Deep Reinforcement Learning Based Robotic Arm Control Simulation to Execute Object Reaching Task for Industrial Application

John Mark Correa[1], Rudolph Joshua Candare[2], Junrie B. Matias[3]
Department of Computer Science, Caraga State University, Butuan City, Philippines[1]
Graduate School, Caraga State University, Butuan City, Philippines[2, 3]

***Abstract*—This study presents a deep reinforcement learning (DRL) approach to train a robotic arm for object reaching tasks in industrial settings, eliminating the need for traditional task-specific programming. Leveraging the Proximal Policy Optimization (PPO) algorithm for its stability in continuous control, the system learns optimal behaviors through autonomous trial-and-error. Central to this work is reward shaping, where structured feedback based on distance to the target, collision avoidance, motion constraints, and step efficiency guides the agent, akin to incremental coaching. A simulated industrial environment was developed using Webots, integrated with OpenAI Gym and Stable-Baselines3, enabling safe training with sensor data (camera, distance sensor) and randomized target placements. Three models with varying reward schemes were evaluated: simpler rewards prioritized rapid convergence, while complex formulations (e.g., perceptual alignment) enhanced long-term accuracy at the cost of initial instability. Experimental results demonstrated that reward shaping reduced the required steps, highlighting its role in accelerating learning. The study underscores the efficacy of combining DRL, simulation-based training, and adaptive reward design to develop efficient robotic controllers. These findings advance scalable solutions for industrial automation, emphasizing the trade-offs between reward complexity and policy convergence. Future work will refine reward functions to bridge simulation-to-reality gaps, fostering practical adoption in manufacturing and assembly systems.**

***Keywords*—Reinforcement learning; deep reinforcement learning; reward shaping techniques; robotic arm; robot simulation**

I. Introduction

Humans possess the innate ability to easily recognize and manipulate objects; conversely, robots encounter significant challenges with this ostensibly straightforward task, which presents a noteworthy obstacle within the field of robotics [1]. This phenomenon has garnered attention in both scholarly research and industrial applications, particularly in relation to Industry 4.0, which prioritizes automation and sophisticated technologies in manufacturing through paradigms such as the Smart Factory [2].

Reinforcement learning (RL) has surfaced as a promising alternative to facilitate the implementation of robots within industrial contexts by enabling robots to acquire tasks without the need for direct programming [3]. Deep reinforcement learning (DRL) integrates RL with deep neural networks to address complex state spaces and enhance the efficiency of the learning process. Although DRL may demand substantial resources for training on actual platforms, simulation-based training represents a feasible solution to scale data and decrease reliance on human intervention [4].

Simulation-based environments can furnish a secure and regulated context for robots to practice and hone their competencies, thereby equipping them to generalize acquired behaviors to real-world applications with greater efficacy [5]. This methodology not only accelerates the learning trajectory but also mitigates risks associated with training in unpredictable real-world conditions, ultimately fostering the development of more resilient and adaptable robotic systems [6]. By capitalizing on advancements in virtual reality and high-fidelity simulations, researchers are able to devise diverse training scenarios that closely replicate real-world challenges, thereby further enhancing the adaptability of DRL algorithms [7]. These innovations lay the groundwork for more effective training methodologies, enabling robots to address complex tasks with heightened confidence and precision [7].

Consequently, the assimilation of these technologies into robotic training programs is poised to transform not only the manner in which robots learn but also their overall efficacy in dynamic environments [8]. This paradigm shift in training methodologies is likely to yield substantial advancements across a variety of applications, ranging from autonomous vehicles navigating congested thoroughfares to industrial robots optimizing production lines with minimal human involvement [9].

However, a critical gap remains in understanding how reward shaping, specifically the design and modular integration of reward signals, affects convergence efficiency and task precision across varying degrees of complexity [10], [11]. Most prior works emphasize algorithmic improvements or domain adaptation, yet underexplore the influence of structured, incremental reward functions within high-fidelity simulations. These reward strategies are essential not only for accelerating convergence but also for fostering generalizable behaviors applicable to industrial use cases [12].

To address this gap, this study poses the following research question: How does the complexity of reward functions influence the convergence efficiency and task success rate of a DRL-based robotic arm control system in a simulated industrial environment?

This study addresses this question by proposing a DRL-based robotic arm control system using Proximal Policy Optimization (PPO) within a Webots simulation environment. By leveraging the recent advancements in continuous control through deep reinforcement learning (DRL) to enhance the operational efficiency of a robotic arm tasked with object-reaching activities. Specifically, this study employs the Proximal Policy Optimization (PPO) algorithm within a three-dimensional simulation environment, guided by a strategically designed reward function. The primary objective is to develop and train a DRL-based control system that enables the robotic arm to accurately reach and make contact with a designated target object, sometimes at a specified angle, an ability with broad applicability across labor-intensive industries. The proposed reward function is crafted to facilitate the convergence of the PPO algorithm, ensuring effective policy learning. Additionally, the study explores the adaptability of the learning framework by modifying the reward function to accommodate similar manipulation tasks of varying complexity. Comparative evaluations of different reward functions will be conducted to determine their influence on the convergence speed and overall performance of the robotic arm, thereby contributing to the design of more efficient and adaptable robotic control systems.

II. Related Work

Deep reinforcement learning (DRL) has emerged as a powerful paradigm for controlling robotic arms, offering robust solutions to complex control tasks. This approach combines the strengths of deep learning and reinforcement learning, enabling robots to learn optimal policies through trial and error in a dynamic environment. Recent advancements in DRL have significantly improved the accuracy, efficiency, and adaptability of robotic arm control systems, addressing challenges such as trajectory tracking, obstacle avoidance, and energy efficiency. This section provides a comprehensive overview of the state-of-the-art methods, key innovations, and applications of DRL in robotic arm control, drawing insights from recent research papers.

A. Algorithmic Advancements in DRL for Robotic Arm Control

One of the most notable contributions to DRL for robotic arm control is the improvement of existing algorithms. Proximal Policy Optimization (PPO) has been widely adopted due to its stability and effectiveness in continuous action spaces [13]. For instance, researchers have proposed an Improved-PPO algorithm that integrates PPO with Model Predictive Control (MPC) to enhance trajectory tracking efficiency. This method has demonstrated a significant increase in convergence speed, outperforming traditional PPO and A3C algorithms by 84.3% and 15.4%, respectively [13]. Similarly, the combination of PPO with Generative Adversarial Imitation Learning (GAIL) has been explored to transfer policies from simulated environments to real-world scenarios, enabling dual-arm robots to perform complex assembly tasks with remarkable efficiency [14].

Another significant advancement is the use of Deep Deterministic Policy Gradient (DDPG) algorithms. Researchers have enhanced DDPG by incorporating techniques such as Hindsight Experience Replay (HER) and double experience replay buffers to improve learning efficiency. These modifications have enabled robotic arms to achieve near-perfect success rates in target-reaching tasks while reducing training time [15], [16]. Additionally, the integration of DDPG with sequential training methods has been shown to expedite the learning process and improve robustness against external disturbances, making it suitable for real-world applications [17].

B. Reward Function Design and Motion Planning

The design of reward functions plays a crucial role in DRL, as it guides the learning process and determines the optimal policy [18]. Researchers have proposed various reward functions tailored to specific tasks, such as trajectory tracking, obstacle avoidance, and energy efficiency [19]. For example, a study introduced a hierarchical reward function that combines motion accuracy, obstacle avoidance, and energy-saving components. This approach has been shown to improve the convergence speed and accuracy of robotic arm control policies [20], [21].

In addition to reward function design, motion planning has been a focal area of research. A study proposed a DRL-based motion planning method that enables robotic arms to navigate complex environments with obstacles. The method leverages the PPO algorithm and reward shaping techniques to ensure efficient and adaptive control [20]. Another study demonstrated the effectiveness of DRL in motion planning for dual-arm robots, where a novel reward and punishment function was designed to guide the robot to approach targets while avoiding collisions [22].

C. Generalization and Real-World Deployment

The generalization capabilities of DRL controllers are essential for their successful deployment in real-world scenarios. Researchers have investigated the generalization of policies learned in simulation to new observations, dynamics, and tasks. For example, a study demonstrated that a DRL controller trained in simulation could generalize well to real-world scenarios, including dynamic trajectory tracking and robustness to external forces [23]. Another study showed that policies trained in simulation could be directly transferred to physical robots without retraining, achieving remarkable performance in tasks such as grasping and motion planning [24].

The generalization capabilities of DRL have also been tested in complex environments, such as those involving soft robotic arms. A study presented a closed-loop controller for soft robotic arms that could generalize to new tasks, including intercepting moving objects and tracking trajectories with varying velocity profiles [23]. These findings highlight the potential of DRL for real-world applications, where robots must adapt to diverse and dynamic environments.

Deep reinforcement learning has revolutionized the field of robotic arm control, offering robust and adaptive solutions to complex control tasks. Recent advancements in algorithms, reward function design, and simulation-to-reality transfer have significantly improved the performance and versatility of DRL-based control systems [25]. As research continues to address challenges such as generalization, energy efficiency, and real-world deployment, DRL is poised to play an increasingly important role in the development of intelligent robotic systems [26].

D. Identified Problems in the Current Implementation of DRL in Robotic Arm Control

The current implementation of Deep Reinforcement Learning (DRL) in the control of robotic arms has revealed several critical issues that warrant attention. One of the primary challenges is the inefficiency in training time, which often results from the high dimensionality of the action and state spaces involved. This complexity can lead to prolonged learning periods and suboptimal performance, as the algorithms struggle to converge on effective strategies [26].

Additionally, there is a notable concern regarding the stability and robustness of the learned policies. Many DRL algorithms are sensitive to hyperparameter settings, which can result in significant variations in performance across different trials [16]. This unpredictability can hinder the reliability of robotic arm operations in real-world applications where consistency is paramount [27].

Another problem is the insufficient exploration of the action space during the training phase. Many DRL implementations tend to exploit known strategies rather than exploring new ones, potentially leading to local optima and a lack of adaptability in dynamic environments. This limited exploration can restrict the robot's ability to handle unforeseen circumstances or variations in tasks [28]. Moreover, the integration of DRL with sensory feedback systems poses additional challenges. Ensuring that the robotic arm can effectively interpret and respond to sensory inputs in real-time is crucial for achieving seamless operation. However, current implementations may struggle with the synchronization of sensory data processing and decision-making, leading to delays and inaccuracies in task execution [29].

Lastly, there is a growing concern about the generalization of learned behaviors across different tasks and environments. Many DRL models tend to overfit to the specific conditions present during training, which can significantly diminish their performance when faced with novel situations or variations in operational contexts [30]. Addressing these issues is essential for advancing the efficacy and applicability of DRL in robotic arm control systems, paving the way for more sophisticated and versatile robotic applications.

III. Methodology

A. Introduction to the Simulation Environment

The simulation environment was developed using Webots, an open-source, multi-platform robotic simulator designed for modeling, programming, and testing robotic systems. The setup emulated a collaborative workspace with a Universal Robots UR5e robotic arm, a 6-degree-of-freedom manipulator equipped with a Tiago Gripper. Key components integrated into the system included:

- A camera with an object recognition node mounted on the gripper to capture visual data.
- A distance sensor is attached to the end-effector to measure proximity to the target.
- A 5 cm^3 cube serving as the target object, randomly repositioned within a 3 m^3 boundary in front of the robot for each training episode.

The environment was designed to simulate industrial scenarios, where the robotic arm must dynamically adapt to varying object positions. The camera transmitted real-time images to the control system, which processed them to generate state observations (e.g., joint angles, end-effector coordinates).

B. Design of the Reward Functions

The performance of deep reinforcement learning (DRL) agents heavily depends on the quality and structure of the reward functions used during training. In this study, three models (A, B, and C) were developed with progressively complex reward functions to guide the robotic arm in learning the object-reaching task. Each reward function was designed to encourage desirable behavior and penalize inefficiencies or errors, thereby accelerating convergence toward optimal policies.

1) Distance-based reward.

a) A primary reward proportional to the Euclidean distance between the end-effector and target:

$$\mathrm{r} = \begin{cases} 10, & \mathrm{dis} < 0.01 \\ -2, & \mathrm{dis} > 1.25 \end{cases} \quad (1)$$

b) Incremental rewards (+2/-1) encouraged movement toward the target.

2) Motion constraints. Penalties (-2) enforced spatial boundaries to limit exploration to relevant areas.

3) Collision avoidance. Negative rewards (-5) deterred collisions with the robot's body or environment.

4) Object recognition. Positive rewards (+1) for detecting the target via the camera.

5) Image-based rewards (Models B and C).

a) Size ratio rewards (Model B): Scaled rewards based on changes in the target's perceived size.

$$\mathrm{size\_ratio} = \mathrm{box\_image\_size} \,/\, \mathrm{prev\_box\_image\ size} \quad (2)$$

$$\mathrm{r} = \mathrm{size\_ratio} \times 2 \quad (3)$$

b) Alignment rewards (Model C): Penalized deviations between the target's center and the camera's focal point.

$$dis\_to\_ctr = \sqrt{\left(\frac{x_{box\_img} + width_{cam}}{2}\right)^2 + \left(\frac{y_{box\_img} + height_{cam}}{2}\right)^2} \quad (4)$$

$$r = \begin{cases} 10, dis\_to\_ctr < 0.1 \\ 5, dis\_to\_ctr < 0.5 \\ 2, dis\_to\_ctr \leq 0.9 \\ -\ dis\_to\_ctr \times 2, dis\_to\_ctr > 0.09 \end{cases} \quad (5)$$

C. Implementation of the Control Mechanism and PPO Algorithm

The control policy was optimized using the Proximal Policy Optimization (PPO) algorithm, selected for its stability in continuous action spaces. Key implementation details included:

a) Policy network: A multi-layer perceptron (MLP) with fully connected layers to map states (joint angles, sensor data) to actions (joint velocity commands).

b) Training parameters: n_steps=2048 interactions per policy update. tensorboard_log for tracking training metrics (e.g., reward trends, episode length).

c) Environment interface: Custom OpenAI Gym integration to synchronize Webots simulations with the Stable-Baselines3 RL library.

The learning process mirrored trial-and-error refinement, where the agent iteratively adjusted its policy based on reward signals, gradually minimizing unnecessary movements and collisions.

D. Experiment and Evaluation

To systematically evaluate the impact of varying reward function complexities on the performance of the Proximal Policy Optimization (PPO) algorithm, a series of experiments were conducted using three distinct models. The primary evaluation criteria included the number of training steps required to achieve effective task performance and the success rate in executing the object-reaching task within a simulated environment.

The training protocol involved 15 rounds of training for each model, with 10,000 interaction steps per round. The models were designed to incrementally incorporate additional reward components, enabling a controlled analysis of how increasing reward complexity influences policy learning.

The models are defined as follows:

- Model A (Baseline): Utilizes a foundational reward structure based on the Euclidean distance between the end-effector and the target object as defined in Eq. (1), augmented with penalties and rewards for collision avoidance, object detection, and motion constraints.
- Model B (Image-Augmented Reward): Builds upon Model A by incorporating image-based size ratio rewards, leveraging Eq. (2) and Eq. (3). These rewards encourage the agent to maximize the perceived size of the target in the camera's view, thereby reinforcing movement toward the object.
- Model C (Alignment-Based Reward): Extends Model B by introducing camera alignment rewards, as described in Eq. (4) and Eq. (5). This component penalizes misalignment between the camera's focal center and the centroid of the target object, promoting not only reaching accuracy but also visual alignment for precise manipulation.

Throughout training, each model's performance was monitored using TensorBoard logs, which recorded metrics such as mean reward values and episode lengths over time. Furthermore, success rates were tested in controlled batches of 10, 30, and 50 episodes to assess consistency and robustness in task execution. This experimental design allowed for a comparative analysis of how reward structure influences learning stability, convergence speed, and task success.

Each model was subjected to 15 training rounds, with each round comprising 10,000 interaction steps between the agent and the simulated environment. The performance of the models was evaluated using two primary metrics:

a) Convergence efficiency: Assessed through TensorBoard visualizations, which tracked the progression of mean episode rewards and episode lengths over the course of training. These indicators provided insight into the stability and speed with which each model approached optimal policy learning.

b) Task success rate: Evaluated by conducting a series of 10-, 30-, and 50-episode trials, wherein a successful trial was defined as the robotic arm's end-effector making contact with the target object within a maximum of 250 time steps.

This evaluation framework was designed to highlight the influence of increasing reward complexity on both the stability of the learning process and the agent's ability to consistently complete the object-reaching task. By comparing the results across the three models, the study provides insights into the trade-offs between reward structure simplicity and control precision in DRL-based robotic systems.

IV. Results and Discussion

A. Initial Development of DRL-Based Robot Arm Controller

Fig. 1 illustrates the simulation environment created using Webots, an open-source and multi-platform desktop application used for modeling, programming, and simulation of robotic systems. The researcher developed a simulation based on the Universal Robots UR5e's collaborative robot arm with 6 degrees of freedom, which was equipped with a Tiago Gripper, an imaging device possessing an object recognition node, a solid node functioning concurrently as a distance sensor and end effector, as well as a cube box serving as the target object. The cube box was strategically positioned in a random manner in front of the robotic arm for each interaction, which is pivotal in motivating the agent to acquire skills in searching for the object within the designated space, thereby inhibiting it from merely learning and concentrating on a target object with a fixed position.

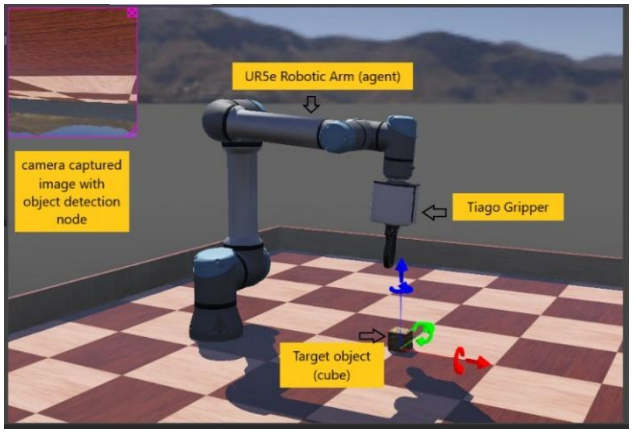

Fig. 1. Simulation environment and robot arm (Agent).

In the simulation environment, the camera transmits the collected image back to the computer. After the image is processed, it will be used as the state S to be observed by the actor. The computer will determine the next move distance of the end of the robotic arm on the X-axis, Y-axis, and Z-axis. According to the current coordinate value of the end of the robot arm and the rotation angle of each joint of the current robot arm,

the computer will calculate the angle at which each joint of the robot arm should rotate when the end of the robot arm reaches the next position.

B. Convergence Efficiency

By analyzing the mean reward values accumulated across iterations during the training process, where the horizontal axis represents the number of training iterations and the vertical axis denotes the reward value obtained per iteration. Fig. 2, Fig. 3 and Fig. 4 illustrate the performance trends of the three models examined in this study. The visualizations highlight how each model progressed and adapted over time during training. To facilitate this analysis, TensorBoard was employed as the primary tool for data processing and visualization, offering a comprehensive suite of utilities for monitoring, evaluating, and interpreting machine learning experiments. These visualizations illustrate the accumulation of reward values obtained across training iterations. The table on the left presents the mean episode length, measured by the number of timesteps or interactions with the environment. Here, the horizontal axis corresponds to the total number of iterations throughout the training process. In contrast, the table on the right displays the distribution of reward values earned in each iteration, with the horizontal axis indicating the iteration count and the vertical axis representing the reward value achieved. These visualizations provide insight into both the learning progress and the stability of the models over time.

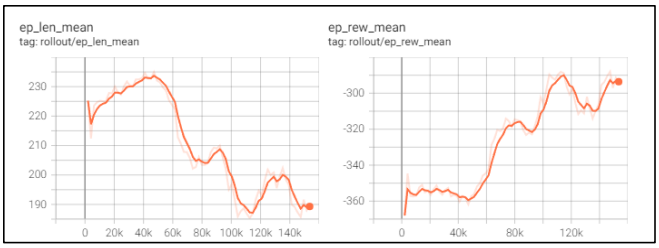

Fig. 2. Model A – Using baseline reward functions (Tensorboard).

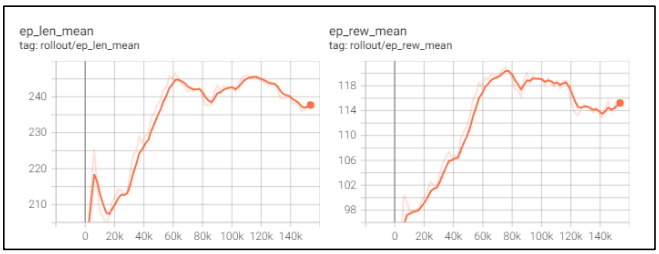

Fig. 3. Model B – Improved Model A with object image size-based reward function (Tensorboard).

The results derived from Model A, as illustrated in Fig. 2, reveal a favorable trajectory in the agent's performance over the temporal dimension. As the training regimen advances, it becomes evident that the quantity of actions necessary to fulfill the task diminishes while the resultant reward amplifies. This indicates that the agent is progressively enhancing its efficacy in making superior and more efficient control decisions to successfully execute the intended task.

In other terms, the agent is acquiring knowledge from the surrounding environment and modifying its conduct in response to the reward signal delineated by the reward function. As the agent develops, it executes movements with greater accuracy and precision in its pursuit of the target object. This phenomenon can be interpreted as an enhancement in the agent's control and decision-making proficiencies, constituting a favorable indication that the training protocol is functioning as anticipated. The observed trend of reduced actions coupled with elevated rewards signifies that the agent is incrementally converging towards the optimal resolution for the task, and it is highly probable that it will continue to experience improvement over time with ongoing training.

Fig. 3 underscores the notion that the architecture of the reward function can substantially influence the conduct of the agent. In this particular instance, the reward function of Model B may not have sufficiently motivated the agent to pursue the most expedient trajectory toward the target object. Conversely, it is plausible that Model B's reward function has prioritized alternative considerations, such as aligning the end-effector with the target object's orientation or maintaining adherence to the defined movement constraints, thus resulting in an elongated route to attain the target object. Irrespective of the underlying factors, the behavioral divergence between Model A and Model B accentuates the critical necessity of meticulously evaluating the formulation of the reward function during the training of a deep reinforcement learning-based agent. It further underscores the imperative for additional inquiry to ascertain the most effective reward function tailored to the specific objective of object reaching.

Moreover, the escalation in the number of actions necessitated in Model B signifies an extended training duration to achieve an optimal policy and thus could be a salient consideration when selecting a reward function for practical robotic applications. Subsequent investigations could delve deeper into this dichotomy between the efficacy of various reward functions and the temporal investment required to attain an optimal policy.

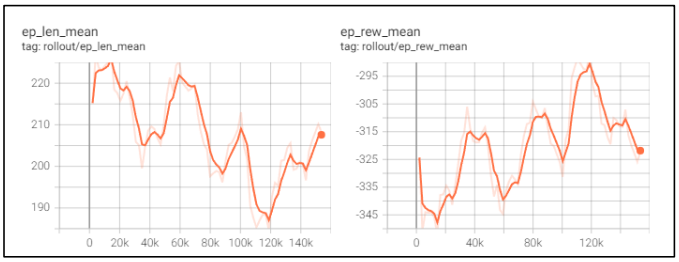

Fig. 4. Model C – Improved model B with object image and camera center distance reward function (Tensorboard).

Lastly, the findings derived from Model C, as illustrated in Fig. 4, demonstrate that, although the overall efficacy of the agent trained utilizing this reward function is commendable, there exist considerable variances in the rewards obtained throughout the training regimen. This phenomenon may be attributable to the complexity of the reward function employed in Model C, which incorporates a broader array of rewards, thereby rendering it more susceptible to fluctuations in the environmental context. Consequently, the agent may struggle to effectively adjust its parameters between successive training iterations, resulting in abrupt alterations in the rewards it acquires. Nevertheless, the overarching trajectory of the training

results for Model C remains favorable, indicating that the agent can acquire knowledge and advance towards the successful completion of the designated task.

C. Task Success Rate

The researchers executed a simulation experiment aimed at assessing the precision of three trained models by employing distinct reward functions. The experiment entailed the agent's gripper, or end-effector, endeavoring to approach the target object as closely as possible and establish contact with it. The efficacy of the models was appraised through 10-episode, 30-episode, and 50-episode trials, with the resultant data encapsulated in Table I.

TABLE I. RESULTS OF THE REACH AND CONTACT EXPERIMENT

Trained Model	Number of Successful Object Reach Actions		
	10 experiments	*30 experiments*	*50 experiments*
A	8	24	37
B	9	22	38
C	8	26	40

Results derived from the reach and contact experiment delineated in Table I indicate that all three models exhibit commendable efficacy with regard to their proficiency in reaching the target object and establishing contact with it. Nonetheless, there exist discernible variances in their performance, as evidenced by the number of successful trials conducted. A plausible explanation for these discrepancies in performance may pertain to the influence of the reward functions implemented in each model. It is plausible that Model A, characterized by its optimized reward function, possesses an enhanced capacity to concentrate on the specified task and acquire the optimal policy at an expedited rate, thereby culminating in superior performance in the reach and contact experiment. An additional factor that may contribute to these differences could be the training methodology employed. The variability in rewards observed in Model C may have precipitated a less stable learning trajectory, which could elucidate its marginally inferior performance in comparison to the other two models.

These findings imply that the proposed deep reinforcement learning-based robotic arm control system demonstrates the capability to execute reaching tasks with substantial effectiveness. Future investigations may be pursued to further refine the reward functions and training methodologies to augment the system's performance.

D. Implications

The agent exhibited exemplary behavior in executing the reaching task, attributable to the effective integration of diverse reward functions within the training process of the robotic arm. The reward functions encompassed parameters such as constraining the number of steps during interaction, quantifying the distance between the centroid of the object image and the centroid of the comprehensive image acquired by the camera recognition module, assessing the dimensions of the object, and identifying potential collisions, thereby furnishing the agent with affirmative or negative reinforcement that either promoted or dissuaded specific actions. These reward functions facilitate the agent's acquisition of the requisite behavior by directing it to recognize the object and its spatial coordinates, orient the end-effector towards the target object, synchronize the image captured by the camera recognition module, and circumvent collisions with the surrounding environment. Moreover, through the modulation of its movements, the agent was able to accomplish the task within the predetermined number of steps and maintain the target object within the defined parameters.

It can be asserted that the incorporation of these reward functions into the training process exerted a significant influence on the agent's performance in the object-reaching task, thereby optimizing both its behavior and resulting outcomes.

The incorporation of these carefully designed reward functions significantly influenced the agent's performance, enhancing both convergence and task efficiency. When comparing the performance of three distinct models trained using different reward structures. The following key observations emerged:

1) Model A matched or exceeded benchmark performances in terms of efficiency and reliability, validating the hypothesis that simplified but well-targeted reward functions improve learning stability.

2) Model B demonstrated slightly better success rates (76%) than Model A in the 50-episode trial, consistent with [22], where added perception cues enhanced accuracy, albeit at the cost of longer training.

3) Model C reached the highest success rate (80%) but required ~30% more training steps and suffered from high reward variance, paralleling issues in over-engineered reward systems like those explored in [10], [28].

Model A demonstrates a favorable trade-off between the simplicity of the reward function and training efficiency. Despite the absence of vision-based shaping used in Model B and C, Model A achieved the shortest convergence time (~70k steps) and competitive performance (74% success rate), with performance matching reward-heavy implementations from related studies [10], [22], [28] in both stability and training duration.

Although Model C introduced higher perceptual alignment (center-distance feedback), it required longer convergence (90k+ steps) and displayed reward fluctuations similar to [10] and [28], which reported volatility due to over-engineered shaping signals.

These results validate that while complex reward functions may offer incremental performance improvements, they often come with increased training cost and instability. The integration of well-structured, goal-specific rewards, particularly those balancing simplicity and effectiveness, is therefore essential in shaping robust robotic control behaviors. Our findings affirm the viability of deep reinforcement learning (DRL) approaches for dynamic object-reaching tasks in simulated environments and highlight the strategic value of reward function optimization.

V. CONCLUSION

This study affirms the feasibility of using deep reinforcement learning (DRL) to train a robotic arm controller in a simulated environment, even with limited computational resources. By leveraging the open-source simulation platform Webots alongside OpenAI Gym and Stable-Baselines3, the research successfully enabled a robotic arm to autonomously and consistently perform object-reaching tasks. Central to this success was the strategic use of reward shaping, a technique similar to guided learning, where the agent receives continuous feedback through structured rewards and penalties to encourage desired behavior. Three models were developed and evaluated to examine the impact of different reward strategies. Model A utilized a baseline reward function incorporating step limits, distance penalties, and collision detection. Model B introduced perceptual feedback by rewarding increases in the visible size of the target object, while Model C further refined agent behavior by penalizing misalignment between the camera's focal center and the object. Although Models B and C showed improved precision in object reaching, their complex reward formulations led to slower convergence and greater variability during training. In contrast, Model A's simplicity facilitated faster learning and more stable performance, highlighting the critical trade-off between reward granularity and training efficiency.

The findings demonstrate that DRL, when integrated with well-calibrated reward functions in a high-fidelity simulation, can generate stable, efficient robotic control policies without relying on hard-coded programming or manual trajectory design. Much like structured, age-appropriate guidance supports a child's learning process, a well-balanced reward architecture enables the agent to generalize optimal behavior through trial and error. However, excessively vague or overly intricate feedback can hinder progress or destabilize learning.

Ultimately, this research underscores the promise of DRL as a scalable and cost-effective approach to robotic automation. It emphasizes the importance of thoughtful reward function design and simulation-based training as key enablers for transferring learned behaviors to real-world applications. The overall contribution of this work lies in its demonstration that intelligent robotic behaviors can be efficiently acquired through reward shaping in resource-constrained settings, thus lowering the barrier to entry for academic institutions, startups, and small-scale industries seeking to adopt DRL-driven automation.

However, this work is not without limitations. Further research is required to refine the reward shaping mechanisms and to fine-tune the agent's learning process for improved generalization and adaptability. To enhance the agent's performance, future work should consider several key directions: integrating additional sensor modalities such as tactile sensors to enrich environmental feedback; exploring a broader range of robotic arm tasks to test adaptability; evaluating the agent's capabilities using objects of varying shapes, sizes, and materials; experimenting with alternative deep reinforcement learning algorithms to improve training efficiency and policy robustness; transferring the learned policies from simulation to real-world robotic platforms; and systematically assessing the agent's performance in real-world environments under realistic constraints and uncertainties.

This contributes meaningfully to the democratization of advanced robotics and strengthens the bridge between simulation research and real-world deployment.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the College of Computing and Information Sciences at Caraga State University for providing the academic framework and resources for this research. Special thanks to the faculty advisers and thesis panel members for their technical guidance and critical feedback. This work was partially funded by the Department of Science and Technology - Science Education Institute (DOST-SEI) through a graduate scholarship grant.

REFERENCES

[1] V. Sejdiu, A. Pajaziti, G. Rexha, X. Bajrami, E. Rrustemi, and J. Kola, "Detection, Recognition, and Grasping of Objects through Artificial Intelligence Using a Robotic Hand," IFAC-PapersOnLine, vol. 55, no. 39, pp. 443–446, 2022, doi: 10.1016/j.ifacol.2022.12.077.

[2] G. Revathy, K. Selvakumar, P. Murugapriya, and D. Ravikumar, "Smart manufacturing in Industry 4.0 using computational intelligence," in Artificial Intelligence for Internet of Things, Boca Raton: CRC Press, 2022, pp. 31–48. doi: 10.1201/9781003335801-3.

[3] R. Nian, J. Liu, and B. Huang, "A review On reinforcement learning: Introduction and applications in industrial process control," Comput Chem Eng, vol. 139, p. 106886, Aug. 2020, doi: 10.1016/j.compchemeng.2020.106886.

[4] W. Serrano, "Deep Reinforcement Learning with the Random Neural Network," Eng Appl Artif Intell, vol. 110, p. 104751, Apr. 2022, doi: 10.1016/j.engappai.2022.104751.

[5] C. Symeonidis and N. Nikolaidis, "Simulation environments," in Deep Learning for Robot Perception and Cognition, Elsevier, 2022, pp. 461–490. doi: 10.1016/B978-0-32-385787-1.00023-3.

[6] J. Lima, R. B. Kalbermatter, J. Braun, T. Brito, G. Berger, and P. Costa, "A realistic simulation environment as a teaching aid in educational robotics," in 2022 Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR), and 2022 Workshop on Robotics in Education (WRE), IEEE, Oct. 2022, pp. 430–435. doi: 10.1109/LARS/SBR/WRE56824.2022.9996083.

[7] M. Zahabi and A. M. Abdul Razak, "Adaptive virtual reality-based training: a systematic literature review and framework," Virtual Real, vol. 24, no. 4, pp. 725–752, Dec. 2020, doi: 10.1007/s10055-020-00434-w.

[8] Ö. Özen, K. A. Buetler, and L. Marchal-Crespo, "Towards functional robotic training: motor learning of dynamic tasks is enhanced by haptic rendering but hampered by arm weight support," J Neuroeng Rehabil, vol. 19, no. 1, p. 19, Dec. 2022, doi: 10.1186/s12984-022-00993-w.

[9] B. Maettig and H. Foot, "Approach to improving training of human workers in industrial applications through the use of Intelligence Augmentation and Human-in-the-Loop," in 2020 15th International Conference on Computer Science & Education (ICCSE), IEEE, Aug. 2020, pp. 283–288. doi: 10.1109/ICCSE49874.2020.9201867.

[10] A. Gupta, A. Pacchiano, Y. Zhai, S. M. Kakade, and S. Levine, "Unpacking Reward Shaping: Understanding the Benefits of Reward Engineering on Sample Complexity," Adv Neural Inf Process Syst, vol. 35, Oct. 2022, Accessed: Jun. 24, 2025. [Online]. Available: https://arxiv.org/pdf/2210.09579

[11] R. Devidze, P. Kamalaruban, and A. Singla, "Exploration-Guided Reward Shaping for Reinforcement Learning under Sparse Rewards," 2022. [Online]. Available: https://github.com/machine-teaching-group/neurips2022_exploration-guided-reward-shaping.

[12] H. Ma, K. Sima, T. V. Vo, D. Fu, and T.-Y. Leong, "Reward shaping for reinforcement learning with an assistant reward agent | Proceedings of the 41st International Conference on Machine Learning." Accessed: Jun. 24, 2025. [Online]. Available: https://dl.acm.org/doi/10.5555/3692070.3693450

[13] J. Zhang, Z. Zhang, S. Han, and S. Lü, "Proximal policy optimization via enhanced exploration efficiency," Inf Sci (N Y), vol. 609, pp. 750–765, Sep. 2022, doi: 10.1016/j.ins.2022.07.111.

[14] Q. ZHENG, Z. PENG, P. ZHU, Y. ZHAO, and W. MA, "Robotic arm trajectory tracking method based on improved proximal policy optimization," Proceedings of the Romanian Academy, Series A: Mathematics, Physics, Technical Sciences, Information Science, vol. 24, no. 3, pp. 237–246, Dec. 2023, doi: 10.59277/PRA-SER.A.24.3.05.

[15] Y. Shao, H. Zhou, S. Zhao, X. Fan, and J. Jiang, "A Control Method of Robotic Arm Based on Improved Deep Deterministic Policy Gradient," in 2023 IEEE International Conference on Mechatronics and Automation (ICMA), IEEE, Aug. 2023, pp. 473–478. doi: 10.1109/ICMA57826.2023.10215662.

[16] T. Tiong, I. Saad, K. T. K. Teo, and H. bin Lago, "Deep Reinforcement Learning with Robust Deep Deterministic Policy Gradient," in 2020 2nd International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), IEEE, Nov. 2020, pp. 1–5. doi: 10.1109/ICECIE50279.2020.9309539.

[17] S. Majumder and S. R. Sahoo, "Enhancing 3D Trajectory Tracking of Robotic Manipulator Using Sequential Deep Reinforcement Learning with Disturbance Rejection," in 2024 European Control Conference (ECC), IEEE, Jun. 2024, pp. 2512–2517. doi: 10.23919/ECC64448.2024.10591259.

[18] J. Eschmann, "Reward Function Design in Reinforcement Learning," 2021, pp. 25–33. doi: 10.1007/978-3-030-41188-6_3.

[19] M. Kang and K.-E. Kim, "Analysis of Reward Functions in Deep Reinforcement Learning for Continuous State Space Control," Journal of KIISE, vol. 47, no. 1, pp. 78–87, Jan. 2020, doi: 10.5626/JOK.2020.47.1.78.

[20] T. Shen, X. Liu, Y. Dong, and Y. Yuan, "Energy-Efficient Motion Planning and Control for Robotic Arms via Deep Reinforcement Learning," in 2022 34th Chinese Control and Decision Conference (CCDC), IEEE, Aug. 2022, pp. 5502–5507. doi: 10.1109/CCDC55256.2022.10033563.

[21] S. Yang and Q. Wang, "Robotic Arm Motion Planning with Autonomous Obstacle Avoidance Based on Deep Reinforcement Learning," in 2022 41st Chinese Control Conference (CCC), IEEE, Jul. 2022, pp. 3692–3697. doi: 10.23919/CCC55666.2022.9902722.

[22] W. Tang, C. Cheng, H. Ai, and L. Chen, "Dual-Arm Robot Trajectory Planning Based on Deep Reinforcement Learning under Complex Environment," Micromachines (Basel), vol. 13, no. 4, p. 564, Mar. 2022, doi: 10.3390/mi13040564.

[23] C. Alessi, H. Hauser, A. Lucantonio, and E. Falotico, "Learning a Controller for Soft Robotic Arms and Testing its Generalization to New Observations, Dynamics, and Tasks," 2023 IEEE International Conference on Soft Robotics, RoboSoft 2023, 2023, doi: 10.1109/ROBOSOFT55895.2023.10121988.

[24] S. Sharma et al., "Learning Switching Criteria for Sim2Real Transfer of Robotic Fabric Manipulation Policies," in 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), IEEE, Aug. 2022, pp. 1116–1123. doi: 10.1109/CASE49997.2022.9926556.

[25] R. Liu, F. Nageotte, P. Zanne, M. de Mathelin, and B. Dresp-Langley, "Deep Reinforcement Learning for the Control of Robotic Manipulation: A Focussed Mini-Review," Robotics, vol. 10, no. 1, p. 22, Jan. 2021, doi: 10.3390/robotics10010022.

[26] W. Feng, C. Han, F. Lian, and X. Liu, "A Data-Efficient Training Method for Deep Reinforcement Learning," Electronics (Basel), vol. 11, no. 24, p. 4205, Dec. 2022, doi: 10.3390/electronics11244205.

[27] S. Li, C. Xu, Y. Wang, and L. Xie, "Serial and parallel reliability models for robot arm reliability analysis," J Phys Conf Ser, vol. 1605, no. 1, p. 012043, Aug. 2020, doi: 10.1088/1742-6596/1605/1/012043.

[28] V. R. F. Miranda, A. A. Neto, G. M. Freitas, and L. A. Mozelli, "Generalization in Deep Reinforcement Learning for Robotic Navigation by Reward Shaping," Aug. 2023, doi: 10.1109/TIE.2023.3290244.

[29] S. Singh, F. M. Ramirez, J. Varley, A. Zeng, and V. Sindhwani, "Multiscale Sensor Fusion and Continuous Control with Neural CDEs," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Oct. 2022, pp. 10897–10904. doi: 10.1109/IROS47612.2022.9982210.

[30] P. Yadav, A. Mishra, J. Lee, and S. Kim, "A Survey on Deep Reinforcement Learning-based Approaches for Adaptation and Generalization," Feb. 2022, Accessed: Jun. 24, 2025. [Online]. Available: https://arxiv.org/pdf/2202.08444