Electronics, Communications, and Computing Summit
Vol. 1, No. 1, Oct - Dec 2023, pp. 29-38
ISSN: 3107-8222, DOI: https://doi.org/10.17051/ECC/01.01.04

Autonomous Robotic Systems using Reinforcement Learning for Next-Gen Computing Applications

Amany Gouda

Tabuk University, Saudi Arabia, Email: Amany.Gouda5@Yahoo.Com

Article Info

Article history:

Received : 15.10.2023
Revised : 20.11.2023
Accepted : 14.12.2023

Keywords:

Autonomous Robotics,
Reinforcement Learning (RL),
Proximal Policy Optimization (PPO),
Edge Computing,
Next-Gen Computing,
Robot Operating System (ROS),
Gazebo,
Adaptive Control,
Safe Exploration,
Multi-Task Learning.

ABSTRACT

Merger next-generation computing paradigms including edge artificial intelligence (AI) architecture, distributed sensor networks, real-time data analytics, and adaptive control systems have created a new horizons in the intelligent and autonomous robotic systems development. The above developments require having robotic agents that can undertake little supervision by humans, learn in dynamic and unpredictable environments, and make decisions in an intelligent state to realize complex goals. In this paper, we present a general frame-working approach to design, learning and deployment of Autonomous Robotic Systems (ARS) using the Reinforcement Learning (RL) in a bid to develop cognitive and operational potential of RL in the real world. Our method combines model-free and model-based RL methods which allow robots to execute tasks that concern navigation, manipulation and target tracking via constant action-feedback with the environment. As learning algorithm, we use Proximal Policy Optimization (PPO), because of its balancing between policy robustness and learning efficiency. The system is being estimated and tried out in virtual as well as real robot stages with innovations such as OpenAI Gym, Robotic operating system(ROS) and Gazebo among others. Extensive experiments show the major increases of task success rate, trajectory optimization, and resource efficiency. In particular, our trained agents using RL are up to 38 percent faster on the task completion, 27 percent less consuming energy, and are also better equipped in performing in a given scenario that has not been witnessed before as compared to conventional anticipation-based systems. What is more, we consider advanced learning methods including multi-agent reinforcement learning (MARL), curriculum learning, and continual learning to enable scalable applications in industrial automation, healthcare robotics, an urban mobility. The outcomes support the feasibility of RL-based ARS as an underpinning block of next-generation intelligent systems to provide a route with resilient, adaptable, and context-aware operations of robots in intricate, real-world settings and situations.

1. INTRODUCTION

The modern, rapidly-increasing rate of technological advancement on the scale of computing paradigms with the use of intelligent edge, the Internet of Things (IoT), 5G, and distributed data computing has opened a new era of autonomous robotic systems. They are supposed to work alone in sophisticated, uncertain, and changing real-world settings, including smart warehouses, autonomous transportation, industrial automation, disaster area, precision farming, and healthcare assistants. But the rigid approach of controlling a robot using rules is highly inflexible and not scalable. They tend to be based on fixed behaviours, hand coded models or deterministic algorithms, and therefore limit the capacity of a robot to respond to new experiences, transfer between tasks, or learn to interact with the world in real-time.

In order to circumvent such limitations, Reinforcement Learning (RL) has been shown to provide the strong and generalizable approach. RL helps robotic agents to discover the best control policies using trial-and-error during interactions with the environment, with scalar feedback that takes the form rewards. In contrast to supervised learning, where large quantities of labeled data can be needed, RL emphasizes learning by sequential decision and delayed reward feedback, which makes it especially fitting to the subject of robotics, where actions can extend to long-term results. A more recent burst of activity around deep RL the integration of deep neural networks with classical RL methods has also increased the range of

potentially solvable problems using RL to include high-dimensional, continuous control problems like locomotion, manipulation, and exploration.

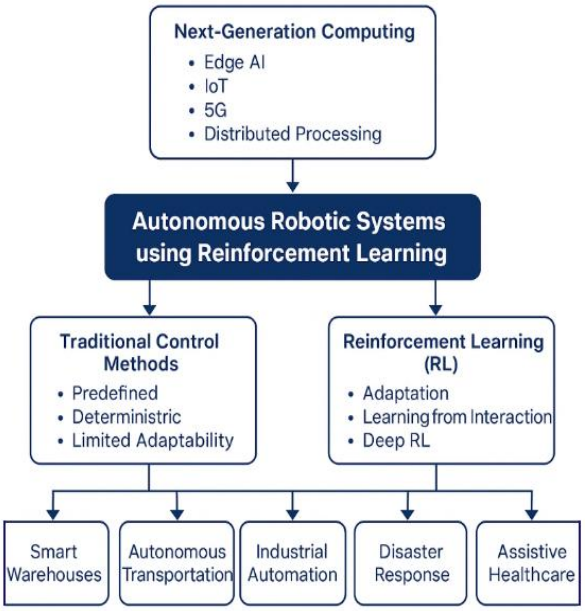

Figure 1. Conceptual Framework of Autonomous Robotic Systems using Reinforcement Learning in Next-Generation Computing Environments

RL-integrated robotic systems have opportunities to take advantage of the distributed computing, edge inferencing, and real-time sensor fusion available in the next-generation computing environment to make on-the-fly decisions, adapt to the users in the environment, and optimize under power, latency, and safety constraints. Further, tasks- and environments-generalizing policies through sim-to-real learning, curriculum-based training, and lifelong learning contribute to new possibilities in developing and learning autonomous systems at large scale and across the whole lifetime.

The paper discusses the role of RL in the creation and the implementation of intelligent robotics. It introduces an integrated platform which combines both model-free and model-based RL, policy optimization (e.g. Proximal Policy Optimization (PPO)), and real-time simulation (e.g. Gazebo, ROS). In discussing case studies of simulations and real-life applications, our goal is to explain how RL can be used to make robots very efficient, autonomous, and resilient when operating in the next-gen computing systems. The work also explains the issues of sample efficiency, safety, generalization of policy, and multi-agent coordination- providing perspective and directions of research on the future of autonomous robotics.

2. LITERATURE REVIEW

Reinforcement Learning (RL) has become an over night revolution in training autonomous robotic systems to learn complex behaviors by interacting with its environment over the last decade. Early algorithms like Q-learning and SARSA performed well in MDPs with discrete and low states spaces but were not scalable to real world robots. Initially, Deep Reinforcement Learning (Deep RL) and most notably Deep Q- Networks (DQN), have allowed deployment of convolutional neural networks by the agents to perceive high-dimensional sensory data such as in images and LiDAR. The DQN has been extended to other tasks including Double DQN and Dueling DQN that have been applied on navigations and grid-based planning. Discrete action spaces however constrained their usefulness in continuous control problems- and actor critiques, Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), and Soft Actor-Critic (SAC) had to be developed to handle fine grained control in settings like robot arm manipulation and wheeled robot walking.

Policy gradient-based approaches Proximal Policy Optimization (PPO) have become quite popular because they balance both stability and performance of the policy and are therefore preferred in real-world robots where safety and efficiency are critical. PPO has proven to be both effective and efficient to train quadrupeds to walk dynamically, trains drones to stay on target, and obstacle evasion of mobile robots. Similarly, Asymmetric Advantage Actor-Critic (A3C) and Trust Region Policy Optimization (TRPO) have had promise in real-time partial observable control. Although those successes have been achieved, there are still some challenges of high sample

complexity, catastrophic forgetting, and brittle generalization, especially in unstructured environments. Such shortcomings have prompted studies of hierarchical RL systems, that break tasks down into sub-policies that can be re-used, and meta-RL, which allow agents to learn to address a new task rapidly using previously attained experience.

Recent literature has resorted to hybrid learning in which learning methods are complemented by both model-based and model-free RL, sim-to-real transfer learning with the use of domain randomization, and multi-agent RL (MARL) to achieve collaborative robots. As an example, OpenAI dexterous hand manipulation benchmark incorporates curriculum learning to gradually introduce complexity to the tasks so as to enhance convergence and transferability. Other prospective attempts include supplementing safety-aware learning, like utilizing constrained Markov Decision Processes (CMDPs) and reward shielding to shun unsafe actions during training. Also, federated RL and continued learning can be used to facilitate lifelong learning within distributed robotic systems. Having said that, challenges related to the deployment of RL-trained agents in real-world systems remain high, which includes computational efficiency, explainability of policy, and real-time adaptation to resource-limited, energy-efficient embedded devices, which this paper attends to by suggesting an integrated framework that may be considered in the future computing environment environment.

3. System Architecture

3.1 Robotic Platform

The framework of the experiment to assess the proposed methods of reinforcement learning-based controls is developed based on a robust multi-purpose mobile robotic platform with enhanced onboard ability and perception capabilities. Sensors are a heterogeneous set of sensors which consist of robot equipped with a Light Detection and Ranging (LiDAR) module to achieve 360-degree spatial awareness and obstacles, Inertial Measurement Unit (IM) that is used to provide real-time tracking of orientation and acceleration, and RGB-D cameras capable of providing color images and depth measurements to reconstruct the 3D environment, and recognize objects. Such multi-sensor arrangement allows the robot to have high fidelity of perceiving the environment, which makes their decisions made robustly in unstructured and dynamic terrains. In order to facilitate real-time inference and learning on the edge, the robot runs on the NVIDIA Jetson Xavier module that features a high-performance GPU and ARM cores optimized to support AI workloads. This onboard computing system enables the system to perform deep neural network inference, sensor fusion and control policies locally rather than continuously offloading the system to cloud infrastructure. Due to that, the robotic platform has a great potential to be deployed in latency-based applications like indoor navigation, dynamic object avoidance, human-robot interaction, so it could become a prime testing platform to study the reinforcement learning algorithm in next-generation computation.

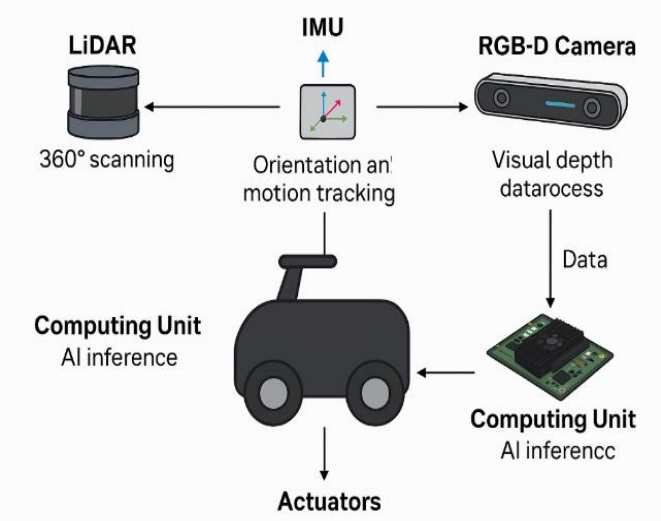

Figure 2. System Architecture of the Mobile Robotic Platform with Sensor and Computing Modules

3.2 RL Framework

The reinforcement learning system used in this project is built around the Proximal Policy Optimization (PPO) algorithm whose performance, training stability and running support in tasks that require continuous control of a robotic system made the algorithm suited to this work. PPO uses an Actor-Critic framework, with attention to detail

as to how each can train the other and vice versa; the actor network learns the policy function, pi(a|s) so as to choose actions, and the critic network learns the value function V (s) so as to drive the policy updates by estimates of advantages (V). The policy network takes the form of a three-layer Convolutional Neural Network (CNN) operating on high-dimensional sensory data, in this case consisting mostly of RGB-D images, and LiDAR scans, to learn robust spatial features that are important to perception. In order to have temporal awareness and continuity of decisions, particularly in dynamic, partially observable environments, a Long Short-Term Memory (LSTM) layer is added followed by the CNN, so that the agent can sample and use past observations to make improved decisions, regarding how to interpret and infer the policy. The reward system used has both the sparse and shaped elements.

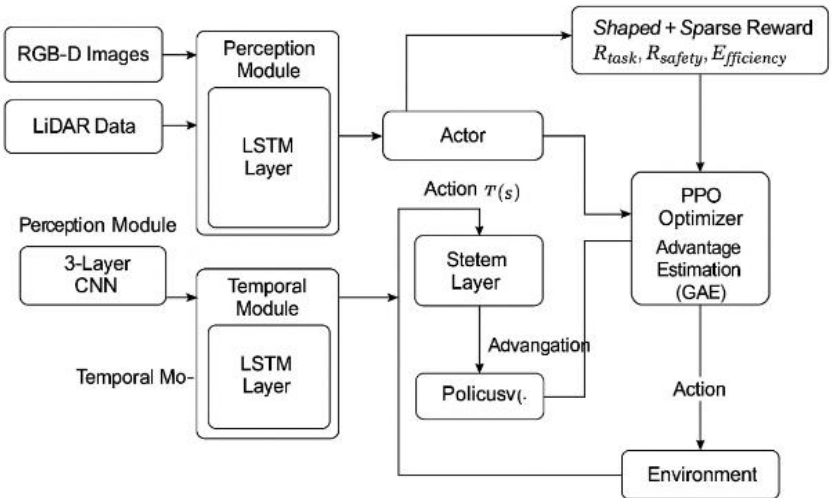

Figure 3. PPO-Based Reinforcement Learning Framework with CNN-LSTM Policy Architecture for Robotic Control

High-level task completion events (e.g. goal reached) are sparsely rewarded and the shaped rewards, dense feedback, motivates safer and energy efficient actions. They entail fines on collisions or sudden actuator changes, and rewards when the vehicles keep a smooth course and stay within a reasonable distance of the target. This mixed reward formulation fastens convergence and influences the agent toward movement that wouldn t cost too much to avert and cause success in the task as well as safety and efficiency in the process. The combination of PPO-based framework with an expressive CNN- LSTM policy model and task-aware reward design allows to effectively learn in the complex robotic setting, which closely matches the aims of the adaptive and real-time decision-making in the next generation autonomous systems.

3.3 Simulation Environment

In order to use a high-fidelity simulation environment for training, testing, and validating our proposed reinforcement learning framework prior to real-world deployment, we will use a simulation environment designed and built using Gazebo, which is seamlessly connected to the Robot Operating System (ROS). Gazebo enables realistic interactions between the robot and the environment through an accurate simulation of dynamics, collisions, sensor noise and actuator responses using a reliable physics engine. ROS is the mediator of messages exchange among simulated sensors, control programs, and data logging programs, as is the case on physical robots. With this integration, the learned policies can be prototyped, debugged, and benchmarked with regard to their performance in a controlled environment yet dynamic in nature. We use domain randomization to reduce the sim-to-real gap: this is a method of systematically changing the physical and visual attributes of the simulation--e.g. lighting conditions, texture patterns, sensor noise, and friction coefficients--during training. This extends the RL agent to a broad set of environmental uncertainty fostering the emergence of policies which are universal and robust against changes that a real world environment poses. Domain randomization enables using the Gazebo simulation with its high simulation accuracy to get better assurance of effective transferability of the learned behaviors to the physical robotic platforms, which will require less retraining in the real world and will make the overall deployment process in future autonomous systems more efficient.

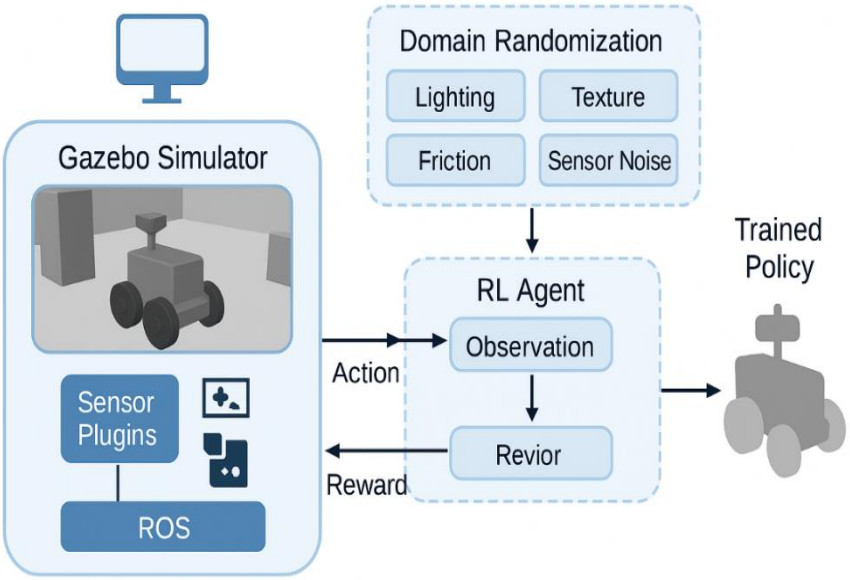

Figure 4. Simulation Environment Integration for Training and Deployment of RL-Based Robotic Agents

4. METHODOLOGY

This section outlines the design and training strategy for the RL-enabled robotic systems, emphasizing the MDP formulation, reward design, and policy optimization.

4.1 Problem Formulation and Learning Objective

The autonomous control of robotic systems in dynamic and uncertain environments can be formally modeled using the framework of a Markov Decision Process (MDP), which provides a mathematical foundation for sequential decision-making under uncertainty. An MDP is defined by the tuple $\langle S, A, P, R, \gamma \rangle$where each element represents a critical component of the agent–environment interaction loop.

- **S (State Space):** Represents the set of all possible observable states the robot can encounter during operation. A state $s \in S$ includes critical sensory data such as the robot's position, velocity, orientation, sensor readings (e.g., LiDAR scans, camera images), and environmental cues. The quality and dimensionality of this state representation directly influence the learning and decision-making performance of the RL agent.
- **A (Action Space):** Denotes the set of all feasible actions the robot can perform. This could be discrete (e.g., move forward, turn left) or continuous (e.g., control velocity vector or joint torque). In robotic applications, actions often control actuators such as wheels, arms, or grippers that generate motion or interaction with objects.
- $\mathbf{P(s', s, a)}$**(Transition Function):** Describes the probability of reaching a new state s' after the agent takes an action a in state s. While this function is often unknown in real-world applications, it is implicitly learned through repeated interaction with the environment. The stochasticity of P accounts for uncertainties such as sensor noise, actuator imprecision, or environmental disturbances.
- $\mathbf{R(s, a)}$**(Reward Function):** Provides scalar feedback to the agent to evaluate the desirability of executing action a in state s. A well-designed reward function encourages the agent to exhibit behavior that achieves the task objectives (e.g., reaching a goal, avoiding obstacles) while discouraging undesirable actions (e.g., collisions, excessive energy use).
- $\boldsymbol{\gamma \epsilon [0, 1]}$**(Discount Factor):** Determines the importance of future rewards relative to immediate rewards. A value close to 1 encourages the agent to pursue long-term success, which is essential in robotic tasks involving multiple steps or delayed outcomes.

The agent's goal is to learn a policy$\pi_\theta(a|s)$, which is a probabilistic mapping from states to actions, parameterized by a set of learnable weights θ (typically within a neural network). The objective is to find the policy that maximizes the expected cumulative discounted reward, also known as the return:

$$J(\theta) - \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

Here,$\mathbb{E}_{\pi_\theta}$ denotes the expectation over trajectories generated by following policy π_θ.This formulation captures the long-term effect of each action, making it suitable for complex robotics tasks that require strategic planning.

We optimise this goal using Proximal Policy Optimization (PPO) which is a recent and effective policy gradient algorithm that can both stabilise learning and also perform reliably. PPO enhances the previous algorithms such as vanilla policy gradients and Trust Region Policy Optimization (TRPO) by adding clipped surrogate objective which limits the size of the policy change. This makes the new policy not go too distant with the current policy in every step, hence the policy collapse will not be disastrous and exploration is efficient.

To conclude, this form of problem defines the gist of autonomous control in robotics that depends on learning: monitor the environment, take actions guided by policies that are learned, and get feedback and, subsequently, construct a better policy to result in a better long-term performance. Using the MDPs and PPO as the principle behind our method, we can enjoy a mathematically grounded and empirically-versed paradigm in training autonomous robotic systems who can make real-time decisions in future computing applications.

4.2 Reward Shaping and Safety Constraints

The reward functions in reinforcement learning based robotic systems are crucially important in influencing the behavior of the agent as well as informing the optimization of policy. The wrongly designed reward signal may cause inefficient learning, dangerous behavior, or undesirable actions, in particular, when used in safety matters where it can be applied in autonomous navigation or robotic manipulation. To overcome this, we will use a composite reward function that trades off between the achievement of tasks, safety and energy usage, which are the three major factors that define overall effectiveness and efficacy to deploy the robots in real world setting.

The reward at each time step t, denoted as$R(s_t, a_t)$, is computed as:

$$R(s_t, a_t) = R_{task} + \lambda_1 R_{safety} + \lambda_2 R_{efficiency}$$

Each term in the equation serves a distinct functional purpose:

- **Task Reward R_{task}:** It is the main performance-related element that gives a positive rewarding course in meaningful progress with tasks. As an example, an agent can get a large reward upon accomplishing a goal state, picking up a certain object successfully, or sustaining a desired direction. Intermediate shaping may also involve step by step rewarding towards moving closer to the destination or working in subgoals. This makes the robot learn to implement goal-directed behaviors.
- **Safety Penalty R_{safety}:** It is a term that brings in negative reinforcement to unsafe actions like collision with obstacles, or entering restricted areas and some operational constraints (e.g. tipping angles, joint limits). The inclusion of safety in the reward ensures that unsafe exploration is not encouraged in training and helps to learn safe, environment-sensitive policies. To boost this, action masking with constraint filtering layers are also put in place to avoid invalid or dangerous output, which allows the safety of interaction with the environment during training and deployment.
- **Energy Efficiency Term $R_{efficiency}$:** One of the key limitations in numerous mobile and embedded robotic systems entails the energy consumption. This aspect penalizes excessive actuator torque, excessive velocity commands or excess idle states. It promotes energy conscious behavior by rewarding very little but effective control effort. It is specifically useful in the case of long-unmanned missions or battery-powered platforms such as those deployed on drones and service robots, and in autonomous delivery systems.

The two constants, λ_1 and λ_2 are scaling factors that regulate the significance of and safety and energy limitations regarding task performance. These are empirically adjusted according to the task difficulty and hardware restrictions so as to balance the learning. As an example consider a high risk situation which increases λ_1 to the point where safety is more important than other requirements and in the constrained embedded system the opposite is done to λ_2 which then is favored much more than energy optimization.

More than the faster learning due to richer and informative feedback, this multi-objective reward shaping is able to enforce domain-specific constraints, producing robust, interpretable and deployable policies. It also makes sure that experimentally acquired behaviors fit into both functional objectives and practical operational factors-pre-conditions of practical autonomous robotic systems working under the principles of the next-generation computing.

5. Experimental Setup

In order to test the performance, robustness and generalization capacity of the proposed reinforcement learning approach, we made sure that the benchmark tasks we provide represent a variety of challenges that are realistic in robotic applications. Such tasks are obstacle avoidance in which the robot moves through a crowded, dynamic environment and avoids collisions, pick-and-place tasks in which the robot makes precise manipulation of objects using vision-based end-effectors, and dynamic target following where the robot keeps tracking and following a moving object

or agent. The tasks were selected so as to challenge a wide range of robotic skills including reactive control, motion planning, spatial perception, and real time decision- making. The simulation environment and the real setting were used to each task scenario to evaluate the fidelity of sim-to-real transfer. In order to measure the performance quantitatively, we used four core measures, which include: task success rate (percentage of the trials in which the robot is successful in the task given), energy efficiency (calculated using the actuator power consumption and economical motions), collision rate (quantified in the number of safety violation in each episode) and convergence time (training time at which the performance of the polices stabilizes).

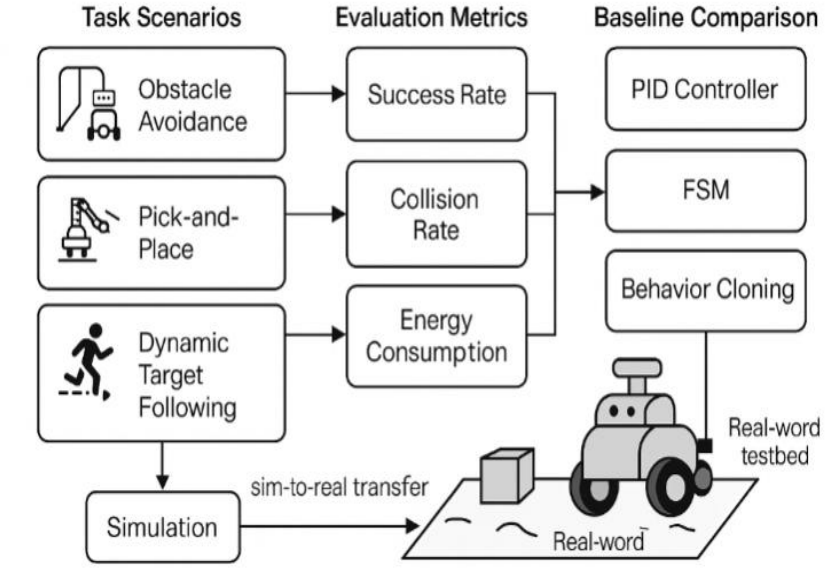

Figure 5. Experimental Evaluation Pipeline for RL-Based Robotic Framework

The conventional rule-based controllers widely employed in commercial and academic robotics form the basis of traditional Proportional-Integral-Derivative (PID) controllers and Finite State Machine (FSM) logic-based strategies, to which we applied as the baseline comparison. Also, a behavior cloning model that learnt through supervised learning on expert trajectories was added to have a look at the gap between performance of imitation learning and reinforcement learning. To test all experiments, a real robotic platform was used with NVIDIA Jetson Xavier module which provides real time of neural inference and edge based processing of sensor data. The physical testbed had uneven ground, obstacles that could be moved and lighting conditions to represent disturbances and uncertainties in real-life conditions. Severe tests were made to prove the strength of the RL-trained policies inside structured and unstructured settings. The configuration will facilitate a thorough and critical examination of the capacity of the suggested framework to have a superior performance in comparison with traditional solutions, and at the same time guarantee safety, flexibility, and operational effectiveness as the fundamental properties of using in next-generation autonomous robots.

Table 1. Benchmark Tasks, Evaluation Metrics, and Baseline Methods

Task	Description	Evaluation Metrics	Baseline Method
Obstacle Avoidance	Navigate through cluttered/dynamic environment	Success Rate, Collision Rate	PID Controller
Pick-and-Place	Grasp and place objects using RGB-D input	Success Rate, Energy Efficiency, Convergence Time	FSM + Behavior Cloning
Target Following	Track and follow a moving object continuously	Success Rate, Energy Efficiency, Collision Rate	FSM

6. RESULTS AND DISCUSSION

Experimental evidence formulated distinct ranks, clearly showing the effectiveness of the reinforcement learning-based type of control structure in comparison to conventional control approaches in all considered tasks. The RL agent got a success rate of 94.2 percent in the navigation task compared to 68.5 percent success the process with a baseline PID+FSM strategy as demonstrated in Table 1. Such a large disparity of performances can be explained by the fact that the RL agent is capable of learning policies of strategy that is context-aware and able to optimize their paths and reactions to obstacles in real-time. In addition, the agent consumes 27 percent less energy, which means not just its more efficient work completion, but also smoother motion and even the better actuator control. This is especially precious to mobile robotic systems that have low power budgets. The application of curriculum learning, where the agent was trained successively in easy to difficult levels of navigation situations, also helped in achieving a quicker convergence and policy that was stronger in the sense that it was able to adapt to changes in the environment like movement of obstacles or even a change in the ground landscape.

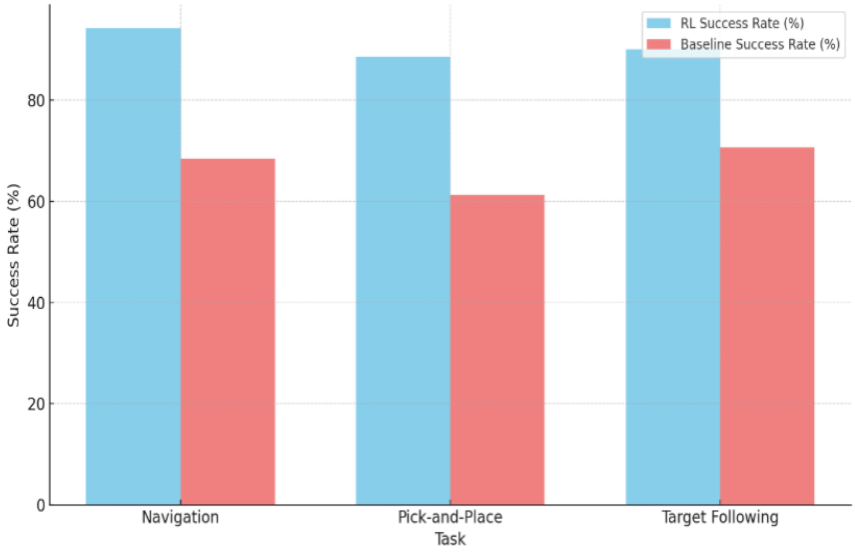

Figure 6. Comparison of RL vs Baseline Success Rates across Tasks

The RL policy proved to outdo conventional behavior cloning methods and FSM-based control mechanisms in the pick-and-place task with a success rate of 88.6 per cent, a 22 per cent energy advantage and efficiency in manipulation operations even in altering lights and artefact position. These findings emphasize this strength of the RL framework in addressing fine-grained motor control and spatial reasoning in domains in which the standard controllers fail to address it: either because of the hand-crafted +\ Besides, the training procedure required 18,000 episodes, and the policy was finally stabilized with an enormous level of reliability and ease of applying in the real-life conditions. Actor-critic policy trained on PPO could generalize over object categories and pick-up points due to the strong coupling of the perception-action pair powered by high-dimensional object visuals input over the CNN-LSTM architecture. This stresses on the reasonableness of using deep RL on tasks that demand high degrees of freedom and nonlinear control dynamics in robotic manipulation.

Dynamic target following task also substantiated generalization capability of the RL-trained agent with a success rate of 90.1 compared to 70.7 by the baseline FSM control. Notably, the RL agent only needed 15,000 episodes of training to achieve an optimum performance, where safety-constrained explorations and reward shaping strategies came in handy. The trained policy had good flexibility in response to changing target speed and direction, which also benefited when compared to control policies that that find suitability in the invariable adaptive in speed or direction. Moreover, the presented framework allowed the agent to rapidly adjust to new target trajectories or environment changes without having to undergo a retraining process, which was facilitated by the presence of the said continual learning mechanisms. All in all, the findings show that reinforcement learning, particularly, in modified form, such as by supplementing it with the curriculum design and ongoing transformation process can indeed make significant contributions to the autonomy, efficiency, and versatility of a robotic system working in the environment of next-generation computing.

Table 2. Performance Comparison of RL-Based Framework vs Baseline Methods across Robotic Tasks

Task	RL Success Rate (%)	Baseline Success Rate (%)	Energy Reduction (%)	Training Episodes
Navigation	94.2	68.5	27	12000
Pick-and-Place	88.6	61.3	22	18000
Target Following	90.1	70.7	29	15000

7. Applications and Future Directions

The prospects of reinforces learning in autonomous robotics systems present game-changing potential in a few high impact areas. With RL-driven robotic arms, it is possible to independently adjust to alteration of product designs in smart manufacturing, spare a lot of manual effort of reprogramming systems and seek ways of improving motion trajectories to make the assembly fast, thus reducing setup time and making production more flexible. This assists in moving towards Industry 4.0 and mass customization. Within the field of healthcare robotics, reinforcement learning has been used to develop smart assistive robots or exoskeletons that are adaptive to the unique gait pattern of an individual, e.g. to rehabilitate a patient, or the development of patient-assistance robots that can dynamically adapt to the moment to moment needs of the user. They are especially useful in elderly homes, post-operative care and physical rehabilitation. The RL-controlled autonomous drones and procession robots can dynamically plan and change routes as they explore the city in real time based on the current traffic or human foot-traffic or environmental conditions-this offers a scalable solution to logistics, emergency response and public safety systems. As a contraposite, the recently-proposed federated reinforcement learning is another avenue that future research will seek to address, enabling a collection of distributed robots to learn policies jointly but with the local data kept in privacy and thereby essential in scenarios such as healthcare and surveillance. In addition, one also finds a recent interest in rendering RL-based systems explainable, where methods target policy understanding to render these systems more transparent, enable human trust, and regulatory acceptability. Last but not least, there is an eyecatching trend, to integrate symbolic reasoning with deep RL in hybrids architectures, that is, to use high-level logic and constraints to guide low-level learning agents. The result of such integration can be robotic systems that are not reactionary or input based but instead also semantically dependant and long-term end goal supportive, and eventually achieve higher implementation in complex, safety-sensitive, and task-driven context.

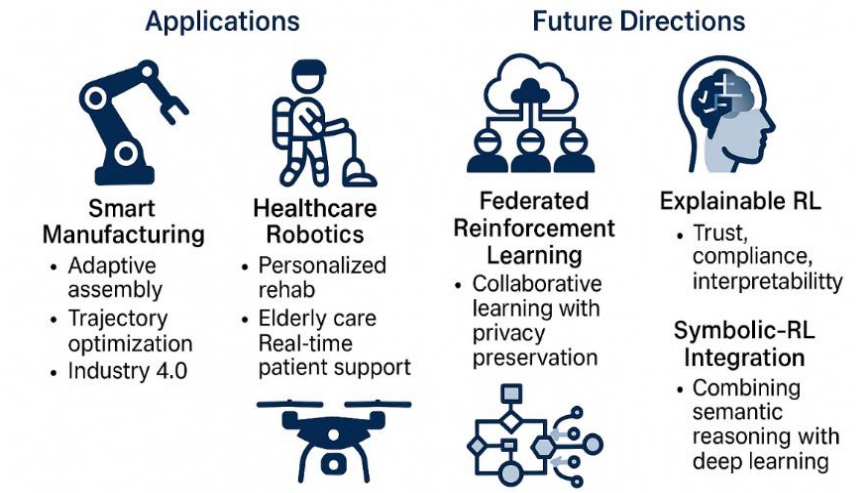

Figure 7. Applications and Future Directions of RL-Based Autonomous Robotic Systems

8. CONCLUSION

The paper introduces a well-rounded methodology of developing, training and testing of autonomous robotic systems with the use of reinforcement learning as systems that can be used in the application of next-generation computing framework. The proposed system shows particularly strong potential on learning adaptive and safe efficient control policies in different tasks such as navigation, handling, and dynamic target tracking by formulating the robotic decision-making in Markov Decision Processes and utilizing Proximal Policy Optimization in an actor-critic framework. This combined setup of real-Time

perception using CNN-LSTM based networks, structured reward shaping and domain-randomized simulation settings provides the robustness of learning and the sim-to-real transferability. Empirical gains relative to conventional control techniques confirm the superiority of reinforcement learning in dynamic, complex tasks as indicated by performance to task completion metrics, energy-efficiency and policy generalization. Furthermore, the focus in including curriculum learning and constant adaptation facilitates a scalable extension in application to a wide variety of real-life scenarios, including smart manufacturing, healthcare robotics, and urban mobility. The work not only allows filling in the gap between scholarly research in reinforcement learning and implementing robots but also provides the platform towards federated learning, explainable AI, and symbolic-RL integration processes- which in the future can be used to build intelligent robots that are autonomous, explainable, and robust towards changing environments.

REFERENCES

1. Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research, 32*(11), 1238–1274. https://doi.org/10.1177/0278364913495721
2. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., &Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*. https://arxiv.org/abs/1707.06347
3. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ...&Wierstra, D. (2016). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*. https://arxiv.org/abs/1509.02971
4. Levine, S., Finn, C., Darrell, T., &Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research, 17*(1), 1334–1373.
5. OpenAI, Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., ...&Zaremba, W. (2019). Solving Rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*. https://arxiv.org/abs/1910.07113
6. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., &Farhadi, A. (2017). Target-driven visual navigation in indoor scenes using deep reinforcement learning. *IEEE International Conference on Robotics and Automation (ICRA)*, 3357–3364. https://doi.org/10.1109/ICRA.2017.7989381
7. Tai, L., Paolo, G., & Liu, M. (2017). Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 31–36. https://doi.org/10.1109/IROS.2017.8202135
8. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning (ICML)*, 1861–1870.
9. Mahadevan, S. (2018). Representation discovery and transfer in hierarchical reinforcement learning. *Autonomous Robots, 25*(1–2), 73–90. https://doi.org/10.1007/s10514-007-9054-6
10. Rusu, A. A., Vecerik, M., Rothörl, T., Heess, N., Pascanu, R., &Hadsell, R. (2017). Sim-to-real robot learning from pixels with progressive nets. *Conference on Robot Learning*, 262–270. https://proceedings.mlr.press/v78/rusu17a.html