Simulation-Based Reinforcement Learning for Autonomous Object Manipulation with a Robotic Arm

Cristian Santiago López Cadena[1] 20222020027[1]
Juan Diego Lozada Gonzalez[2] 2022020014[2]

Universidad Distrital Francisco José de Caldas

***Abstract*—Some production processes suffer from inefficiencies due to the high demands placed on certain steps, which ultimately reduce overall profitability. To address this issue, we propose the implementation of an autonomous robotic arm to optimize these processes and handle multiple products efficiently. Our approach involves training the robotic arm using reinforcement learning, enabling it to make autonomous decisions and adapt to varying production scenarios. This solution aims to improve operational efficiency and increase productivity in industrial environments.**

***Index Terms*—Reinforcement Learning, Machine Learning, Cybernetic, Sensors, Vision Computer**

I. INTRODUCTION

In many industries, robotic systems are used to automate simple and repetitive tasks. These robots typically follow predefined sequences of instructions and perform reliably in controlled environments. However, their performance degrades significantly when the environment changes when objects are moved, rotated, or presented in varying shapes and positions.

One of the challenges in robotic automation is enabling a robot arm to detect, grasp, and manipulate objects autonomously. Traditional systems lack the flexibility to adapt to dynamic, real conditions. For instance, if an object appears in a new orientation or location, the robot may fail to grasp it correctly without manual reprogramming.

To address this limitation, recent advancements have incorporated machine learning (ML) and artificial intelligence (AI), particularly reinforcement learning (RL). Unlike conventional programming, ML allows systems to learn from data and improve their behavior over time without explicit human intervention.

Reinforcement learning, in particular, is highly suitable for robotics. In this paradigm, an agent interacts with its environment, receives feedback in the form of rewards, and incrementally learns the best strategy to achieve its goals. Since RL does not rely on labeled data, training can be conducted in simulated environments, significantly reducing the need for physical experimentation and lowering development costs.

This simulation based training approach makes robotic systems more scalable and cost-effective for industrial applications. It requires a well-modeled understanding of the system dynamics but eliminates the need for extensive manual programming or constant supervision.

Research has shown promising results using deep reinforcement learning for robotic manipulation tasks. These systems benefit from continuous feedback and can scale over time as the learned policies improve and adapt to new scenarios.

In this project, we propose a robotic system that combines deep reinforcement learning with a simulated training environment to enable a robotic arm to autonomously detect, grasp, and move objects based on visual input. The system uses the Soft Actor-Critic (SAC) algorithm, which is well-suited for continuous control tasks, and is trained in simulation before being deployed in the real world.

II. METHODS AND MATERIALS

A. Methods

To solve the problem of object manipulation using a robotic arm, we propose a method that combines reinforcement learning with simulation-based training. The system is developed and tested in a virtual environment using Gymnasium Robotics, which allows for safe and efficient experimentation before deploying to the real world.

The robot interacts with the environment by receiving state information such as the position and orientation of the object and the arm from the simulation. Based on this input, the reinforcement learning agent learns how to control the robotic arm to reach, grasp, and move the object to a designated location, as we have in the casual loops

This approach avoids the need for image-based perception, simplifying the training process and focusing on learning optimal actions from structured state data. By training entirely in simulation, we reduce the cost and complexity associated with physical experiments while enabling the system to generalize to real-world tasks.

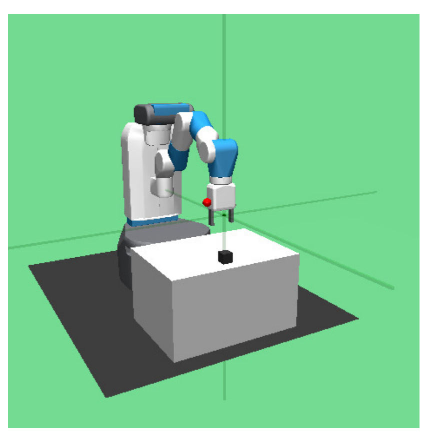


Fig. 1 | Gymnasium Robotic

To train the robot decision process, we use a reinforcement learning algorithm called Soft Actor-Critic (SAC). This algorithm is chosen because it works well with continuous actions, such as the smooth movement of a robotic arm. The robot receives a reward signal based on how successful its actions are—for example, positive reward for picking up the object correctly and negative reward if it drops or misses it. Over many training episodes, the robot learns a better policy to maximize these rewards. This a image about how is the flow of SAC

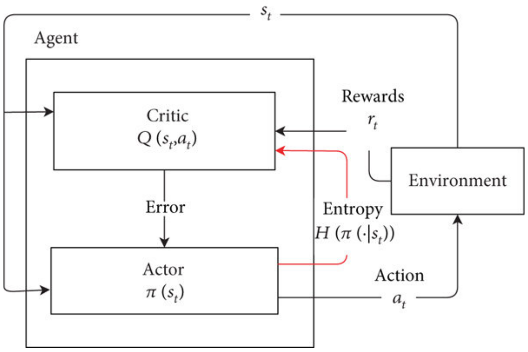


Fig. 2 | Soft Actor Critic

The next Link serve to see all the design system we implement for the project, with its inputs, moduls and outputs Project Demonstration Video

B. Materials

The system was developed entirely using simulation tools to avoid the cost, complexity, and limitations of real-world experimentation during the initial stages. All implementation was carried out in Python, which offers a robust ecosystem for scientific computing and is widely adopted in the fields of machine learning and robotics.

For the simulation environment, we used Gymnasium Robotics, a toolkit built on top of OpenAI Gym. This platform provides standardized tasks for robotic manipulation and allows direct access to key environment variables such as joint angles, object positions, and gripper status. By relying on these structured observations rather than raw visual input, we simplify the learning problem and improve training efficiency. The environment also ensures safety and repeatability during experimentation, which is essential for reinforcement learning workflows that require extensive interaction between the agent and its environment.

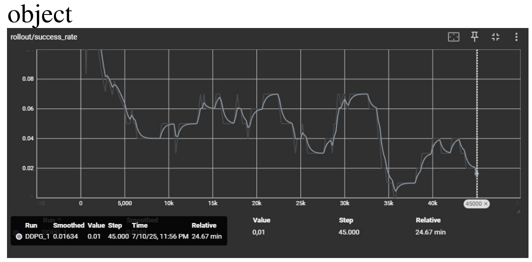
The control policy of the robotic arm was trained using the Soft Actor-Critic (SAC) algorithm. SAC is a model-free, off-policy reinforcement learning method designed for continuous action spaces. Its ability to balance reward maximization with entropy encourages the exploration of diverse strategies, making it particularly suitable for robotic control tasks that require precision and adaptability. During training, the agent receives continuous feedback from the environment and learns to optimize its actions based on observed state transitions and associated rewards.

The input to the learning algorithm is composed entirely of low-dimensional numerical state vectors provided by the simulator. These include information about the position and orientation of both the robotic arm and the target object, as well as relevant joint and gripper states. No images, depth data, or other forms of visual input were used in this project.

All training was conducted on a standard personal computer equipped with a modern CPU and GPU. The use of simulation not only eliminates the need for physical sensors or robotic hardware but also allows rapid prototyping and iterative improvement of the system in a controlled and replicable environment.

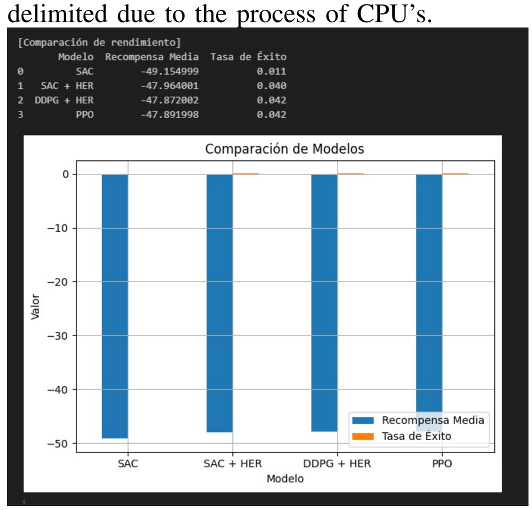
III. RESULTS

The algorithm who was the best according the acurracy and precision the robot had after the complete training was DDPG + HER due to it reach a major sucess rate, now, comparing the beginning of the robot and after the training we could check the big difference analyzing and grasping the object



Graph SAC + HER

Some could think that thousand of steps is a big number for a test, but in fact, it is a small proof about all data need a program to get a accuracy learning, our computer were a delimited due to the process of CPU's.


[Comparación de rendimiento]

	Modelo	Recompensa Media	Tasa de Éxito
0	SAC	-49.154999	0.011
1	SAC + HER	-47.964001	0.040
2	DDPG + HER	-47.872002	0.042
3	PPO	-47.891998	0.042

Comparative

IV. CONCLUSIONS

- The use of reinforcement learning, particularly the Soft Actor-Critic (SAC) algorithm, enables the robotic arm to autonomously learn and adapt to object manipulation tasks without relying on predefined instructions. This significantly improves the robot's ability to handle dynamic and unpredictable environments.

- Training entirely in a simulated environment, such as Gymnasium Robotics, proved to be a cost-effective and safe method. It eliminates the need for physical hardware in early development stages, allowing for rapid experimentation and scalable learning without the risk of damaging real-world equipment.
- Integrating Hindsight Experience Replay (HER) with SAC slightly improved the robot's success rate in a sparse-reward setting, suggesting that HER enhances learning by allowing the agent to learn from past failures and reinterpret them as successful experiences under different goals.
- The decision to rely on structured state vectors instead of visual inputs like images or depth data simplified the training process and improved computational efficiency. This approach proved sufficient for achieving meaningful robotic control in the manipulation task.
- The limit of resources in our devices were significantly crucial in the process due to the time who take process a thousand of steps were insignificant compare the millions we needed. We tried with GPU but we found some problems of compatibility

REFERENCES

[1] M. Matl, et al., "Grasping with vision: A survey," Robotics Journal, 2020.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[4] V. Mnih, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv preprint arXiv:1707.06347, 2017.

[6] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic Algorithms for Continuous Control," in *Proc. Int. Conf. on Machine Learning (ICML)*, 2018.

[7] I. Kalashnikov, et al., "QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation," arXiv preprint arXiv:1806.10293, 2018.

[8] OpenAI, "Gym: A toolkit for developing and comparing reinforcement learning algorithms," arXiv preprint arXiv:1606.01540, 2016.

[9] M. Abadi, et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," in *Proc. OSDI*, 2016.