Autonomus Robot for the Detection of Landmines in a Simulated Environment

Jhojan Stiven Aragon Ramirez
Dept. of Computer Engineering
Universidad Distrital Francisco José de Caldas
Email: jhsaragonr@udistrital.edu.co

Kevin Emmanuel Tovar Lizarazo
Fundamentos de ciencias y sistemas
Universidad Distrital Francisco José de Caldas
Email: ketovarl@udistrital.edu.co

I. Abstract

II. Introduction

En este trabajo, se presenta un agente de aprendizaje por refuerzo profundo (DRL) diseñado para la detección de minas terrestres en un entorno simulado. La detección de minas es un desafío crítico en la ingeniería y la seguridad, ya que estas representan una amenaza significativa para la vida humana y la infraestructura. La combinación de tecnologías avanzadas, como el aprendizaje a utomático y los sensores LiDAR, ofrece una solución prometedora para abordar este problema.

To develop and test our deep reinforcement learning (DRL) agent, we use the Gymnasium library, specifically the CarRacing environment [1]. This library provides a simulated environment that is both flexible and widely used in the research community. The CarRacing environment allows us to create realistic scenarios for training and evaluating our agent. By using this tool, we can focus on improving the agent's performance without the need for a physical setup, saving time and resources.

We rely on the Gymnasium CarRacing environment [1], which uses the Box2D library to provide all vehicle dynamics out of the box.

Box2D solves the planar rigid-body equations

$$m\dot{v} = \sum F, \quad I\dot{\omega} = \sum \tau$$

and handles wheel chassis joints, tire friction and steering internally.

By using this built-in model, we avoid manual implementation of physics and focus on training our DRL agent efficiently.

Recent advancements in reinforcement learning have demonstrated the potential of feedback linearization techniques for controlling autonomous vehicles in complex environments. For instance, [2] explores the use of reinforcement learning to design a linearizing controller for car racing dynamics, enabling efficient path planning and trajectory tracking. Inspired by these methods, our work adapts similar principles to address the unique challenges of landmine detection in simulated environments.

III. Methods and Materials

We began by drawing a component diagram to show how each part of our system fits together (see Fig. 1). This diagram clarifies the relationships between our sensor inputs, the learning agent, and the robot's control outputs before describing the technical details.

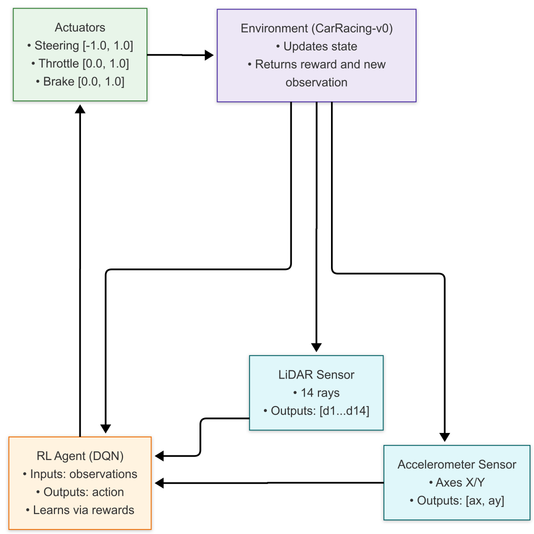

Fig. 1. Component diagram of the system.

Our setup uses the CarRacing-v0 environment from Gymnasium, which provides a two-dimensional, top-down view of a procedurally generated racetrack. At each time step, the agent receives a single 96×96 RGB image showing the car at the center of the track. In addition to this visual input, Gymnasium exposes four ABS sensor readings (one for each wheel), true speed, steering position, and gyroscope measurements. We combine these data into a single state representation that the agent observes.

In addition to the visual and sensor data provided by the CarRacing-v0 environment, the initial problem formulation also considered the use of LiDAR sensors. We aim to explore the integration of LiDAR, potentially by modifying the existing environment or interfacing with a simulated LiDAR. This additional sensory input can be seamlessly incorporated into the state representation fed to our DQN agent, further enriching its perception of the surroundings.

To navigate and avoid obstacles, we implemented a Deep Q-Network (DQN) agent. The DQN uses a convolutional neural network to approximate the action-value function over continuous control actions—steering, gas, and brake. During train-

Funded by *Your Funding Source* under grant No. 12345.

ing, the agent's experiences (state, action, reward, next state) are stored in a replay buffer. We periodically sample mini-batches from this buffer to update the main network, while a separate target network—updated every few episodes—helps stabilize learning.

The reward function, provided by the Gymnasium CarRacing-v0 environment [1], is designed to encourage both speed and track coverage. At each frame, the agent receives a -0.1 penalty, which pushes it to finish the track quickly. Each newly visited track tile yields a bonus of $1000/N$ points, where N is the total number of tiles. For example, covering all tiles in 732 frames yields:

$$1000 - 0.1 \times 732 \approx 926.8 \text{ points.}$$

Crashing off-track for too long or failing to visit new tiles ends the episode.

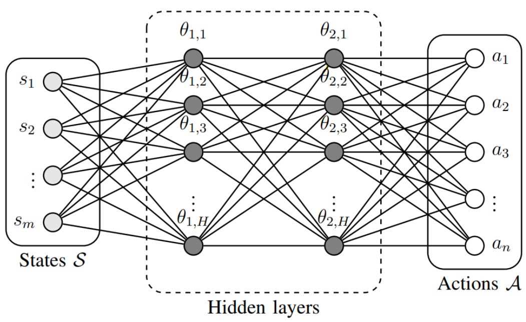

Fig. 2. DQN algorithm diagram [3].

Figure 2 illustrates the overall DQN workflow. In our implementation, the input image and sensor readings pass through convolutional and fully connected layers to produce Q-values for each action. The action with the highest Q-value is selected at each step. Over thousands of episodes, the agent learns to balance acceleration, braking, and steering to maximize cumulative reward while avoiding track boundaries.

By combining visual information with sensor data and training with the DQN algorithm in the 2-D CarRacing-v0 environment, our system learns to navigate complex tracks efficiently and safely. This approach lays the foundation for future extensions, such as adding more sophisticated perception modules or testing on physical robots.

IV. Results

V. Conclusions

Acknowledgment

VI. bibliography

References

[1] F. Foundation, "Gymnasium carracing environment," 2023. [Online]. Available: https://gymnasium.farama.org/environments/box2d/car_racing/
[2] S. Li, M. Estrada, and X. Cai, "Feedback linearization of car dynamics for racing via reinforcement learning," 2021, final research paper for Berkeley's CS 285 (Deep Reinforcement Learning) in Fall 2020. [Online]. Available: https://arxiv.org/abs/2110.10441
[3] F. Mismar, J. Choi, and B. Evans, "A framework for automated cellular network tuning with reinforcement learning," 08 2018.