Journal of Robotics and Control (JRC)
Volume 5, Issue 6, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i6.23850

Autonomous Robotic Systems with Artificial Intelligence Technology Using a Deep Q Network-Based Approach for Goal-Oriented 2D Arm Control

Murad Bashabsheh [1*]
Department of Robotics and Artificial Intelligence, Jadara University, Irbid, Jordan
Email: [1] m.bashabsheh@jadara.edu.jo
*Corresponding Author

***Abstract*—Accurate control robotic arms in two-dimensional environments present significant challenges, particularly in dynamic, real-time applications. Traditional model-based approaches require substantial system modeling, rendering them computationally extensive. This paper presents an adaptive Artificial Intelligence (AI)-driven approach through the use of Deep Q-Networks (DQN) control for a two–link robotic arm thus supporting better scalability. The DQN algorithm, a model-free Reinforcement Learning (RL) technique, allows the robotic arm to independently learn optimal control strategies by interaction with the environment and adapting to dynamic conditions. The task of the robot established reaches a specific target (red point) within a limited number of episodes. Key components of the methodology contain problem statement, DQN architecture, representation of the state and action spaces, a reward function, and the training process. Experimental results indicate that the DQN agent effectively learns to find optimal actions with high accuracy and robustness in guiding the arm to the target. The performance steadily improves during initial training, followed by stabilization, indicating an effective control policy. This study contributes to the knowledge of reinforcement learning in robotic control tasks and demonstrates, in particular, the potential of DQN for solving complex, goal-oriented tasks with minimal prior modeling. Compared to conventional control approaches, the DQN-driven one reveals higher flexibility, scalability, and efficiency. Although carried out in a simplified 2D environment, the novelty of this research lies in its emphasis on enabling the robotic arm to accomplish goal-oriented reaching tasks, lays a strong foundation for future applications in industrial automation and service robotics.**

***Keywords*—Artificial Intelligence (AI); Autonomous Robotic Systems; Robotic Arm; Deep Q Network (DQN); Reinforcement Learning (RL); Model-Free Control; Goal-oriented Control.**

I. INTRODUCTION

Artificial Intelligence (AI) that describe computers as devices that depend on humans. Other activities include perception, learning, and cognition, comprehension of natural language, speech recognition and vision. In various contexts, people use AI to attempt at emulating the way human brains work to make complex tasks easier. Although the overall concept of AI includes a wide range of technologies, it necessitates many definitions in different businesses. Engineers describe AI as constructing robots that can execute jobs that humans would typically be able to achieve [1]. Regardless of their sophistication, it is critical to understand that these robots and programs lack actual human intelligence; rather, they demonstrate intelligent behaviors [2]. The idea of intelligence remains relative, as no system can reach complete intelligence. The intelligence of systems can vary depending on their ability to gather knowledge, reorganize information, and adapt to changing circumstances [3].

AI entails employing technology systems to execute activities that mirror human cognitive capacities. The basic aspect of AI is machine learning. This refers to the ability of a computer system to simulate intelligent behavior. More specifically, AI is the study and appropriate application of computer systems that were able to perform tasks previously based on human intelligence. These tasks include speech recognition, visual perception, decision making, language translation, and data analysis [4][5]. AI is established in the modern world as innovations that are improving people's quality of life. Autopilot systems, telemedicine, chatbots, Big Data, smart home, automated monitoring, as well as some of the AI fields including cyber justice, education, and defense have become more apparent. Consequently, with daily interactions of the practical tasks, the AI-powered devices undergo modifications that reveal the learning capabilities that would make the processes more efficient [6][7].

The rapid advancement of AI, machine learning, robotics, and automation is driving profound transformations in industries and societies worldwide. These changes are poised to revolutionize how we work, live, and interact with one another, surpassing anything seen in human history in terms of speed and scale. While this new industrial revolution holds the promise of enhancing and improving our lives and societies, it also carries the potential for significant disruptions to our way of life and societal norms. The window for understanding the impact of these technologies and mitigating their negative effects is rapidly closing. Humanity must adopt a proactive approach to managing this new industrial revolution rather than merely reacting to its consequences [8]-[11].

The importance of technological automation through AI is picking up across development sectors such as in banking, data analysis, healthcare, marketing among others. This development triggers questions concerning the impacts that AI is likely to have over the industry, consumer and the global economy at large [12].

Employees are becoming more and more interested in the effects that AI will have on their employment and wages. In more existing and potentially areas, the ability of AI automation to revolutionize manufacturing is now well understood, and numerous companies across numerous industries on the planet anticipate that AI in manufacturing will address central global issues. Driven automation came around has many advantages for the corporations namely increased tendency towards optimization of the process as well as overall increase in production and stability of the final quality of the products. The benefit of employing artificial intelligence in manufacturing is that manufacturers may benefit from the technology to improve performance and efficiency. Moreover, such solutions allow the creation of smart factories with the flexibility of manufacturing, by which manufacturing can be adapted immediately according to the demand [13][14].

Apart from cost-effectiveness, AI automation presents an opportunity for creating solutions for factors of sustainable development and efficient use of limited resources in the world. By using AI industries enforced environmentally conscious approach, minimized environmental impacts, and maximized efficient utilization of the resources. Further, AI-based systems can solve the problem of personnel deficit and skills mismatch by supplying human employees with devices and programs that increase personnel productivity [15]. Automation systems are composed of a network of actuators, controllers, and sensors through which tasks are performed with little human input. Most, of these systems have developed from manually operated including welding, drilling and cutting, in which robotic arms are used to manipulate tools that perform the operations [16]-[21].

In process control which is the third layer, automation systems are applied to control actuating parameters for devices such as motors, pumps, heaters, and compressors. These systems are available in many forms, and some are specially developed for a single purpose. Many procedures, including but not limited to cutting, milling, threading, welding, and inspection, employ automation. While it may be used to increase efficiency and improve the product value added, the primary goal of automation is to eliminate the operators from the process so that the process is less likely to be affected by their shortcomings. Through automation, the probity of the company can improve alongside with attainability, security, and quality of production [22][23]. In addition to automation, the capabilities of AI are also hailed for unprecedented ability to revolutionize the manufacturing industry as well as other operations around the world and solving some of the world's biggest challenges [24].

In recent years, the innovations and new technologies coupled with their proliferation across various sectors, have hastened the pace of digital transformation. The combined use of automation and AI is likely to result in innovative business models and advanced technologies with enhanced productivity in all sectors of the economy. With the continuous improvement of robotic technologies, the practical utilization of AI is becoming clearer. Such sectors as self-driving cars, medicine, service, and industrial robotics are being rapidly enhanced with AI [25]-[27].

The term robot is defined as a machine, automatic in function and capable of being instructed by a computer that can take over the actions of a human and be programmable. Software development, electronics, and mechatronics fuse in order to design and control the robot. Machine cannot be left unattended to complete a whole task hence there are applications whereby robots perform certain parts of the operation effectively. It is the application of software and processes to increase production using machines, and processes or machine may be used to increase efficiency in various processes and practices [28]-[30].

Employment of robotic technologies is among the most effective methods of process automation in industrial systems including logistics and transportation. Present day autonomous robots can complete order picking, order retrieval from the shelves, and order assembly for shipping tasks independently. They provide great value for logistics optimization by mechanization of operations such as the assembly processes of the complex integrating loading and unloading of products from the storage spaces [31]. The industrial entrepreneurs have witnessed the recent trends of utilizing robotics and its accompanying technology. Firstly, it raises the boiler and productivity, since robots work round the clock without tiresome leading to the suitable use of time. Secondly, robots increase the degree of accuracy and reliability reducing the human elements which could lead to disadvantageous quality. Thirdly, even though automation is costly at first, it helps in reducing the costs in future by lowering the number of workers required and waste, while also improving operational efficiency. Fourthly, the working robots reduce improved workplace hazards by taking care of dangerous operations, thus lowering the susceptibility to injuries. Fifthly, there is programmable and adjustable character of robotic systems which enables engineers to use them for other operations thus more efficient. Lastly, robotics automation is effective in encouraging quality without fewer defects in production due to accurate and consistent production methods like repetition [32]-[35].

Robotics can be considered the link between action and perception, and thus, it is self-evident that AI is necessarily connected with the ability to control a robotic system in an intelligent manner. AI provides answers to key issues such as what kind of knowledge is necessary for the completion of given cognitive tasks, the representation of the acquired knowledge, and its utilization in the required manner [37], [36]. At the same time, the area of robotics is a strong challenger to AI because the incorporation of the real-time physical world comes with methods and representations which are more than cognitive activities usually done within a desktop setting. The purpose of the document is not to describe robotics as such but rather the tasks of AI in the embodiment of this technology when extending traditional approaches of AI into a physical platform that includes positioning systems, control elements, sensors, and computing resources [38]. Robotics and AI, while separate fields of study, have been coupled both from their outset in the 1950s as well as historically. These fields were coalesced for many years and did not in fact differentiate much, due to the common use of the term "intelligent machines" which applied equally to robots as it did to AI

[39]. Robotics, in particular, and industrial systems more generally have been a fruitful interaction. The characterization of the industrial robotics research is to intelligently controlling of robotic manipulators, with special attention to using them in manufacturing environments. The traditional methodologies on in industrial robotics are coming from automatic control theory, which takes a feedback based approach to handle the interactions of robots with their environment [40]–[42].

Controlling robotic arms and other similar devices remains a challenging task even with great progress. The capacity of traditional control systems to manage uncertain circumstances is limited since they frequently rely on exact mathematical models and inflexible rule-based reasoning. Robotic systems find it challenging to adjust to real-world situations when circumstances change often due to this rigidity. Robots used for pick-and-place activities, for instance, can have trouble adjusting to changes in item size, shape, or orientation; user intervention or significant retraining may be necessary to handle these variations [43]. Also, robotic components must deal with a large number of degrees of freedom (DOF), such that one task can accomplished in many ways. As a result, trajectory planning and control will be highly complex especially if several joints are implemented in robot arm. To achieve robust control over uncertainty-stressed conditions of mechanical wear, sensor noise, and unexpected environmental situations for example, we need more than dealing with standard approaches. Such drawbacks emphases the necessities of more versatile and scalable control systems, which are able to cope with continual ambiguities, learn through past mistakes for gradual performance improvement [44]–[48].

Robotic arms are used in many industries like healthcare and Industrial Automation. For exact-actions, such as drawing, installing things together or manipulating objects — these arms have to be controlled meticulously [49]. Over the past few years AI has been widely applied for better understanding to the control and autonomy of robotic systems. While traditional control methods like Proportional-Integral-Derivative (PID) controllers [50], [51], inverse kinematics, and model-based approaches can work well in static conditions losing performance when dynamics of environment change or physical systems are become multi-degrees-of-freedom such as robotic arm control. Since they require a detailed representation (model) of the robot and its environment they are called as model-based control systems. These methods are generally used in environments and the models of the systems that the solution needs to adhere to are well defined. Thus, they do not possess the capability to manage contingencies and variations that are typical for real-life situations. Inability of traditional control methods to address dynamism and uncertainty of environments is a crucial hole in robot control systems [52]-[54].

In spite of this, model-based approaches cannot work well where the environment is very volatile or there is insufficient time to develop such a precise model [55][56]. This is where the model-free control comes in, where the controller does not require the plant model to operate. This is the type of control where no specific model of the controlled system and or its environment may be initially presupposed. Unlike model-based control techniques which involve the use of a perfect mathematical model, model-free techniques modify behavioral parameters using real time interaction with the environment, normally through a trial and error basis. These systems can change and improve because of feedback making them more versatile to address complicated and unpredictable matters. This is where the modern AI techniques, more specifically, Deep Reinforcement Learning (DRL), come to play [57]-[60].

An prominent example of Model-Free control is discovered in the Reinforcement Learning techniques which incorporates deep learning such as the Deep Q Learning in which the robot agent is awarded a Neural Network to make a guess of the best actions in a given environment without having to have a model of the whole working [61][62]. Based on these challenges, Deep Q-Networks (DQNs), which belong to the class of DRL, seem to provide a good solution. Conventional approaches lack the ability to learn and refine its policy for the manipulation of robotics systems in a dynamic manner through experimentation. As with using the concept of exploration in neural networks, DQNs can pass raw sensor inputs directly into control actions, allowing robots to make decisions of their functionality in real-time with a profound understanding of their working environments. This adaptive approach not only improves the robot's basic and complex task performance capacity but also improves scalability in challenging situations where set models fail [63].

A robotic arm controlled by DQN is the main shown in Fig. 1, which illustrates the convergence of sophisticated AI methods with autonomous robotic systems. The robotic arm represents the integration of DQN, an RL algorithm that uses interactions with the environment to learn optimum strategies and make goal-oriented motions.

Fig. 1. Robotic arm with DQN integration for goal-oriented tasks

This approach is in opposition to traditional control strategies that use predefined models and can therefore perform poorly in adapting to change. It also presents the act of performance in controlled environment to illustrate the future application on industrial automation and service robotics where the precision accuracy and adaptive control is highly desirable. But this Figure also indicates RL's

potential for increasing the intelligence and decision making capability of autonomous systems to form a part of advancing robotics in the small real world environments [64][65].

Thus, it is a goal of this research to show that by supplementing the conventional methods of control with DQNs, the overall efficiency and effectiveness of goal-oriented tasks would enhance especially in the controller for a robotic arm. As this work further elucidates using a profound analysis of how the application of AI optimizes the intelligence of robotic systems, this work suggests another widening contribution to learning-based approaches and practical robotic applications [66].

To this end, the research proposes DQNs as an innovative solution to the control of the robotic arm. DQNs use reinforcement learning allowing robots to learn control policies from the high-dimensional state spaces like raw sensory data. DQNs have memory advantages over other methods because they do not require specific models in order to transform or learn new needs, which are ideal for complicated goal-oriented tasks [67][68].

The research contribution is in two folds. First, it aims to show that DQNs can be used to control the 2D robotic arm on specific goal-oriented tasks that involve the optimization of the position where the arm needs to be. This goes a long way in addressing one of the major areas that the traditional control methods have not captured especially in ability to address flexibility and real world performance based on AI solutions for asserting intelligent robots, thus opening up a new frontier for better and advanced adoption of robots in real life applications.

Second, the research makes a contribution through designing and deploying an automated robotic system with AI and robots the DQN algorithms for moving a two-arm robot towards a fixed point or what they call the ‘red point’. This is achieved by integrating robotics automation with an intelligent system known as the DQN to increase the efficiency and precision of a 2D robotic arm. The system demonstrates an application of how AI can improve the precision and speed of robotic arms for real-world applications, arguing for smarter, more learning-based robots. Such contributions will help open the next frontiers in the development of robotic control systems in areas where flexibility and precision are critical.

This paper's structure is set up as follows: Section 2 focuses on related previous research works on the topic. Section 3 is as follows the approach that is offered and recommended. Section 4 presents the experimental results in detail and the suggestion for the future studies and the conclusion demonstrated in Section 5.

II. LITERATURE REVIEW

The Deep DQN can operate in large dimensional state spaces, such those seen in video games, since a deeper convolutional neural network approximates the Q value function. This approach handled a problem that traditional Q-learning was unable to address, namely the issue of enormous state spaces. DQN discovered that two of these key insights are target networks and experience replay. Previous experiences need to be stored and recalled in a random manner to reduce correlation between consecutive events, which would otherwise lead to oscillations during training. Changing Q-values can be devastating during training.

Wu et al. [69] achieved precise position control of a robotic soft arm by combining the RL control technique with the data-driven modeling approach. A deep Q learning-based control method was used to achieve this. A control strategy learning approach is proposed to tackle the problems of unstable effects and sluggish convergence in the simulation and migration process of applying deep RL to real robot control tasks. This approach, which is based on experimental data, entails creating a simulation environment in which the control technique is trained before being implemented in the actual world. Test results have unequivocally shown that the technique is able to successfully manage the soft robot arm, and that it is more resilient than the traditional method.

For its application, Hwangbo et al. [70] has expanded the usage of DQNs for robotic arm control to emulate actual physics. This was done by teaching a 2D robotic arm to put its tip at a desired location subject to constraints arising from environment such as friction and interferences. This study showed that DQNs if trained alongside with domain randomization methodologies created robots that could perform well in different terrains. To achieve this kind of control, the locomotion mechanics of robots are built with the flexibility to adapt to the environment real time, maintaining the efficiency of the robotic arm accurate in the real world for use in amputees. Such developments see the importance and reliability of DQNs in task-oriented robotic functions.

Liang et al. [71] proposed a way to learn without outside supervision and presented a challenging object-handling task. The assignment's goal is to obtain an object by exploiting environmental fixtures like walls, furniture, or heavy objects instead than relying just on a single parallel gripper. Other than a cursory examination of a target object, no prior information is needed for this Slide-to-Wall gripping challenge. As such, the robot needs to learn an effective strategy through scene observation, which includes the target object, surrounding objects, and any other disrupting objects. They suggest using a target-oriented deep Q-network (TO-DQN) to learn ergodic visual affordance maps that provide action guidance for a robot. The problem is framed as visual affordance acquisition. The TO-DQN algorithm is trained offline on a simulated robot manipulator and then deployed online on the real end-effector, as active training requires that the robot should explore while colliding with the fixtures. Empirical evidence is presented to show that TO-DQN can solve the problem successfully in both simulation and real-world scenarios under different environmental conditions. Additionally, in terms of training resilience and efficiency, TO-DQN performs better than both a modified version of DQN and a standard DQN. The evaluation findings from both simulated and real-world experiments show that the performance attained by the policy trained by TODQN is similar to that of human beings.

To this end, Zhu et al. [72] have introduced DRL methods, as well as DRL-based navigation frameworks in this paper. Navigation is a more basic issue of these robots, and thus DRL has become a hot topic in the field due to its excellent representation and experience learning capabilities. Currently, the use of DRL has been increasingly seen in the control of mobile robot navigation systems. Then systematically compare and analyze the relationship and differences between four typical application scenarios: Local OAA, Indoor-IN, multi-MRN, and Social N. Then, bring out the general analysis of DRL-based navigation. Last, discuss the challenges and some possible solutions regarding DRL-based navigation.

Indeed, this paper explores the automatic exploration idea under the unknown environment as pointed out by Li et al [73] which raise the key point of applying the robotic system to some social tasks. By stacking decision rules it is impossible to cover various environments and sensor properties to solve this problem. These situations require learning based control methods because of this. However, these methods are marred by low learning efficiency and poor transfer of learnt skills from simulation to the real world. An exploration framework for this paper is general and proposed by analyzing decision making, planning, and mapping parts of the exploration process to make the structure of the robotic system more modular. On the bases of this framework a decision algorithm that applies deep reinforcement learning is put forward which employs a deep neural network for learning the exploration strategy from the partial map. The above-mentioned results demonstrate that this proposed algorithm has higher learning rate and unknown environment adaptability. Furthermore, experiments were carried out on the physical robot and the results indicated that the learnt policy is transferable on to the real robot.

W. Zhao etl. [74] This kind of deep reinforcement learning has recently proven highly effective in several domains of robotics at large. Due to impracticality of acquiring actual data, i.e., high variance and costly, simulation environments are used to train the various agents. This is not only helpful in giving a potentially endless data base, but also eliminates risk issues regarding true robots. However, the simulation to real-world transition detriments the performance of the policies when the models are implemented in real robots. This research hence points to multiple current efforts devoted to reducing the size of the sim-to-real gap and achieve better policy transfer. Multiple methods have been proposed in the context of recent years with specific applicability across various domains, but, to the authors' knowledge, no review proposed in the contemporary literature provides a holistic analysis that places all the proposed methods into context. In this survey paper, we cover the fundamental background behind sim-to-real transfer in deep reinforcement learning and overview the main methods being utilized at the moment: It is categorized by the methods such as domain randomization, domain adaptation, imitation learning, meta-learning and knowledge distillation. Here we group some of the most related latest publications and describe the principal domains of artificial intelligence usage. Finally, it was described the main advantages and disadvantages observed in the various approaches and highlight the most significant prospects.

Gupta et al. [75] investigated how robotic technology and AI are altering plant phenol typing, in order to address the impact of climate change on global food security. The article investigates recent advances and future possibilities, with a special focus on how robotics could help achieve high-quality data in plant science. The survey assessed a variety of robotic platforms and systems, including aerial drones, ground-based robots, wheelchairs and self-driving cars with a plethora of non-invasive sensors for phenotypic evaluations. It then investigated how massive data were being processed by the AI-driven algorithms, so as to deliver key understandings about plant traits and environment-responses.

A technique for controlling the motions of an industrial robotic arm using RL was presented by Jafari-Tabrizi et al. [76]. During an automated quality inspection, they improved the process of telling the robotic arm to perform a thorough inspection of the surface of free-formed components. Right now, manual training by experienced specialists is the most common way to teach a robotic arm to follow an intricate course. As such, it takes a human professional a considerable lot of time and effort. Moreover, in the event that a new component with a modified design needs to be inspected, the human specialist has to create an inspection path for this component, which causes a major disruption to the automated inspection procedure as a whole. They also experimented with a domain transfer scenario, where using RL techniques to change the tool center point (TCP) of the robot between different components would speed up learning by exploiting knowledge already available in relation to component geometry. The robot has already been setup in a simulation where we can command the TCP (Tool Center Point) position and orientation. On the robot's panel, a randomly generated 2D trajectory is shown throughout the simulation episodes. By observing the points on this route, the robot—which was taught using the Deep Deterministic Policy Gradient algorithm—follows it. The robot's goal is to complete the trajectory in the least amount of time and with the least amount of deviance from the original plan. They gave an explanation of the initial results from the simulation environment and outlined the next steps that needed to be followed.

This proposed method represents a new approach and fills in one of the key gaps in the state-of-the-art by presenting a model-free adaptive control DQN based technique for robotic arms that even outperforms traditional and previous AI-based methods. Our method shows large benefits to adaptability, accuracy and efficiency for goal-oriented 2D robotic arm control than the previous literature. While prior work leveraged RL for accurate control, they either used soft robots or simulation domain randomization to obtain environmental generalization, but were not trained efficiently or did not transfer well from simulation to real environments. On the other hand, by optimizing interaction with environment during achieving control actions directly from the states for each cycle, this method using DQN has increased performance features of fast learning and high-

accuracy processing in real-time adaptation. In comparison to previous DQN methods (original and those modified for dynamic conditions) the method in the experimental results presented displays superiority, reflecting its robustness and practicability for real-world applications.

III. METHODOLOGY

A. Overview

This section describes the approach utilized in this study to guide a two-link robotic arm and bring it towards a target or red point as in your task. The method proposed employs an advanced AI methodology called DQN which is used to create an agent that learns and acts the optimal control strategies for reaching task of robotic arm. The method provides a specific formulation that comprises various components: problem definition, DQN architecture, state representation and action selection, reward function and training process. The task is to ensure the precise positioning of the robotic arm in a defined workspace. It has joint angles and velocities that define the state of the arm and moving these joints around, makes up its action space. The problem is how to find the optimal movement sequence so that you can reach out by the closest path with your hand. Similarly, due to extensive field experiments and deep reinforcement learning studies, it can be argued that DQNs which are learned from real-time interactions between a controller and the system tend to outperform traditional control approaches in many aspects such as flexibility, learning efficiency, and task performance in complex environments. This is very important to deal with uncertainties encountered in practice [77][78].

For this type of robotic control task, DQNs have several advantages. Firstly, DQNs enable our robot arm to learn in a model-free way: instead of learning a predefined model we train it directly in experiential data and thus adapt much better to changing scenarios. This flexibility is needed to accommodate real life uncertainties and variations. Second, DQNs are well suited for learning optimal function-value functions from scratch by trial-and-error interactions with the environment and hence they exhibit better learning over time. Furthermore, the agent can handle high-dimensional sensory inputs, such as joint angles and velocities due to the architecture of DQNs which helps it make more intelligent decisions [79].

B. Deep Q Network Algorithm

In recent years, significant improvements have been made in AI, and new approaches have appeared for addressing the issues of robotics control with the help of DRL. One of the most effective forms stands for DQNs what is the combination of the traditional Q-Learning with Deep Neural Networks that allow robots learning different tasks through interacting with their environment. Consequently, DQNs should be able to learn and respond to system change pertaining to flexibility and adaptability in robotic control applications [80]–[83].

DQN stands for Deep Q Learning which is a type of reinforcement learning that incorporate both Q-learning, which is a model free type of reinforcement learning and deep learning. To estimate the Q-value function, commonly referred to as the predicted future reward of acting in the current state, deep neural networks are applied. The DQN allows an agent to learn the best policy through trial in the environment making it ideal for complex control tasks like the operation of robotic arms [84][85].

The Q-value update rule in a DQN is derived from the Bellman equation is given as follows (1):

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

where, $Q(s,a)$ is the current Q-value for state s and action a. α is the learning rate, determining how much new information overrides old information. r is the reward received after taking action a in state s. γ is the discount factor for future rewards, balancing immediate and long-term rewards. s' is the next state resulting from action a. $Q'(s',a')$ is the Q-value estimated by the target network for the next state.

The DQN algorithm is a significant reinforcement learning, especially in tasks, such as robotic arm manipulation, where precision control and decision-making are required. With its architecture and capabilities it is good to learn optimal policies in environments where efficiency counts, which makes it a very sensible approach for goal-oriented tasks.

The DQN architecture is shown in Fig. 2. A diagram of the DQN layers, from input (state representation), through convolutional layers (if applicable), to fully connected layers and output Q-values.

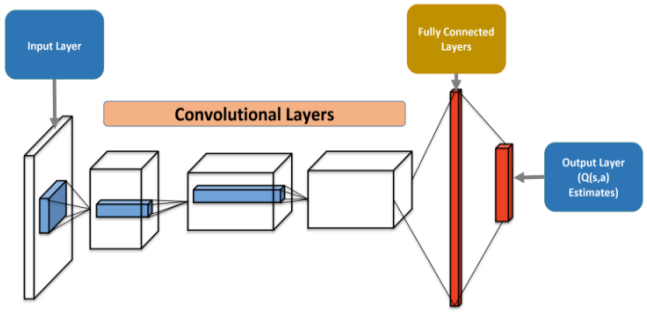

Fig. 2. DQN architecture diagram

Based on Fig. 2, the output layer shows Q-values for every action that is feasible, while the input layer reflects the state (e.g., joint angles, velocities), convolutional layers for feature extraction (if applicable), and fully connected layers to approximate Q-values [86].

The DQN architecture consists of a deep neural network (DNN) to approximate the Q-value function that gives us the expected future rewards for some state-action pair. Using convolutional layers is especially beneficial when the state inputs are high-dimensional, such as images or otherwise complex sensory data. Convolutional layers excel in feature extraction, enabling networks to identify spatial hierarchies and patterns of relevance in input data. This capability is important in robotic control tasks, where the agent must learn important features of the environment in order to make appropriate decisions. Convolutional layers and completely linked layers together help approximate the Q-values more precisely, which in turn improves

performance overall and facilitates more efficient action selection [87]–[91].

The DQN agent works directly on the image-frames from the video stream and does not require the state to be manually abstracted into features derived from the images. Such feature generation is automatically done within the agent's Convolutional Neural Network (CNN), which acts as an approximation-function to predict the probability of different possible actions [92][93].

Fig. 3 illustrates the design of a DQN for managing a 2D robotic arm, with an emphasis on the flow and connections between components.

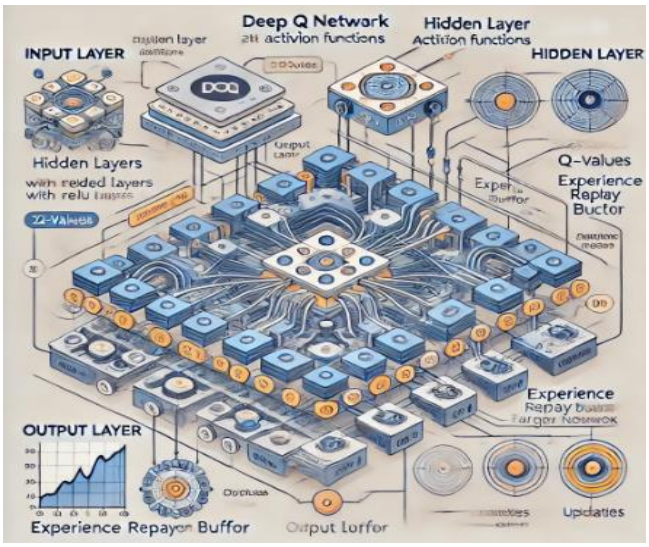

Fig. 3. The architecture of a DQN for controlling a 2D robotic arm

It shows the main components, such as the input layer, hidden layers, output layer, experience replay buffer, target network, and data flow between them [94].

The Fig. 4 shows the architecture of a Deep Q-Network (DQN) for controlling a 2D robotic arm, often used to make decisions in reinforcement learning tasks.

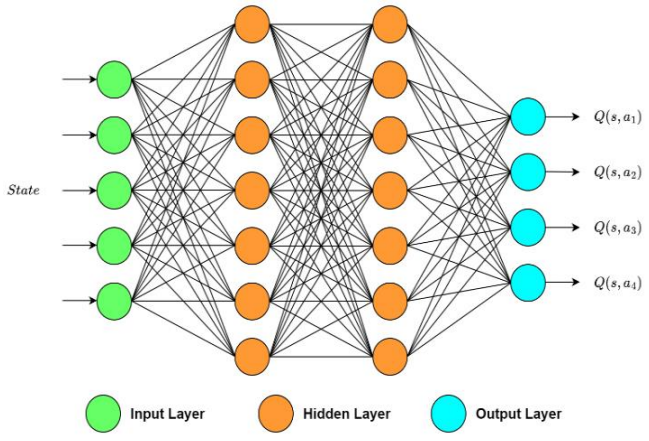

Fig. 4. The architecture of a DQN for controlling a 2D robotic arm

The input layer correctly is just a representation of the current state of whatever system you are trying to learn about (in a RL environment). Each node (also known as a neuron) relates to a different characteristic, or an attribute of the state. What these inputs would have meant without considering the robotic arm for an example, are joint angles i.e. velocities, or positions of the links in the arm.

Hidden layers are used to capture meaningful patterns and correlations in the input data. They parse the state information and convert it into a higher level of abstractions. There are multiple layers to a neural network, and with more layers and neurons in the network you have a more complex model. Hidden layers in deep learning help by learning the complex, sometimes nonlinear relationships between the state and the actions.

The output layer is Q values to all four action ($Q(s,a_1)$, $Q(s,a_2)$, $Q(s,a_3)$, $Q(s,a_4)$) These are the expected future rewards for taking each corresponding action a_1, a_2, a_3, a_4 given the state s in that state. The DQN approximates these Q-values and this helps in selecting optimal action.

The Fig. 4 shows the fully connected neural network used in Deep Q-Learning for multiple actions are predicted based on the current state. The Q-value of each action represents the neural network's estimate of the amount of reward you are going to get if take that action. DQN updates its parameters (weights and biases) to improve these estimations with time by minimizing the difference between expected Q-values and target Q-values extracted from real rewards gathered during interactions with an environment.

The DQN algorithm is shown by the flowchart in Fig. 5. A DQN-based strategy requires many crucial processes, which are outlined in the diagram. These include initializing networks, choosing actions based on a ε-greedy policy, updating the Q-network, and regularly updating the target network.

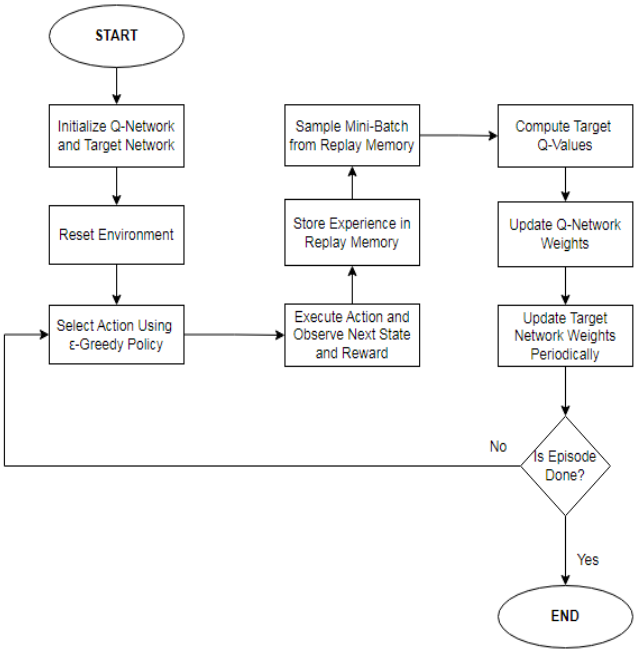

Fig. 5. DQN algorithm

In order to show how the DQN algorithm learns and adapts to direct the 2D robotic arm toward goal-oriented activities, this graphic may effectively depict the process flow [95].

Reinforcement Learning (RL) is a process of learning optimal behaviors through interactions with the environment. In it, the agent learns to make choices that will lead to the largest long-term payoff. Just as an athlete improves by doing his or her sport over and over again until the feedback makes it more robust, an RL agent interacts with its environment in a loop fashion, changing its plans based on what worked last time. The agent will have to assess potential alternatives on its own, unless directed otherwise, in order to determine which one works best. This deductive process is similar to the unique way a puzzle solver would use different pieces in order to fit them together correctly [96]. The future potential rewards in RL are as important as the immediate gains, and they direct the learning process of the agent. The unique power of RL is that it should be able to find the best path to success inside dynamic complex environments without supervision. RL is a particularly potent algorithm because it can autonomously determine the actions required for success in an unfamiliar environment [97][98].

To complete an objective in the RL problem, an agent needs to investigate a foreign environment. The fundamental principle of reinforcement learning is that all goals may be characterized by maximizing expected cumulative reward. The agent needs to learn how to sense and modify the state of the environment in order to reap the highest possible reward. A paradigm for the formal framework utilized in RL is the problem of Markov Decision Process (MDP) optimum control [99].

Fig. 6 shows the basic diagram of the RL process. It illustrates the interaction loop between the key components: agent, environment, policy, and reward signal.

Fig. 6. RL components

- Agent: The decision-maker in the RL system by choosing actions according to a policy.
- Environment: The system that is outside the agent that the agent communicates with. It answers to the action of the agent and offers new perceptions and rewards.
- Observation: Subsequently, the agent understands the current state of the environment through observation.
- Action: From the current observation, the agent incorporates its policy to take certain action.
- Reward: Subsequently, the agent gets a reward, which is a feedback which tells the agent how good or bad the action taken for achieving the goal was.
- Policy: This is the actual reason for the agent to pop out a decision on what to do depending on the observations made.

The continuous feedback loop is depicted in the diagram, in which the agent operates in accordance with the policy it adheres to, observes the environment, and takes actions and receives rewards. Through this cycle, the agent can gradually learn from its actions and enhance its performance to optimize the total reward.

The diagram shows the continuous feedback loop where by the agent behaves according to a policy, observes the environment, takes actions and receives rewards. The agent can begin to learn through its actions to improve optimization of the total reward over time through this cycle.

The value function accurately captures the "goodness" of a state and is a highly helpful tool for modeling the standard representation of the reward signal. On the other hand, the reward signal just shows the amount of reward that is likely to be obtained when an entity is in that condition, but the value function shows the entire amount of predicted benefit once an entity is in that state and beyond. The goal of an RL algorithm is to choose the best plan of action that maximizes the average value that may be obtained from each system state [100].

The DQN is one efficient method in the field of RL. It integrates the ideas of deep neural networks with Q-learning to allow agents to learn optimal rules in complicated scenarios. Fig. 7 shows the architecture of DQN.

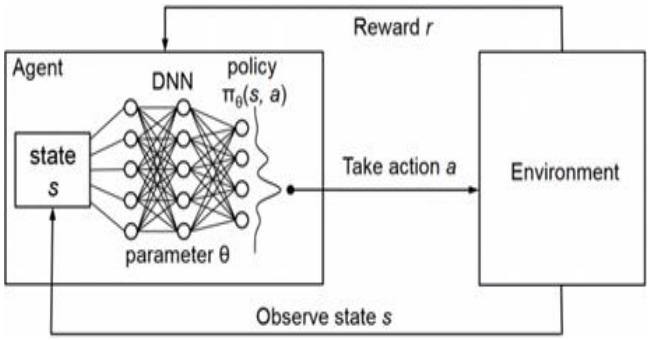

Fig. 7. Simple schematic of DQN architecture

DQNs relate environmental conditions to the expected return, or the total of potential rewards, for each action that may be taken. They do this by approximating the action-value function using a neural network. Finding the best course of action to maximize the expected return for every state is the aim of the DQN [101].

The action observation and rewards help the agent to interact with the environment and train the DQN. Such experiences are then stored at a memory buffer of the agent and are then used to update DQN at frequent intervals. This is the way that experience replay is used to update the DQN. It entails randomly picking a batch of events randomly from the memory buffer. This procedure can put the learning

process on a stable footing and would allow the agent to learn from a wider variety of events.

Especially ideal for the DQN, the algorithm used differs from the original Q-learning for the purposes of off-policy, adapting the action-value function in regard to discrepancies between the target and expected values. The target value is calculated using the Bellman equation which states the expected value of taking a particular action is the reward obtained for that action plus the greatest expected value for the next state. One of the key components of deep Q-learning networks and other neural network techniques that use reinforcement learning is experience replay. It involves using a memory buffer to store a set of experiences (state-action-reward-next state) and then using those experiences to train the DQN. The main concept of experience replay is that allows the agent to update the data from a set of encounters rather than focusing in new ones. It is assumed that this can enhance the maintenance and control of the learning process, and can improve the overall performance of the DQN.

The following is a summary of the working process:

- State Representation: Give sufficient discretized quantitative description of the current state of the environment, for instance, raw pixel values or the original feature values.
- Architecture of Neural Networks: An extended neural network that maps states to action-values of all possible actions must be used. A very famous type of deep neural networks is the CNN.
- Experience Replay: What needs to be saved in the replay memory buffer are state and action, reward and next state and some other attributes going by the name of experiences.
- Q-Learning Update: Individual batches of experiences are then sampled from the given replay buffer on the play memory and the weights are tuned on the given neural network. This is done to update by minimizing the difference between the objective and predicted action values using the Bellman equation's loss function.
- Exploration and Exploitation: Ideally, determine pure exploitation in a random manner as a means to explore potential opportunities, or pure exploitation in a greedy manner to follow the existing plan.
- Target Network: Beneath the main operation of the network with the specified design, the target network is employed for stabilization of the learning process, with a different configuration. The weights of the primary network will be copied periodically to the target network to replicate their weights.
- In Step 7, Steps 1 through 6 are reiterated. Engage with an environment, gather evidence from the field of operation, modify the network and continue the process of choosing another policy until the ideal policy is reached.

C. Robot Arm Environment

For this paper, we employed an advanced AI approach, the DQNs to create an agent that solves the 2D robot arm Reacher problem well. Our starting point for experiments are the gym_robot_arm environment which is based on the Gym library, an open source Python framework for developing and comparing reinforcement learning algorithms. This library is an API that affords requestors consistent interfaces they can use for interaction with different environments when employing the learning algorithms. The setup of the robot arm environment is quite important; the state space, action space, and reward configuration defines what the DQN agent will be expected to play with. For example, the range, the length of the arm segments, the target positions, and the physics of movement can influence learning to a great extent. A well-defined configuration flexible configuration of the agent permits it to search the action space while the too strict parameters may give the ideas of the best strategies. As such, not only does the environment design impact the efficiency of training but also introduces variance in the DQN agent for the target goal due to the dynamics and complexities within environment.

This environment contains two main components:

- Robot Arm: Environment involves a robotic arm with two links of which each is 100 pixels in length. The main manipulator in the environment is the robot arm which is responsible for getting to the intended location.
- Red Point: At random throughout each episode, the target point referred to as the red point is created. This red point is also highlighted and the robotic arm should move in order to align with this position.

The training process is specifically designed to enable the DQN agent to acquire adequate control techniques of the multifaceted robotic arm in a given episode. In the Python code, each training scenario is defined to run 20 episodes which are enough to explore and learn from. The primary objective is to assess the robot arm's feasibility for learning and implementing the control strategies necessitated for the precise attainment of the red point across multiple episodes. Some of these hyperparameters comprise include a learning rate of 0.001 and a discount factor (γ) of 0.99.

This value was chosen through preliminary experiments that compared the rate of convergence and stability level at which it would stabilize. The learning rate set to be 0.001 is good for the agent to update the Q-values while learning good policies for actions without causing big fluctuations around the network.

The discount factor is set to 0.99 to prioritize immediate rewards while also considering future rewards. This is particularly important in tasks requiring long-term planning, such as reaching the randomly generated red point. By using a high discount factor, the agent is encouraged to explore actions that may have delayed rewards, thereby enhancing its ability to attain the target effectively.

The discount factor is set to 0.99 to prioritize immediate rewards while also considering future rewards. This is

especially significant for functions that involve certain time limits, especially when reaching at the randomly appearing red point. With the help of high discount factor, the agent is encouraged to explore actions that may have delayed rewards, thereby enhancing its ability to attain the target effectively.

Fig. 8 shows environment components that the aim to reaches red point generated randomly every episode.

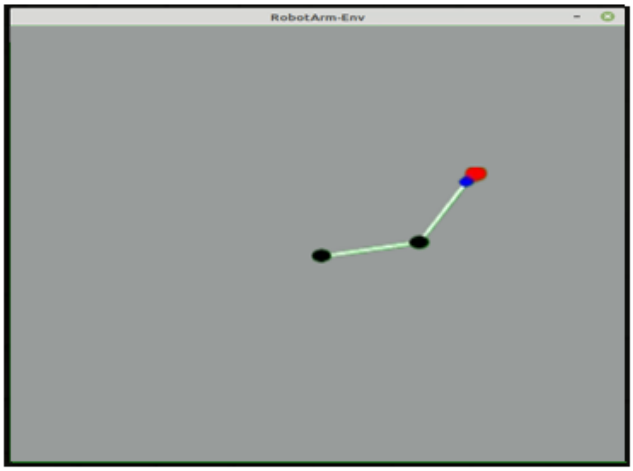

Fig. 8. Environment components

Altogether, all these components help in overcoming the issues related to high dimensionality and dynamism of the scenario of controlling the robotic arm. By incorporating strategies, such as the target network and experience replay, DQN can make training stable, thus training the agent to learn the optimal policies in the process of reaching the randomly appearing red point targets in each episode.

The following points explain the instructions for the implement the Python code to accomplish the reaching goal to the red point.

1) Create the environment
2) Run episodes with certain number. In this experiment, we determine the number of epochs is 20.
3) Reset the environment for each episode
4) Run for a maximum of 100 time-steps
5) Render the environment
6) Print the observation
7) Sample an action from the action space
8) Take a step in the environment
9) Check if the episode is done
10) Print the number of time-steps taken
11) Close the environment.

There are two variants of the robot arm environment, namely robot-arm-v0 and robot-arm-v1, In designing this architecture, two main goals were achieved as stated below The first one is to create the architecture of the robot arm apparatus that will challenge the understanding of the DQN agent while the second one is to develop an environment for DQN that is different from the basic five environments listed above. The comparison of observation spaces, action spaces, reward functions and terminal condition in Gym 2D Robot Arm Environment of both versions are explained in the Table I.

TABLE I. THE COMPARISON OF OBSERVATION SPACES, ACTION SPACES, REWARD FUNCTIONS AND TERMINAL CONDITION IN GYM 2D ROBOT ARM ENVIRONMENT OF BOTH VERSIONS

Criteria	Robot-arm-v0	Robot-arm-v1
Observation Spaces (Continuous)	- Target position in x direction (in pixels) - Target position in y direction (in pixels) - Current joint 1 position (in radians) - Current joint 2 position (in radians)	- Target position in x direction (in pixels) - Target position in y direction (in pixels) - Current joint 1 position (in radians) - Current joint 2 position (in radians)
Action Spaces (Discrete)	0: Hold current joints angle value 1: Increment joint 1 2: Decrement joint 1 3: Increment joint 2 4: Decrement joint 2 5: Increment joint 1 and joint 2 6: Decrement joint 1 and joint 2 By default, increment or decrement rate for both of joints are 0.01 radians	0: Joints 1 value (in range -1 to 1) 1: Joints 2 value (in range -1 to 1) Value will be scaled into minimum and maximum of joint angle
Reward Function	Robot will get penalty -1 if current distance between tip and target position is greater equal than previous distance Robert will get reward 1 if current distance between tip and target position is > -epsilon and < epsilon, where epsilon = 10 pixels	reward = -distance_error/100
Terminal Condition	Current reward is -10 or +10	If target position is > -epsilon and < epsilon, where epsilon = 5 pixels

The reward function is a part of RL that defines how the learning process happens and how the robotic arm behaves in environments, originally robot-arm-v0 and the modified version robot-arm-v1. It provides the feedback necessary for the DQN agent to evaluate the effectiveness of its actions and adjust its strategy accordingly.

The structure of the reward function in the case of the Robot-arm-v0 environment is based on relative distance measurements. If the current distance between the tip of the arm and the provided target position is greater than or equal to previous distance, the robot receives a penalty which is -1. On the other hand, the agent gains +1 if the distance is within the constant epsilon (10 pixels). This assignment reward strategy affords trial-and-error process; hence the agent will employ action to work towards minimizing the distance to the target. However, this kind of reward function does not allow fine grained control as the penalties and rewards are binary, which causes oscillations in the control by the agent and the agent continuously oscillates between exploration and exploitation.

For the reward function of Robot-arm-v1, the reward function of this version becomes more of a continuous structure, offering a reward corresponding to the negative distance error normalized by the factor of 100. This means that the agent gains a reward depending its distance to the goal, with higher values of negative distance error implying higher penalty. This makes the process recurrent and gives a better understanding of the action execution and the result in relation to the goal. For this reason, the agent is encouraged to reduce distances to the target more accurately, thereby resulting in smoother control strategies. This refinement in the reward function hugely helps in promoting the learning process of the agent and even results in improving the policies for the environment.

The existence of such versions is explained by the fact that further improvements to the learning process and agents are needed. For example, the move from a discrete action space as seen in robot-arm-v0 where actions are defined as increments or decrements to joint angles to a continuous action space in robot-arm-v1 enables better control stance of the robotic arm and for more nuanced control over the robot's movements. This change can also have a drastic effect on the ability of the agent to learn the better policies because actions that are continuous can allow better adjustments that are best for the convergence during training.

Although we pointed out the usage of the ε-greedy policy as one of the basic approaches that help combine exploration and exploitation, it is necessary to discuss its application in more detail. During training, we initialized ε to a higher value (for instance 1.0) to encourage the agent to explore the environment early in learning. This let the agent learn many actions and states. As training progressed, we implemented a decaying epsilon strategy, where ε was gradually reduced (e.g., from 1.0 to 0.1) over a predetermined number of episodes. This approach ensured that the agent transitioned from exploratory behavior to a more exploitative approach as it became more confident in its learned policies.

Taken together, these modifications indicate a principled approach to environment design and are in place to tackle difficulties agents may experience during robotic control tasks. There is, however, a need to carefully choose the environment version as this determines the learning characteristics of the agent to the extent of efficiency in the tasks at hand.

IV. EXPERIMENTAL RESULTS

In the following section, we present the results achieved for the reaching task using the gym_robot_arm environment in the version 0. The progression of the robotic arm's movement from the initial position to the final target position is illustrated in Fig. 9 and its corresponding subfigures (a) to (i). The figure captures key stages of the arm's approach to the target, represented by the red point, and demonstrates the incremental steps of the DQN-based agent in solving the reaching task.

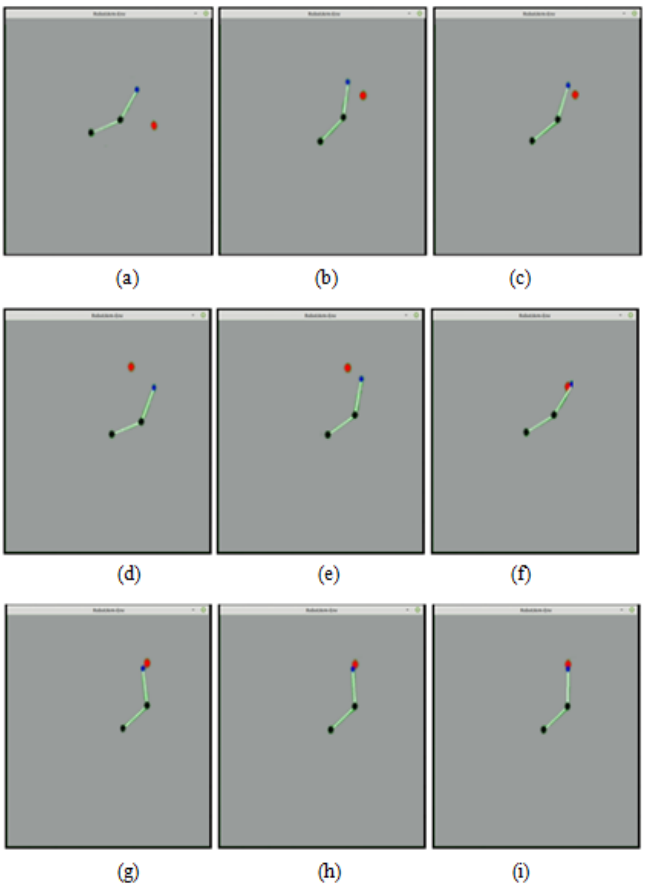

Fig. 9. Experimental results steps

The procedure starts in subfigure (a) where the environment sets the initial pose of the two-link robotic arm to a random position and the target red point is located anywhere in the workspace. In the course of the episode, subfigure (b) shows the starting position of the robotic arm towards the red point: two links rotate and stretch to minimize the distance. The representation in subfigure (c) of the 'robotic arm' and the 'red point' closer together suggests that early-stage navigation was successful.

In the next step, the red point shifts slightly to the left, this requires the movement of the robotic arm, shown with the help of subfigure (d). As shown in subfigure (e) the arm responds appropriately with both links adjusting to maintain their approach. In subfigure (f), the distance between the end effector of the arm and the red target is at the shortest possible to complete the presented task effectively.

Finally, in subfigures (g)-(i), the movements of the upper end of the robotic arm are shown with the end effector placed on the target red point. Such a progression demonstrates that the DQN agent is capable of learning how to modify the control of the arm in performing the reaching task. Every subfigure is important for visual representation of how the arm is getting better and getting control over its position during trials and errors to reach the target point.

In this experiment, the ability of the DQN to solve goal defined problems in continuous environments has been well shown because of the successful movement of arm and its ability to make correct alignment changes of the red target point.

Fig. 10 shows direct observation of the performance of the DQN in the control of the robotic arm environment. The figure shown is a frame of the robot arm planning and executing the operations in that environment and a red point is the target that needs to be achieved by the robot arm. In the two-link arm design depicted here by clear connections between the joints and links, it is seen adjusting to fit the target.

The addition of a Graphics Interchange Format (GIF) image to extends this understanding as the viewers are able to observe the arm continuously interacting with its environment. Such dynamic presentation makes the learning easier to understand and maneuver which in return helps explain the efficiency and behavior of the DQN model in real life. Therefore, the Figure helps to overcome the gap between the performance of the theoretical model and its application, which demonstrates the output of the DQN interactively.

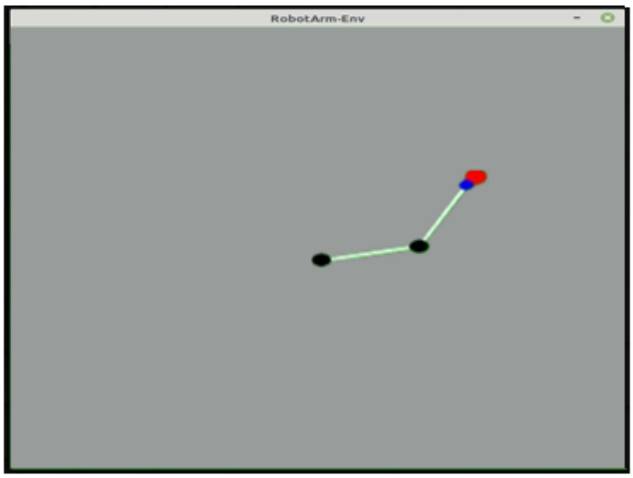

Fig. 10. DQN Output – GIF format

Using a two-armed robot, the agent traverses a two-dimensional space and these subsequent pictures represent his actions as well as his thoughts. It is clear from the way the model reaches a certain point that it is capable of moving the locations and angles of the robot's arms. Each shot measures how the agent engages with its environment, providing an account of the trajectory and methods to achieve the goal. When the output from the model is represented in a GIF format, one can easily observe the progression of the model because the time factor is incorporated, this helps the audience appreciate the abilities of the model especially the performance of the model in challenging situations. This above visual proves to be beneficial to any experimental work, analytical study and in the proof of claims about the results of the DQN model as it provides a clear example of the working of the model in real circumstances.

During training, DQN agent's performance is appraised by using its reward function measured over a total of 2000 episodes. In the Fig. 11 the overall and average reward variations across training episodes are presented.

Looking at this progression, in the initial phase (episodes 0 to 500), the average reward positively increases, which shows that the agent's performance in the reaching task improves steadily over time. This gradual increase in average reward indicates that the DQN has been successfully trained to shorten the distance from the end effector of the robot arm to the red circular target by refining its motions.

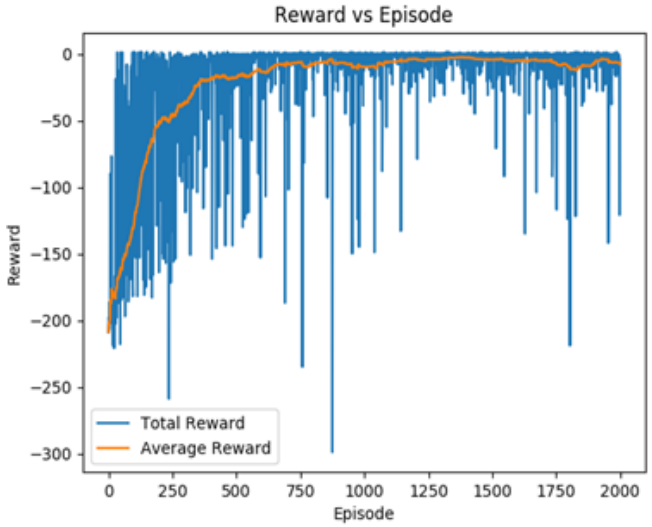

Fig. 11. DQN Performances in training phase

Beginning from the 500 episode, the so-called phase of reward stabilization is observed and the increase of the reward also becomes stable, with total and average reward parameters hovering on a certain level. This phase of stabilization means that the agent has performed sufficiently well in learning the dynamics of the environment it is placed into, and only minor or no changes in performance can be observed with further learning. The observed oscillations in the value of reward indicate that there are times the agent does explore, which is a normal behavior in training when the agent is checking available actions for the best policy.

It may further be noted that the randomness associated with the environment and the need to balance exploration and exploitation are likely the reasons for the drop in total rewards over certain periods. The consistent reward figures in the following runs indicate that even with these variations, the DQN is able to consistently optimize to a desired policy in the later episodes. To provide a comprehensive analysis, a comparison between the DQN and other reinforcement learning methods or control approaches would be ideal. For instance, evaluating algorithms such as Proximal Policy Optimization (PPO) or Actor-Critic models on the same task could offer insights into the relative performance and efficiency of the DQN. Including a baseline comparison (e.g. traditional PID control or random action selection) would provide context for the effectiveness of DQN in this robotic control scenario.

A more detailed quantitative analysis also would be useful in this case; for instance, the average episode duration, the average rate of success, the average convergence time, etc., would enrich the picture of model's performance. For example, the parameter that shows how far on average from the target point the model is along the time axis would also give additional proof of the model's

ability to minimize the error. Such metrics would enable the readers to evaluate how well the DQN performed the reaching task and its precision.

Statistical measures, such as the variance of rewards or confidence intervals, would offer insight into the consistency of the agent's performance. In high variance in reward values from DQN, there is a tendency of instability that can be attend to by changing learning rate or discount factor or any other hyperparameter.

This work's primary contribution is the use of DQN to operate an autonomous robotic arm, which allows for the potential for transfer skills and works well for a variety of job archetypes. Learning curves and qualitative data both shed light on how well the strategy works for teaching sophisticated control strategies without requiring the arm's behavior to be programmed. Additionally, the technique may be repeated thanks to the provided comprehensive experimental design.

In summary, the comparison of DQN performance in the reaching task, further assessments of other models, provision of intricate performance indices, and elucidation of the merits and demerits of the model would provide a more comprehensive evaluation of its capabilities.

V. CONCLUSION AND FUTURE WORK

One of the key achievements of this work is proving the effectiveness of the DQN algorithm in solving the 2D robotic arm task, which required the arm to autonomously discover a policy for reaching the target point. The model was not only able to converge fast, but also had improvement over time, which was evident in the trends of the rewards. This illustrates that, while other researchers have engaged the DQN in similar robotic tasks, herein, its application is unique to that task. Thus, we showed that the DQN algorithm can be employed to solve the reaching task for a robot in a bounded space, through the completion of a pre-specified number of episodes and using a reward function to assess performance. Our work corroborates the use of DQN algorithm as an effective way of reaching in 2D robotic arm task, indicating its use in the existing research.

The effects of these findings go beyond the task at hand, indicating that DQN can be used for other more practical tasks in the field of robotic, automation and artificial AI. This capability of DQN in driving a two-link robot arm to a target in a constrained setting can be extended to more complex tasks such as assembling components in a factory, carrying out surgery in medical robots, or building a drone capable of picking up an object. Additionally, the fact that DQN can be used in an active learning scenario instead of implementing a static controller makes it ideal for applications where there is no possibility or sense of using a priori controller models.

This research has large-scale repercussions in how it enhances the field of autonomous robotics, which is arguably the main reason for carrying out this study. A task-oriented control implementation strategy that does not need any real programming is provided in this work. This could for instance allow for greater efficiency and versatility of robotic systems in sectors like medical or industrial automation and self-driving vehicles. Still, there is more science to do especially with the DQN where it is rather easy when it comes to such activities in the two-dimensional environment and with simple tasks – increasing the scale to real-life situations or tasks that are more complicated present's challenges. Issues like overfitting, sample inefficiency, or difficulties in handling continuous action spaces may arise. To mitigate these challenges, future research will explore the integration of hybrid RL approaches that combine the strengths of different algorithms or leverage transfer learning to enhance performance in more complex tasks.

Notwithstanding the encouraging outcomes, however, this particular research has certain limitations which are worth highlighting. First, the study was done in a controlled setting and using only one algorithm of reinforcement learning. While this arrangement worked in showing the possibility of DQN, it does not capture the complexities of real-world conditions, where other extraneous observations such as moving objects and noise and multi-action dimensions are very important. Moreover, there is no performance analysis of the DQN against other reinforcement learning approaches or conventional controller techniques.

Future study will attempt to expand the scope beyond simple contexts and encompass more complicated scenarios in order to overcome these constraints and build on our discoveries. Future research, for example, might investigate the use of multi-link robotic arms or more complex environments with dynamic targets or outside disruptions like barriers or shifting work environments. In order to ascertain which RL algorithms work best for different task difficulties, we also plan to compare DQN with other algorithms such as Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), and Deep Deterministic Policy Gradient (DDPG).

In addition, future research ought to investigate how this strategy may be incorporated into more intricate multi-link robotic arms or even multi-robot systems where agent cooperation and coordination are necessary. Adding robustness tests in environments with dynamic targets or external perturbations would provide further validation of the algorithm's effectiveness in real-world scenarios.

In conclusion, although DQN has exhibited impressive potential in the controlling of robots autonomously, further analytical studies and practical experiments on realistic approaches are required to understand its merits and demerits. This paper enunciates the groundwork for investigating the use of more advanced and pragmatic RL algorithms in the future control of robots.

REFERENCES

[1] C. Zhang and Y. Lu, "Study on artificial intelligence: The state of the art and future prospects," *Journal of Industrial Information Integration*, vol. 23, p. 100224, Sep. 2021, doi: 10.1016/j.jii.2021.100224.

[2] I. H. Sarker, "AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems," *SN Computer Science*, vol. 3, no. 2, Feb. 2022, doi: 10.1007/s42979-022-01043-x.

[3] P. Wang, "On Defining Artificial Intelligence," *Journal of Artificial General Intelligence*, vol. 10, no. 2, pp. 1–37, Jan. 2019, doi: 10.2478/jagi-2019-0002.

[4] S. P. Yadav, D. P. Mahato, and N. T. D. Linh, "Distributed Artificial Intelligence," *CRC Press*, 2020, doi: 10.1201/9781003038467.

[5] Y. Li and O. Hilliges, "Artificial Intelligence for Human Computer Interaction: A Modern Approach," *Springer International Publishing*, 2021, doi: 10.1007/978-3-030-82681-9.

[6] S. Kumar, A. K. Verma, and A. Mirza, "Digitalisation, Artificial Intelligence, IoT, and Industry 4.0 and Digital Society," *Digital Transformation, Artificial Intelligence and Society*, pp. 35–57, 2024, doi: 10.1007/978-981-97-5656-8_3.

[7] V. V. Krishna, "A I and contemporary challenges: The good, bad and the scary," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 1, p. 100178, Mar. 2024, doi: 10.1016/j.joitmc.2023.100178.

[8] W. Wang and K. Siau, "Artificial Intelligence, Machine Learning, Automation, Robotics, Future of Work and Future of Humanity," *Journal of Database Management*, vol. 30, no. 1, pp. 61–79, Jan. 2019, doi: 10.4018/jdm.2019010104.

[9] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics*, vol. 3, pp. 54–70, 2023, doi: 10.1016/j.cogr.2023.04.001.

[10] A. K. Tyagi, T. F. Fernandez, S. Mishra, and S. Kumari, "Intelligent Automation Systems at the Core of Industry 4.0," *Intelligent Systems Design and Applications*, pp. 1–18, 2021, doi: 10.1007/978-3-030-71187-0_1.

[11] T. V. N. Rao, A. Gaddam, M. Kurni, and K. Saritha, "Reliance on Artificial Intelligence, Machine Learning and Deep Learning in the Era of Industry 4.0," *Smart Healthcare System Design*, pp. 281–299, Jun. 2021, doi: 10.1002/9781119792253.ch12.

[12] L. Vandewinckele *et al*., "Overview of artificial intelligence-based applications in radiotherapy: Recommendations for implementation and quality assurance," *Radiotherapy and Oncology*, vol. 153, pp. 55–66, Dec. 2020, doi: 10.1016/j.radonc.2020.09.008.

[13] K. K. H. Ng, C.-H. Chen, C. K. M. Lee, J. Roger Jiao, and Z.-X. Yang, "A systematic literature review on intelligent automation: Aligning concepts from theory, practice, and future perspectives," *Advanced Engineering Informatics*, vol. 47, p. 101246, Jan. 2021, doi: 10.1016/j.aei.2021.101246.

[14] Y. Himeur *et al*., "AI-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 4929–5021, Oct. 2022, doi: 10.1007/s10462-022-10286-2.

[15] J.-A. Johannessen. *Artificial Intelligence, Automation and the Future of Competence at Work*. Routledge, Dec. 2020, doi: 10.4324/9781003121923.

[16] H. Chen et al., "From Automation System to Autonomous System: An Architecture Perspective," *Journal of Marine Science and Engineering*, vol. 9, no. 6, p. 645, Jun. 2021, doi: 10.3390/jmse9060645.

[17] F. Folgado, D. Calderón, I. González, and A. Calderón, "Review of Industry 4.0 from the Perspective of Automation and Supervision Systems: Definitions, Architectures and Recent Trends," *Electronics*, vol. 13, no. 4, p. 782, Feb. 2024, doi: 10.3390/electronics13040782.

[18] M. S. Xavier *et al*., "Soft Pneumatic Actuators: A Review of Design, Fabrication, Modeling, Sensing, Control and Applications," *IEEE Access*, vol. 10, pp. 59442–59485, 2022, doi: 10.1109/access.2022.3179589.

[19] D. Xie, L. Chen, L. Liu, L. Chen, and H. Wang, "Actuators and Sensors for Application in Agricultural Robots: A Review," *Machines*, vol. 10, no. 10, p. 913, Oct. 2022, doi: 10.3390/machines10100913.

[20] L. Martirano and M. Mitolo, "Building Automation and Control Systems (BACS): a Review," *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, pp. 1-8, Jun. 2020, doi: 10.1109/eeeic/icpseurope49358.2020.9160662.

[21] R. Stetter, "A Fuzzy Virtual Actuator for Automated Guided Vehicles," *Sensors*, vol. 20, no. 15, p. 4154, Jul. 2020, doi: 10.3390/s20154154.

[22] R. Sivapriyan, K. M. Rao, and M. Harijyothi, "Literature Review of IoT based Home Automation System," *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, pp. 101-105, Jan. 2020, doi: 10.1109/icisc47916.2020.9171149.

[23] S. M. Zinchenko, A. P. Ben, P. S. Nosov, I. S. Popovych, P. P. Mamenko, and V. M. Mateichuk, "Improving The Accuracy And Reliability Of Automatic Vessel Moution Control System," *Radio Electronics, Computer Science, Control*, no. 2, pp. 183–195, Sep. 2020, doi: 10.15588/1607-3274-2020-2-19.

[24] M. Bashabsheh, "Simulation of An Automatic System of Robotics for Artificial Animated Being Manufacturing Using AnyLogic Simulation Software," *International Journal of Electrical and Electronics Engineering*, vol. 11, no. 5, pp. 129–137, May 2024, doi: 10.14445/23488379/ijeee-v11i5p112.

[25] P. I. Kalandarov, Z. M. Mukimov, and A. M. Nigmatov, "Automatic Devices for Continuous Moisture Analysis of Industrial Automation Systems," *Proceedings of the 7th International Conference on Industrial Engineering (ICIE 2021)*, pp. 810–817, 2022, doi: 10.1007/978-3-030-85230-6_96.

[26] C. Xia *et al*., "A review on wire arc additive manufacturing: Monitoring, control and a framework of automated system," *Journal of Manufacturing Systems*, vol. 57, pp. 31–45, Oct. 2020, doi: 10.1016/j.jmsy.2020.08.008.

[27] Z. Van Veldhoven and J. Vanthienen, "Digital transformation as an interaction-driven perspective between business, society, and technology," *Electronic Markets*, vol. 32, no. 2, pp. 629–644, Mar. 2021, doi: 10.1007/s12525-021-00464-5.

[28] F. Vicentini, "Collaborative Robotics: A Survey," *Journal of Mechanical Design*, vol. 143, no. 4, Oct. 2020, doi: 10.1115/1.4046238.

[29] J. Zhu *et al*., "Challenges and Outlook in Robotic Manipulation of Deformable Objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, Sep. 2022, doi: 10.1109/mra.2022.3147415.

[30] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, "A survey of robot manipulation in contact," *Robotics and Autonomous Systems*, vol. 156, p. 104224, Oct. 2022, doi: 10.1016/j.robot.2022.104224.

[31] M. Bashabsheh, "Comprehensive and Simulated Modeling of a Centralized Transport Robot Control System," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 5, 2024, doi: 10.14569/ijacsa.2024.0150552.

[32] R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo, and J. Barata, "Industrial Artificial Intelligence in Industry 4.0 - Systematic Review, Challenges and Outlook," *IEEE Access*, vol. 8, pp. 220121–220139, 2020, doi: 10.1109/access.2020.3042874.

[33] D. Carou, A. Sartal, and J. P. Davim, "Machine Learning and Artificial Intelligence with Industrial Applications," *Springer International Publishing*, 2022, doi: 10.1007/978-3-030-91006-8.

[34] B. Ferreira and J. Reis, "A Systematic Literature Review on the Application of Automation in Logistics," *Logistics*, vol. 7, no. 4, p. 80, Nov. 2023, doi: 10.3390/logistics7040080.

[35] S.-H. Chung, "Applications of smart technologies in logistics and transport: A review," *Transportation Research Part E: Logistics and Transportation Review*, vol. 153, p. 102455, Sep. 2021, doi: 10.1016/j.tre.2021.102455.

[36] M. Raj and R. Seamans, "Primer on artificial intelligence and robotics," *Journal of Organization Design*, vol. 8, no. 1, May 2019, doi: 10.1186/s41469-019-0050-0.

[37] K. Rusia, S. Rai, A. Rai, and S. V. Kumar Karatangi, "Artificial Intelligence and Robotics: Impact & Open issues of automation in Workplace," *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, vol. 8, pp. 54–59, Mar. 2021, doi: 10.1109/icacite51222.2021.9404749.

[38] Y. Yang, Y. Wu, C. Li, X. Yang, and W. Chen, "Flexible Actuators for Soft Robotics," *Advanced Intelligent Systems*, vol. 2, no. 1, Dec. 2019, doi: 10.1002/aisy.201900077.

[39] L. Iocchi *et al*., "Development of intelligent service robots," *Intelligenza Artificiale*, vol. 7, no. 2, pp. 139–152, 2013, doi: 10.3233/ia-130055.

[40] A. Dzedzickis, J. Subačiūtė-Žemaitienė, E. Šutinys, U. Samukaitė-Bubnienė, and V. Bučinskas, "Advanced Applications of Industrial Robotics: New Trends and Possibilities," *Applied Sciences*, vol. 12, no. 1, p. 135, Dec. 2021, doi: 10.3390/app12010135.

[41] Z. Li, S. Li, and X. Luo, "An overview of calibration technology of industrial robots," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 23–36, Jan. 2021, doi: 10.1109/jas.2020.1003381.

[42] J. H. Jung and D.-G. Lim, "Industrial robots, employment growth, and labor cost: A simultaneous equation analysis," *Technological Forecasting and Social Change*, vol. 159, p. 120202, Oct. 2020, doi: 10.1016/j.techfore.2020.120202.

[43] X. Chen, X. Huang, Y. Wang, and X. Gao, "Combination of Augmented Reality Based Brain- Computer Interface and Computer Vision for High-Level Control of a Robotic Arm," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 3140–3147, Dec. 2020, doi: 10.1109/tnsre.2020.3038209.

[44] M. Raj and R. Seamans, "Primer on artificial intelligence and robotics," *Journal of Organization Design*, vol. 8, no. 1, May 2019, doi: 10.1186/s41469-019-0050-0.

[45] R. Azmeera, "Robotics Process Automation: Artificial Intelligence with SAP," *International Journal of Science and Research (IJSR)*, vol. 12, no. 11, pp. 1871–1876, Nov. 2023, doi: 10.21275/sr231126070716.

[46] S. Cebollada, L. Payá, M. Flores, A. Peidró, and O. Reinoso, "A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data," *Expert Systems with Applications*, vol. 167, p. 114195, Apr. 2021, doi: 10.1016/j.eswa.2020.114195.

[47] J. Long, J. Mou, L. Zhang, S. Zhang, and C. Li, "Attitude data-based deep hybrid learning architecture for intelligent fault diagnosis of multi-joint industrial robots," *Journal of Manufacturing Systems*, vol. 61, pp. 736–745, Oct. 2021, doi: 10.1016/j.jmsy.2020.08.010.

[48] K. Nam, C. S. Dutt, P. Chathoth, A. Daghfous, and M. S. Khan, "The adoption of artificial intelligence and robotics in the hotel industry: prospects and challenges," *Electronic Markets*, vol. 31, no. 3, pp. 553–574, Oct. 2020, doi: 10.1007/s12525-020-00442-3.

[49] M. Aljalal, S. Ibrahim, R. Djemal, and W. Ko, "Comprehensive review on brain-controlled mobile robots and robotic arms based on electroencephalography signals," *Intelligent Service Robotics*, vol. 13, no. 4, pp. 539–563, Jun. 2020, doi: 10.1007/s11370-020-00328-5.

[50] G. Shi, D. Li, Y. Ding, and Y. Q. Chen, "Desired dynamic equational proportional-integral-derivative controller design based on probabilistic robustness," *International Journal of Robust and Nonlinear Control*, vol. 32, no. 18, pp. 9556–9592, Jul. 2021, doi: 10.1002/rnc.5667.

[51] M. Samuel, M. Mohamad, M. Hussein, and S. M. Saad, "Lane Keeping Maneuvers Using Proportional Integral Derivative (PID) and Model Predictive Control (MPC)," *Journal of Robotics and Control (JRC)*, vol. 2, no. 2, 2021, doi: 10.18196/jrc.2256.

[52] A. R. Al Tahtawi, M. Agni, and T. D. Hendrawati, "Small-scale Robot Arm Design with Pick and Place Mission Based on Inverse Kinematics," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, 2021, doi: 10.18196/jrc.26124.

[53] L. Yiyang, J. Xi, B. Hongfei, W. Zhining, and S. Liangliang, "A General Robot Inverse Kinematics Solution Method Based on Improved PSO Algorithm," *IEEE Access*, vol. 9, pp. 32341–32350, 2021, doi: 10.1109/access.2021.3059714.

[54] J. A. Abdor-Sierra, E. A. Merchán-Cruz, and R. G. Rodríguez-Cañizo, "A comparative analysis of metaheuristic algorithms for solving the inverse kinematics of robot manipulators," *Results in Engineering*, vol. 16, p. 100597, Dec. 2022, doi: 10.1016/j.rineng.2022.100597.

[55] [32] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-Based Deep Learning," *Proceedings of the IEEE*, vol. 111, no. 5, pp. 465–499, May 2023, doi: 10.1109/jproc.2023.3247480.

[56] W. Li, H. Yuan, S. Li, and J. Zhu, "A Revisit to Model-Free Control," *IEEE Transactions on Power Electronics*, vol. 37, no. 12, pp. 14408–14421, Dec. 2022, doi: 10.1109/tpel.2022.3197692.

[57] B. Zhang and P. Liu, "Model-Based and Model-Free Robot Control: A Review," *RiTA* 2020, pp. 45–55, 2021, doi: 10.1007/978-981-16-4803-8_6.

[58] C. Zhang, C. Cen, and J. Huang, "An Overview of Model-Free Adaptive Control for the Wheeled Mobile Robot," *World Electric Vehicle Journal*, vol. 15, no. 9, p. 396, Aug. 2024, doi: 10.3390/wevj15090396.

[59] P. Ladosz, L. Weng, M. Kim, and H. Oh, "Exploration in deep reinforcement learning: A survey," *Information Fusion*, vol. 85, pp. 1–22, Sep. 2022, doi: 10.1016/j.inffus.2022.03.003.

[60] S. E. Li, "Deep Reinforcement Learning," *Reinforcement Learning for Sequential Decision and Optimal Control*, pp. 365–402, 2023, doi: 10.1007/978-981-19-7784-8_10.

[61] X. Wang *et al*., "Deep Reinforcement Learning: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5064–5078, Apr. 2024, doi: 10.1109/tnnls.2022.3207346.

[62] Y. Yang, L. Juntao, and P. Lingling, "Multi-robot path planning based on a deep reinforcement learning DQN algorithm," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 3, pp. 177–183, Aug. 2020, doi: 10.1049/trit.2020.0024.

[63] E. Khelifi, A. Saki, and U. Faghihi, "Causal Deep Q Networks," *Advances and Trends in Artificial Intelligence. Theory and Applications*, pp. 254–264, 2024, doi: 10.1007/978-981-97-4677-4_21.

[64] J. Li, Y. Chen, X. Zhao, and J. Huang, "An improved DQN path planning algorithm," *The Journal of Supercomputing*, vol. 78, no. 1, pp. 616–639, May 2021, doi: 10.1007/s11227-021-03878-2.

[65] J. Wan, X. Li, H.-N. Dai, A. Kusiak, M. Martinez-Garcia, and D. Li, "Artificial-Intelligence-Driven Customized Manufacturing Factory: Key Technologies, Applications, and Challenges," *Proceedings of the IEEE*, vol. 109, no. 4, pp. 377–398, Apr. 2021, doi: 10.1109/jproc.2020.3034808.

[66] J. Escobar-Naranjo, G. Caiza, P. Ayala, E. Jordan, C. A. Garcia, and M. V. Garcia, "Autonomous Navigation of Robots: Optimization with DQN," *Applied Sciences*, vol. 13, no. 12, p. 7202, Jun. 2023, doi: 10.3390/app13127202.

[67] Ó. Pérez-Gil *et al*., "DQN-Based Deep Reinforcement Learning for Autonomous Driving," *Advances in Physical Agents II*, pp. 60–76, Nov. 2020, doi: 10.1007/978-3-030-62579-5_5.

[68] Y.-C. Wu, T. Q. Dinh, Y. Fu, C. Lin, and T. Q. S. Quek, "A Hybrid DQN and Optimization Approach for Strategy and Resource Allocation in MEC Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4282–4295, Jul. 2021, doi: 10.1109/twc.2021.3057882.

[69] Q. Wu *et al*., "Position Control of Cable-Driven Robotic Soft Arm Based on Deep Reinforcement Learning," *Information*, vol. 11, no. 6, p. 310, Jun. 2020, doi: 10.3390/info11060310.

[70] J. Hwangbo *et al*., "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, Jan. 2019, doi: 10.1126/scirobotics.aau5872.

[71] H. Liang, X. Lou, Y. Yang, and C. Choi, "Learning Visual Affordances with Target-Orientated Deep Q-Network to Grasp Objects by Harnessing Environmental Fixtures," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2562-2568, May 2021, doi: 10.1109/icra48506.2021.9561737.

[72] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, Oct. 2021, doi: 10.26599/tst.2021.9010012.

[73] H. Li, Q. Zhang, and D. Zhao, "Deep Reinforcement Learning-Based Automatic Exploration for Navigation in Unknown Environment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2064–2076, Jun. 2020, doi: 10.1109/tnnls.2019.2927869.

[74] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey," *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 737-744, Dec. 2020, doi: 10.1109/ssci47803.2020.9308468.

[75] N. Gupta and P. K. Gupta, "Robotics and Artificial Intelligence (AI) in Agriculture with Major Emphasis on Food Crops," *Digital Agriculture*, pp. 577–605, 2024, doi: 10.1007/978-3-031-43548-5_19.

[76] A. Jafari-Tabrizi and D. P. Gruber, "Reinforcement-Learning-based Control of an Industrial Robotic Arm for Following a Randomly-Generated 2D-Trajectory," *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, vol. 518, pp. 1–6, Aug. 2021, doi: 10.1109/coins51742.2021.9524158.

[77] D. Han, B. Mulyana, V. Stankovic, and S. Cheng, "A Survey on Deep Reinforcement Learning Algorithms for Robotic Manipulation," *Sensors*, vol. 23, no. 7, p. 3762, Apr. 2023, doi: 10.3390/s23073762.

[78] M. Al-Gabalawy, "Path planning of robotic arm based on deep reinforcement learning algorithm," *Advanced Control for Applications*, vol. 4, no. 1, Mar. 2022, doi: 10.1002/adc2.79.

[79] S. Balhara *et al*., "A survey on deep reinforcement learning architectures, applications and emerging trends," *IET Communications*, Jul. 2022, doi: 10.1049/cmu2.12447.

[80] M. Botvinick, J. X. Wang, W. Dabney, K. J. Miller, and Z. Kurth-Nelson, "Deep Reinforcement Learning and Its Neuroscientific Implications," *Neuron*, vol. 107, no. 4, pp. 603–616, Aug. 2020, doi: 10.1016/j.neuron.2020.06.014.

[81] S. Carta, A. Ferreira, A. S. Podda, D. Reforgiato Recupero, and A. Sanna, "Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting," *Expert Systems with Applications*, vol. 164, p. 113820, Feb. 2021, doi: 10.1016/j.eswa.2020.113820.

[82] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based Reinforcement Learning: A Survey," *Foundations and Trends® in Machine Learning*, vol. 16, no. 1, pp. 1–118, 2023, doi: 10.1561/2200000086.

[83] T. Zhang and H. Mo, "Reinforcement learning for robot research: A comprehensive review and open issues," *International Journal of Advanced Robotic Systems*, vol. 18, no. 3, p. 172988142110073, May 2021, doi: 10.1177/17298814211007305.

[84] Q. Huang, "Model-Based or Model-Free, a Review of Approaches in Reinforcement Learning," *2020 International Conference on Computing and Data Science (CDS)*, pp. 219-221, Aug. 2020, doi: 10.1109/cds49703.2020.00051.

[85] A. Moudgalya, A. Shafi, and B. A. Arun, "A Comparative Study of Model-Free Reinforcement Learning Approaches," *Advanced Machine Learning Technologies and Applications*, pp. 547–557, May 2020, doi: 10.1007/978-981-15-3383-9_50.

[86] M. Sewak, S. K. Sahay, and H. Rathore, "Value-Approximation based Deep Reinforcement Learning Techniques: An Overview," *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, pp. 379–384, Oct. 2020, doi: 10.1109/iccca49541.2020.9250787.

[87] Y. Zhao, Y. Wang, Y. Tan, J. Zhang, and H. Yu, "Dynamic Jobshop Scheduling Algorithm Based on Deep Q Network," *IEEE Access*, vol. 9, pp. 122995–123011, 2021, doi: 10.1109/access.2021.3110242.

[88] Y. T. Quek, L. L. Koh, N. T. Koh, W. A. Tso, and W. L. Woo, "Deep Q-network implementation for simulated autonomous vehicle control," *IET Intelligent Transport Systems*, vol. 15, no. 7, pp. 875–885, May 2021, doi: 10.1049/itr2.12067.

[89] T. Li, X. Zhu, and X. Liu, "An End-to-End Network Slicing Algorithm Based on Deep Q-Learning for 5G Network," *IEEE Access*, vol. 8, pp. 122229–122240, 2020, doi: 10.1109/access.2020.3006502.

[90] Y. Huang, "Deep Q-Networks," *Deep Reinforcement Learning*, pp. 135–160, 2020, doi: 10.1007/978-981-15-4095-0_4.

[91] H. Zhang, R. Huang, and S. Zhang, "Integrating Learning and Planning," *Deep Reinforcement Learning*, pp. 307–316, 2020, doi: 10.1007/978-981-15-4095-0_9.

[92] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/tnnls.2021.3084827.

[93] S. Zhao and B. Zhang, "Learning Salient and Discriminative Descriptor for Palmprint Feature Extraction and Identification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5219–5230, Dec. 2020, doi: 10.1109/tnnls.2020.2964799

[94] Z. Lu and R. Huang, "Autonomous mobile robot navigation in uncertain dynamic environments based on deep reinforcement learning," *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, vol. 518, pp. 423–428, Jul. 2021, doi: 10.1109/rcar52367.2021.9517635.

[95] W. Guan, W. Luo, and Z. Cui, "Intelligent decision-making system for multiple marine autonomous surface ships based on deep reinforcement learning," *Robotics and Autonomous Systems*, vol. 172, p. 104587, Feb. 2024, doi: 10.1016/j.robot.2023.104587.

[96] V. Mnih *et al*., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.

[97] H. Wang *et al*., "Deep reinforcement learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 12, pp. 1726–1744, Oct. 2020, doi: 10.1631/fitee.1900533.

[98] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," *Knowledge-Based Systems*, vol. 214, p. 106685, Feb. 2021, doi: 10.1016/j.knosys.2020.106685.

[99] A. Iftikhar, M. A. Ghazanfar, M. Ayub, S. Ali Alahmari, N. Qazi, and J. Wall, "A reinforcement learning recommender system using bi-clustering and Markov Decision Process," *Expert Systems with Applications*, vol. 237, p. 121541, Mar. 2024, doi: 10.1016/j.eswa.2023.121541.

[100] V. Zobernig *et al*., "RangL: A Reinforcement Learning Competition Platform," *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4168309.

[101] N. Dalla Pozza, L. Buffoni, S. Martina, and F. Caruso, "Quantum reinforcement learning: the maze problem," *Quantum Machine Intelligence*, vol. 4, no. 1, May 2022, doi: 10.1007/s42484-022-00068-y.