated errors, countering any bias or persistent error that is not eliminated by the proportional term alone.

-Derivative Term (K_d): This term considers the rate of change of the error, predicting future error based on its current rate. The constant K_d scales the influence of the rate of error change, helping to stabilize the control by damping oscillations that could arise from the proportional term.

The PID controller's effectiveness lies in its ability to address both the current state of the system (via the proportional term), its historical performance (via the integral term), and its future trends (via the derivative term), which can substantially improve the precision and stability of the robotic system.

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d d/dt e(t) \quad (16)$$

where, $e(t) = \theta_{desired} - \theta_{actual}(t)$ is the orientation error at time t .

K_p , K_i , and K_d are the proportional, integral, and derivative gains, respectively.

3.5 Fuzzy controller

A fuzzy controller is a type of algorithm used in control systems that emulates the process of human decision-making, using 'fuzzy' logical reasoning rather than binary (on/off) logic. It is based on fuzzy logic - a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than fixed and exact. In fuzzy control systems, input variables are processed by a set of rules (the fuzzy logic ruleset) and interpreted using membership functions, which define how each point in the input space is mapped to a degree of membership between 0 and 1. These rules and membership functions describe the behavior of the system linguistically, allowing the system to handle vague and noisy information effectively. Fuzzy controllers are used in trajectory tracking for mobile robots due to their ability to handle the uncertainties and inaccuracies inherent in real-world environments. In trajectory tracking, the robot needs to follow a predetermined path, but factors such as wheel slip, irregular terrain, and unexpected obstacles can cause deviations. The fuzzy controller interprets sensor inputs in a way that mimics human response to such uncertainties, allowing for smooth corrections that keep the robot on its intended path. Moreover, fuzzy controllers do not require a precise mathematical model of the system, making them easier to implement and adapt to different systems. They excel in systems where responses to changing dynamics are required in real-time. This adaptability and robustness make fuzzy controllers highly suitable for the complex task of trajectory tracking in differential drive mobile robots, where precision and flexibility are paramount [22].

4. PROPOSED SYSTEM DESIGN

Due to the critical role of robot path planning and control across various industries and everyday life applications, especially in light of rapid global advancements and the pursuit of simplifying human life through efficient alternatives, this research endeavors to explore a mathematical model for a differential drive mobile robot utilizing the Q-learning algorithm. This algorithm is chosen for its ability to operate without prior knowledge of the robot's environment, incorporating the application of the Bellman equation. Through trial-and-error techniques, the algorithm trains the robot, offering four possible movements (up, down, right, left) that represent the actions in Q-learning. The robot was initially trained using multiple scenarios, starting from a simple environment free of any obstacles. Then, the environment was developed by adding specific obstacles in certain places to study the robot's behavior and observe its maneuvers to avoid obstacles. Afterward, new obstacles were placed in the paths taken by the robot to reach the goal. The robot succeeded in avoiding these obstacles as well. Then, the environment was further developed by adding obstacles that constitute 0.25 of the environment's volume randomly, and they change with each operation. The robot succeeded each time in avoiding the obstacles and reaching the goal by relying on Q-learning.

The primary objective of this study is to analyze the state

and elucidate how the robot selects its path, followed by studying the effect of adding a PID controller to control the theta of a head to improve the maneuver of the robot in avoiding obstacles.

4.1 Testing differential drive trajectory tracking using Q-learning algorithm

The differential drive mobile robot is trained to find the best path from the starting point to the target point using the Q-learning algorithm, which acts as a guide for the robot. This is achieved through a quick scan of the map, which changes with each new execution process. The path is executed through four possible movements: up, down, right, and left. The robot's movement is chosen based on the presence of obstacles in its path, selecting the direction that leads the robot along the shortest path. The Figures 4-8 display the results of the training. In Figure 4, the path taken by the robot from the starting point to reach the goal while avoiding obstacles is depicted. Then, in Figure 5, the linear velocity and angular velocity of the robot's movement are illustrated. Figure 6 demonstrates the right wheel and left wheel velocities that determine the direction of the robot's movement. Finally, in Figure 7, the theta of avoidance for the robot is shown.

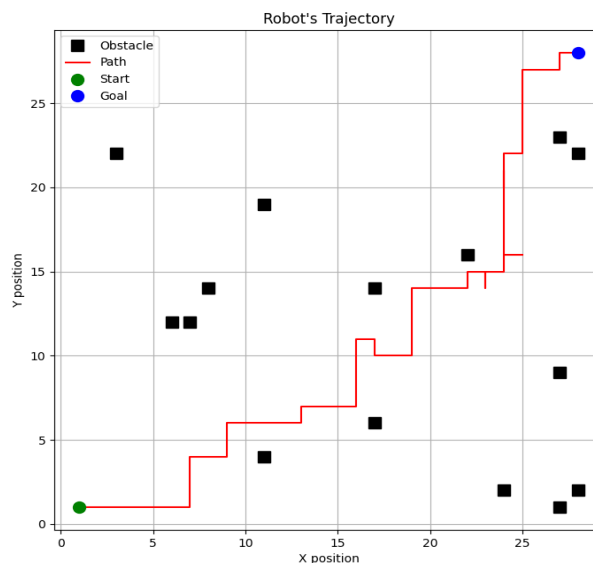


Figure 4. Q-learning based trajectory tracking in action

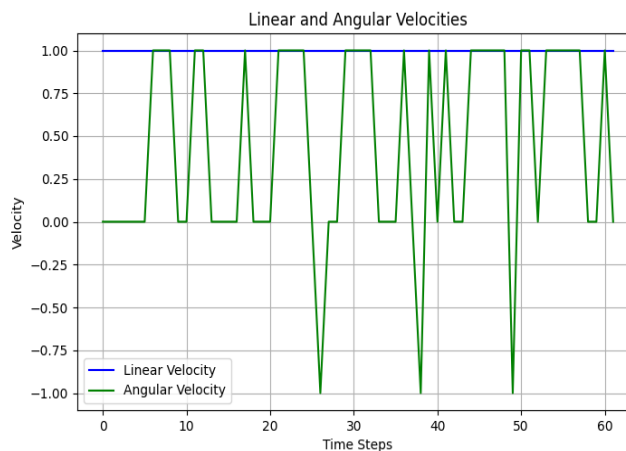


Figure 5. Robot's linear and angular velocity profiles.

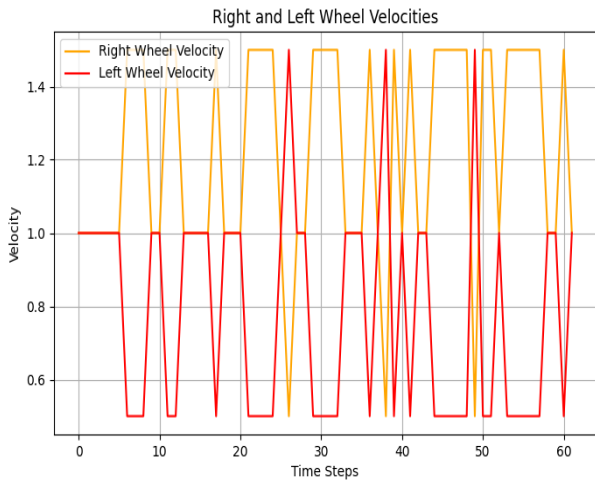


Figure 6. Differential wheel velocities for robot motion

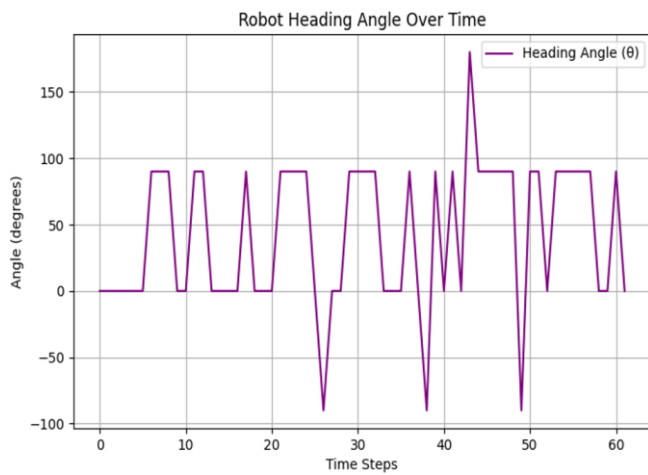


Figure 7. Orientation (Theta) adjustments of the robot's heading

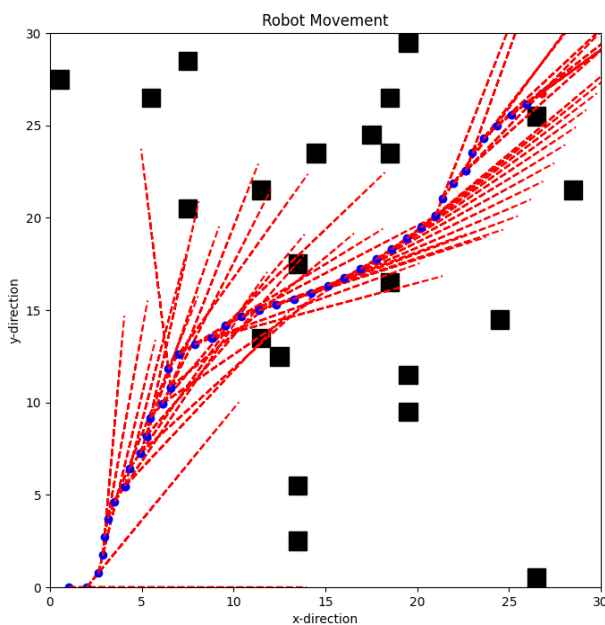


Figure 8. Enhanced trajectory tracking with Q-learning and PID

The behavior of the robot, when utilizing the Q-Learning

algorithm, demonstrates linear progression. Upon executing any of the four directional commands (upwards, downwards, to the right, or to the left), it advances at its highest linear speed, with its angular speed remaining at zero. Specifically, during forward movement, it achieves peak linear speed without any angular velocity change. Turning to the right or left results in an angular velocity of +90 or -90, respectively. This suggests that while the Q-Learning algorithm effectively directs the robot toward its goal, the capability for agile maneuvering is somewhat limited.

4.2 Training differential drive using Q-learning algorithm with PID controller

The integration of Q-learning with the PID controller enhances the robot maneuverability and obstacle avoidance capabilities. In our proposed navigation system, Q-learning is employed as path planning algorithm, guiding the robot through the environment by detecting obstacles and determining the optimal path to avoid them. The PID controller complements this by fine-tuning the robot's movement, ensuring that it accurately the desired path. The robot's maneuverability in avoiding obstacles has been improved by adding a PID controller. This controller regulates the heading angle (theta) based on Lidar sensor readings that detect obstacles and their distances. The PID controller adjusts the heading angle (theta) to optimize the robot's path, making it more suitable for avoiding obstacles. The Q-learning algorithm allows the robot to learn from its interactions with the environment, optimizing its pathfinding strategy based on real-time feedback from sensors. This learning process enables the robot to navigate efficiently and adapt to dynamic conditions. When obstacles are detected by Lidar sensors, the Q-learning algorithm updates the robot's trajectory to avoid collisions, effectively guiding the robot around obstacles. To enhance the precision of the robot's movements, the PID controller is introduced to regulate the heading angle (theta). By calculating the difference between the desired path and the actual path (recorded as theta), the PID controller adjusts the robot's heading angle to minimize this error. This adjustment ensures smoother and more accurate navigation, making the robot's path more suitable for avoiding obstacles while maintaining the intended trajectory.

Algorithm 2 explains how the robot works when applying Q-learning with PID and sensors. Figures 8-11 illustrate the results that appeared in the simulation process. Figure 8 shows the path taken by the robot to reach the goal after avoiding obstacles, where it appears that the robot followed a circular path to avoid obstacles due to the influence of adding PID. Figure 9 indicates the linear and angular velocity of the robot, while Figure 10 shows the velocity of the right and left wheels. Figure 11, on the other hand, represents the theta of avoidance.

Algorithm 2. Robot autonomous navigation

1. Start
 2. Collect data from environment sensors
 3. Repeat: Process sensor data and navigate
 - 3.1. Get sensor readings
 - 3.2. Calculate movement from kinematics
 - 3.3. Apply Q-learning for control optimization
 - 3.4. Calculate wheel velocities
 - 3.5. Adjust movement with PID control
 - 3.6. Output updated robot position and orientation
 4. End upon target reach or task completion
-

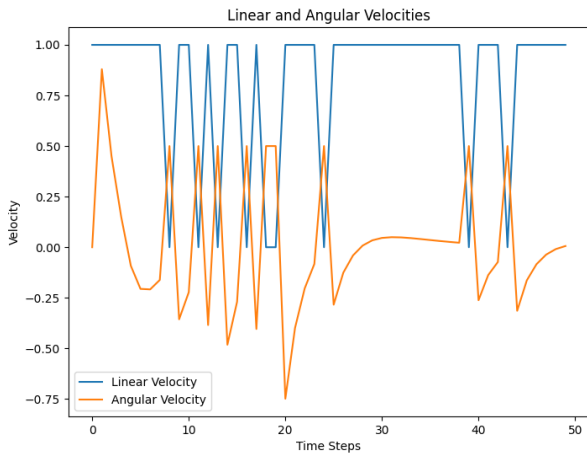


Figure 9. Improved velocity profiles of Q-learning with PID control integration

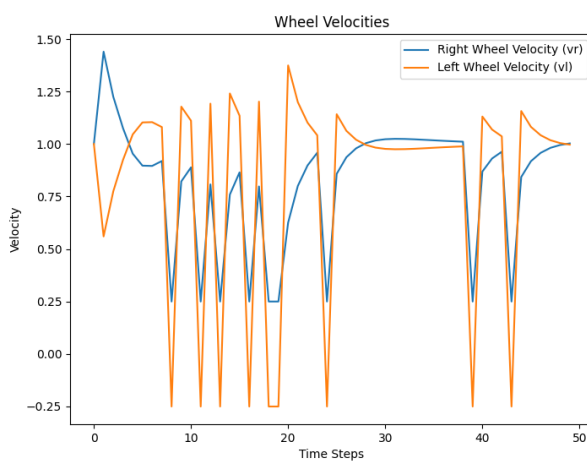


Figure 10. Wheel velocity optimization using Q-learning PID control

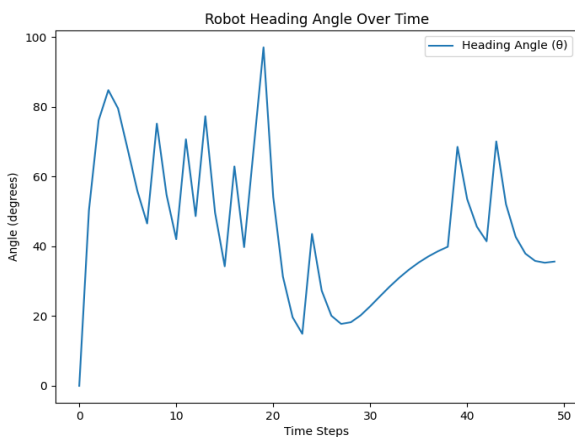


Figure 11. Heading orientation optimization with PID control

By combining Q-learning with PID control, our system leverages the strengths of both approaches. Q-learning provides the adaptability and learning capability needed for effective path planning in dynamic environments, while the PID controller ensures precise and smooth movement, enhancing the overall performance and reliability of the robot's navigation system.

5. RESULTS AND DISCUSSION

This work undertook a comparative analysis of two distinct control strategies for a differential drive mobile robot: path planning using solely the Q-learning algorithm and an integrated approach combining Q-learning for pathfinding with a PID controller for real-time heading adjustment. The efficacy of these methodologies was rigorously evaluated across a series of simulations, with a focus on trajectory tracking, obstacle avoidance, and computational efficiency. Figures 4-11 provide a visual representation of the robot's performance under each control strategy. The strengths of the paper are in adaptive learning, precision and stability, versatility, and robust obstacle avoidance while the limitations are parameter tuning, environmental dependence and complexity and computational load. While the potential areas are hybrid approaches like explore combining different algorithms and advanced learning algorithms.

5.1 Q-learning pathfinding

Q-learning showcased strong pathfinding in unknown environments, optimizing the robot's route efficiently. Figures 4-6 demonstrate the robot's trajectory optimization, indicating Q-learning's adaptability without needing a predefined environmental model. This method proves especially useful in unpredictable settings.

5.2 Integration of Q-learning and PID controller

Adding a PID controller to Q-learning significantly enhanced navigation and obstacle avoidance, as seen in Figures 8-11. This method adjusted the robot's heading in actual time, taking into consideration smoother impediment navigation. The stepped forward linear and angular velocities (Figures 9 and 10) and the maintained heading attitude (Figure 11) suggest an extra responsive and controlled navigation gadget. Integrating Q-learning with a PID controller improves robot navigation in unknown environments, supplying a combination of adaptability, precision, and safety. Future studies ought to amplify in this hybrid approach, exploring further upgrades for autonomous robotics.

5.3 Comparative analysis of trajectory tracking control strategies: Q-learning with PID vs. fuzzy controller

This section provides a detailed comparison among distinct management strategies implemented to a differential drive cellular robotic for trajectory tracking and impediment avoidance: a hybrid Q-learning with a PID controller and a fuzzy logic controller. Our analysis is grounded on empirical records illustrated in Figures 12-17, focusing on their performance in unknown environments where no predefined models of the environment exist.

Q-learning with PID controller: The integration of Q-learning with a PID controller demonstrates advanced performance in navigating via environments weighted down with limitations. The adaptive nature of Q-learning, blended with the perfect manipulate afforded by way of the PID mechanism, lets in the robot to alter its trajectory dynamically. This approach no longer best allows direct and efficient paths to the target but also correctly circumvents obstacles. Figures 4-11 showcase the robotic's adept maneuverability, preserving high course efficiency and strong obstacle avoidance abilities.

Trajectory tracking: As depicted in Figure 12, the robotic

constantly follows the optimized route closer to the purpose, with minimum deviation. This is attributed to the PID controller's effective handling of the robotic's heading, significantly enhancing trajectory alignment.

Velocity profiles: Figures 13 and 14 highlight the robot's linear and angular velocities, respectively, underscoring the PID controller's position in preserving constant development even as adeptly managing turns and obstacle avoidance maneuvers.

Wheel velocities: The differential velocities of the wheels (Figures 14 and 15) further confirm the robot's agile response to environmental changes, facilitating smooth navigation around obstacles.

Fuzzy logic controller: Conversely, the bushy logic controller, whilst excelling in environments requiring nuanced selection-making under uncertainty, exhibits obstacles in impediment-wealthy situations. The controller's tendency closer to cautious and gradual changes consequences in much less direct paths and, in a few instances, difficulties in reaching the purpose inside complex obstacle fields.

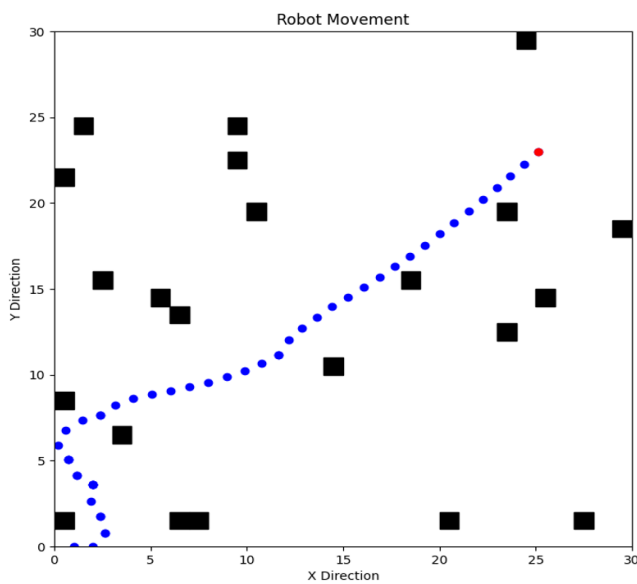


Figure 12. Optimized robot trajectory tracking using Q-learning combined with PID control

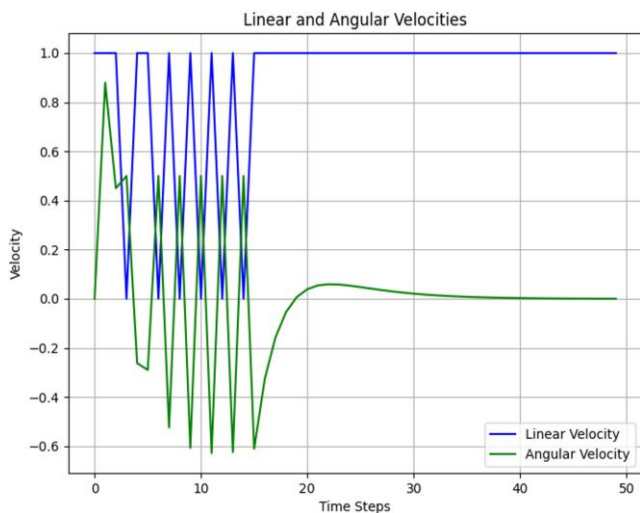


Figure 13. Robot's linear and angular velocity with Q-learning and PID control

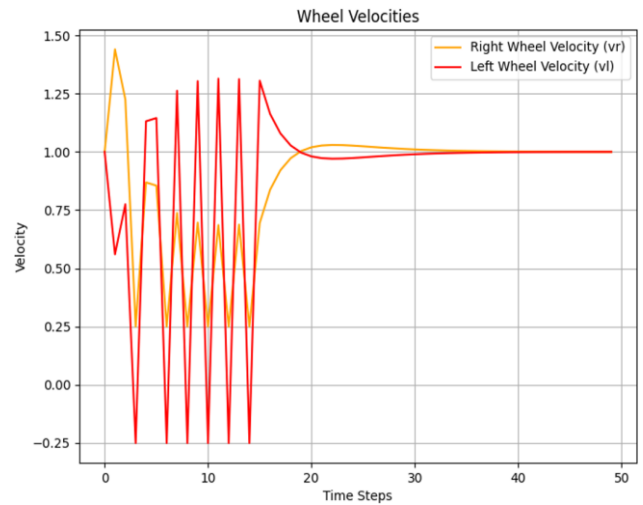


Figure 14. Comparative right and left wheel velocities with PID enhancement

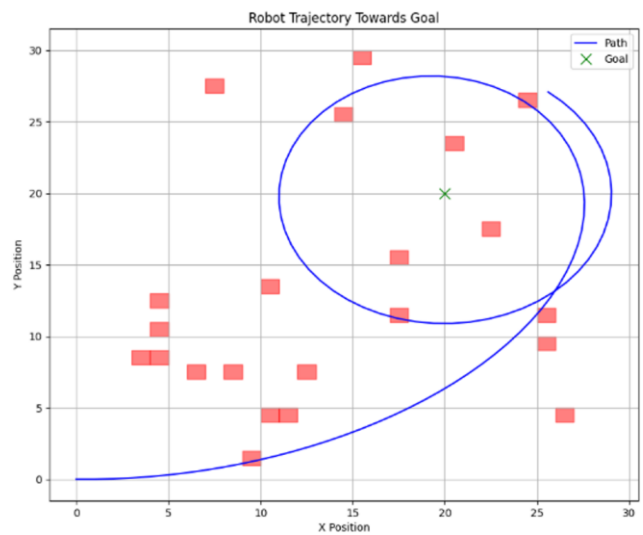


Figure 15. Robot trajectory tracking using fuzzy logic control

Trajectory tracking: The trajectories produced through the fuzzy controller, as proven in Figure 15, reveal a greater circuitous route toward the purpose. This behavior suggests a prioritization of protection over directness, main to prolonged travel times and capability challenges in attaining the purpose amidst dense limitations.

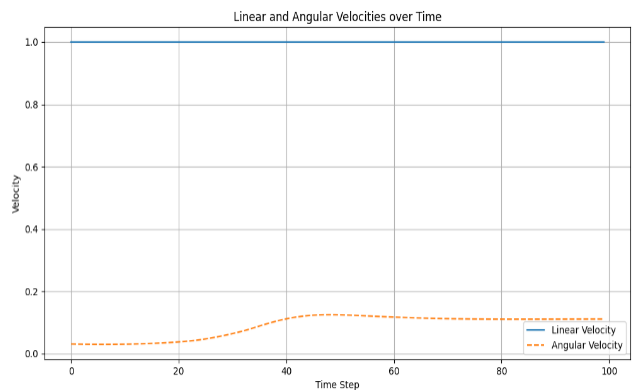


Figure 16. Linear and angular velocity profiles under fuzzy logic control

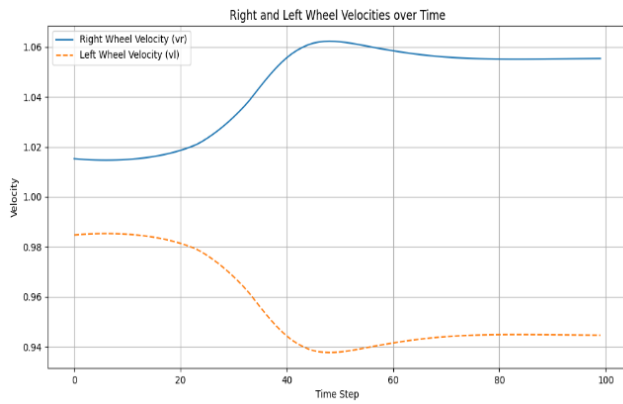


Figure 17. Differential wheel velocities using fuzzy logic control

Velocity profiles: Figures 16 and 17 illustrate the velocities under fuzzy control, depicting a more conservative approach to speed adjustments. This conservative stance aids in navigating through uncertain terrains but at the cost of directness and efficiency.

Wheel velocities: The comparative analysis of wheel velocities further highlights the fuzzy controller's cautious navigation strategy, potentially impeding direct goal attainment in obstacle-dense environments.

The comparative analysis underscores the distinct advantages and limitations of each control strategy in unknown environments. The Q-learning with PID controller emerges as a more robust solution for achieving efficient trajectory tracking and effective obstacle avoidance. It optimizes path efficiency and adapts more dynamically to environmental changes, making it highly suitable for applications requiring precision and adaptability. On the other hand, the fuzzy logic controller, with its inherent caution and gradual decision-making process, might be more appropriate in scenarios where safety and meticulous navigation are paramount, albeit with enhancements needed to improve its performance in complex obstacle scenarios.

6. CONCLUSION

These paintings marks a extensive development inside the field of independent navigation for differential drive cellular robots, demonstrating the powerful integration of Q-learning with PID manage mechanisms. By juxtaposing this hybrid version against a fuzzy good judgment controller, we have unearthed essential insights into trajectory tracking and impediment avoidance strategies in environments without prior knowledge.

Our research exhibits that the aggregate of Q-gaining knowledge of for pathfinding with the precision of PID manipulate for heading adjustments substantially complements the robot's ability to navigate complicated terrains. This combination now not best optimizes route efficiency however also guarantees dynamic adaptability in actual-time to unforeseen barriers, presenting a superior navigation answer in comparison to standard fashions. The Q-learning with PID controller model showcases a strong capacity for direct and green path selection, making sure the robotics' a hit arrival at its target with minimum deviations and inside the shortest viable time.

Conversely, whilst the bushy controller demonstrates a

commendable capability to handle obstacle avoidance, its trajectory closer to the goal is marked by using greater indirectness, leading to extended tour instances and less efficiency in accomplishing the vacation spot. This discrepancy highlights the enhanced performance and adaptableness of the Q-studying with PID version in dynamically changing and unpredictable environments. The findings from this look at emphasize the ability of integrating machine studying techniques with control structures to revolutionize independent navigation in cell robots. As we move ahead, in addition exploration into various system studying fashions and their synergies with diverse manipulate algorithms is crucial. This exploration will not handiest refine the autonomy and efficiency of robotic systems but also open avenues for tailor-made packages throughout distinctive sectors along with healthcare, agriculture, and industrial automation. In conclusion, our studies contributes a great jump closer to expertise and improving self sustaining navigation in robotics. It underscores the transformative effect of combining Q-mastering with PID control mechanisms over conventional fuzzy logic controllers, setting a brand new benchmark for destiny traits in robotic navigation systems. As the sector progresses, the continuing fusion of artificial intelligence with advanced manipulate strategies will undoubtedly bring in a new generation of tremendously sophisticated and self sustaining robot abilities.

REFERENCES

- [1] Zhang, S., Wang, W. (2019). Tracking control for mobile robot based on deep reinforcement learning. In 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, pp. 155-160. <https://doi.org/10.1109/ICoIAS.2019.00034>
- [2] Zheng, L. (2022). Predictive control of the mobile robot under the deep long-short term memory neural network model. *Computational Intelligence and Neuroscience*, 2022(1): 1835798. <https://doi.org/10.1155/2022/1835798>
- [3] Matraji, I., Al-Durra, A., Haryono, A., Al-Wahedi, K., Abou-Khousa, M. (2018). Trajectory tracking control of skid-steered mobile robot based on adaptive second order sliding mode control. *Control Engineering Practice*, 72: 167-176. <https://doi.org/10.1016/j.conengprac.2017.11.009>
- [4] Jasim, F.M., Al-Isawi, M., Hamad, A.H. (2022). Guidance the wall painting robot based on a vision system. *Journal Européen des Systèmes Automatisés*, 55(6): 793-802. <https://doi.org/10.18280/jesa.550612>
- [5] Trojnecki, M., Dąbek, P. (2019). Mechanical properties of modern wheeled mobile robots. *Journal of Automation Mobile Robotics and Intelligent Systems*, 13(3): 3-13. <https://doi.org/10.14313/JAMRIS/3-2019/21>
- [6] Urbaneck, D., Rehlaender, P., Schafmeister, F., Boecker, J. (2020). LLC converter design in capacitive operation utilizes ZCS for IGBTs—A concept study for a 2.2 kW automotive DC-DC stage. In *PCIM Europe digital days 2020; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management*, Germany, pp. 1-8.
- [7] Jiang, Q. (2022). Path planning method of mobile robot based on Q-learning. *Journal of Physics: Conference Series*, 2181(1): 012030. [1032](https://doi.org/10.1088/1742-

</div>
<div data-bbox=)

- 6596/2181/1/012030
- [8] Yang, S., Li, C. (2017). Behavior control algorithm for mobile robot based on q-learning. In 2017 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, pp. 45-48. <https://doi.org/10.1109/ICCNEA.2017.26>
- [9] Carlucho, I., De Paula, M., Acosta, G.G. (2019). Double Q-PID algorithm for mobile robot control. *Expert Systems with Applications*, 137: 292-307. <https://doi.org/10.1016/j.eswa.2019.06.066>
- [10] Kolodenkova, A., Vereshchagina, S. (2019). Diagnostics of industrial electrical equipment using modern information technologies. In 2019 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Sochi, Russia. <https://doi.org/10.1109/ICIEAM.2019.8742994>
- [11] Farias, G., Garcia, G., Montenegro, G., Fabregas, E., Dormido-Canto, S., Dormido, S. (2020). Position control of a mobile robot using reinforcement learning. *IFAC-PapersOnLine*, 53(2): 17393-17398. <https://doi.org/10.1016/j.ifacol.2020.12.2093>
- [12] Thanh, V.N., Vinh, D.P., Nam, L.H., Nghi, N.T., Le Anh, D. (2020). Reinforcement Q-learning PID controller for a restaurant mobile robot with double line-sensors. In Proceedings of the 4th International Conference on Machine Learning and Soft Computing, Haiphong City, Viet Nampp, 164-167. <https://doi.org/10.1145/3380688.3380718>
- [13] Yu, J., Su, Y., Liao, Y. (2020). The path planning of mobile robot by neural networks and hierarchical reinforcement learning. *Frontiers in Neurorobotics*, 14: 63. <https://doi.org/10.3389/fnbot.2020.00063>
- [14] Gu, Y., Zhu, Z., Lv, J., Shi, L., Hou, Z., Xu, S. (2023). DM-DQN: Dueling Munchausen deep Q network for robot path planning. *Complex & Intelligent Systems*, 9(4): 4287-4300. <https://doi.org/10.1007/s40747-022-00948-7>
- [15] Hidayat, H., Buono, A., Priandana, K., Wahjuni, S. (2023). Modified Q-learning algorithm for mobile robot path planning variation using motivation model. *Journal of Robotics and Control*, 4(5): 696-707. <https://doi.org/10.18196/jrc.v4i5.18777>
- [16] He, N., Yang, Z., Fan, X., Wu, J., Sui, Y., Zhang, Q. (2023). A self-adaptive double q-backstepping trajectory tracking control approach based on reinforcement learning for mobile robots. *Actuators*, 12(8): 326. <https://doi.org/10.3390/act12080326>
- [17] Ma, T., Lyu, J., Yang, J., Xi, R., Li, Y., An, J., Li, C. (2022). CLSQL: Improved Q-learning algorithm based on continuous local search policy for mobile robot path planning. *Sensors*, 22(15): 5910. <https://doi.org/10.3390/s22155910>
- [18] Khan, M.A. (2020). Design and control of a robotic system based on mobile robots and manipulator arms for picking in logistics warehouses. Doctoral dissertation, Normandie Université.
- [19] Hasan, H. M., & Mohammed, T. H. (2017). Implementation of mobile robot's navigation using slam based on cloud computing. *Engineering and Technology Journal*, 35(6A): 634-639. <https://doi.org/10.30684/etj.35.6a.11>
- [20] Allen, P. (2013). CS W4733 NOTES-Differential Drive Robots. Columbia University: Department of Computer Science, Evido Váno.
- [21] Kothandaraman, K. (2016). Motion planning and control of differential drive robot. Master thesis, Wright State University.
- [22] Štefek, A., Pham, V.T., Krivanek, V., Pham, K.L. (2021). Optimization of fuzzy logic controller used for a differential drive wheeled mobile robot. *Applied Sciences*, 11(13): 6023. <https://doi.org/10.3390/app11136023>