Multi-Agent Reinforcement Learning: A Hardware Perspective

Atharva Sharma, Rajdeep Niyogi

Department of Computer Science and Engineering, IIT Roorkee

1 Abstract

This project demonstrates the implementation of Multi-Agent Reinforcement Learning (MARL) on a custom-designed embedded robotic platform. A team of ground robots collaboratively learns to make decisions using a distributed reinforcement learning approach. Each robot is equipped with a microcontroller, motor drivers, sensors, and communication modules to navigate its environment and respond to learned behavior.

2 Introduction

Multi-Agent Reinforcement Learning (MARL) is a powerful paradigm that enables teams of agents—such as ground robots—to learn cooperative behaviors in a decentralized manner. Unlike centralized systems, MARL relies on local observations and interactions, improving robustness and scalability. This approach is particularly relevant in real-world scenarios like search and rescue, agricultural monitoring, and warehouse automation, where centralized control may fail due to communication delays or dynamic environments.

This work explores how simple ground robots can learn and adapt within a hybrid MARL system. Due to the Arduino Mega 2560's computational limitations—limited memory, low processing power, and lack of floating-point support—the reinforcement learning model is executed on a central laptop. However, all sensing, actuation, and inter-robot communication occur locally on each robot, making the system modular and scalable.

Robots use ultrasonic sensors (to be integrated post-demo) for environmental awareness and communicate via HC-05 Bluetooth modules. Although peer-to-peer communication is possible, all messages in this implementation are routed through the laptop, which acts as a relay—not a global controller—preserving the decentralized nature of decision-making.

This hybrid architecture enables agents to learn distributed control policies that adapt to dynamic environments and each other's behavior without requiring centralized decision-making or offline training. The project demonstrates the feasibility of real-world MARL deployment and sets the stage for future work involving more complex behaviors, additional sensors like vision, and embedded on-board learning.

3 System Design and Hardware

Each robot is built using low-cost, readily available components:

- **Arduino Mega 2560**: Main controller for sensor processing and control.
- **L298N Motor Driver**: Enables bidirectional motor control via PWM.
- **Bluetooth HC-05 Module**: Facilitates peer-to-peer agent communication.
- **Sensors**: HC-SR04 ultrasonic sensor, IR proximity sensors, MPU6050 IMU, and A3144 Hall effect sensor.
- **Custom Wheel Encoders**: Two types—IR-reflective and magnet-based—for velocity and displacement feedback.

The platform implements a Multi-Agent Reinforcement Learning (MARL) framework using decentralized robots controlled via the Arduino Mega. Each robot perceives its environment and communicates with others using onboard sensors and Bluetooth modules. Motion is achieved through differential drive with L298N drivers. The chassis is made from plywood, with four driven wheels, with two having encoders.

Sensor data informs local decision-making, contributing to each agent's state and reward. The MPU6050 provides angular velocity (yaw), enabling heading control via a dedicated control loop. Robots are powered by onboard rechargeable batteries for untethered operation.

Encoder Design:

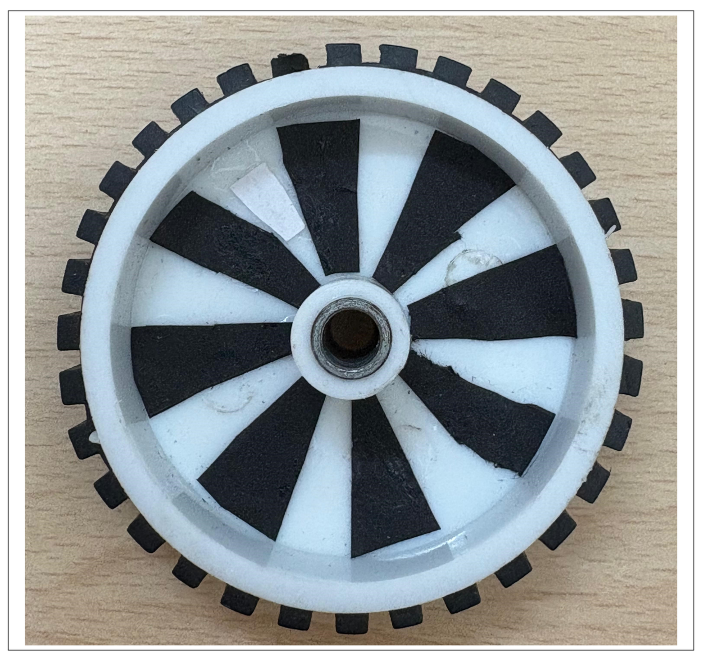

Figure 1: IR-reflective encoder design

Figure 2: Hall sensor-based magnetic encoder

Two encoder systems were developed:

- The IR encoder counts transitions between black and white stripes using a reflectance sensor.
- The magnetic version uses small Nd magnets and A3144 Hall sensors to detect pulses from south poles.

Both types measure wheel velocity and displacement via interrupt-based pulse counting. The magnetic encoders are now preferred due to lower noise and better reliability—optical encoders were sensitive to ambient lighting and wheel misalignment.

Control System Architecture:

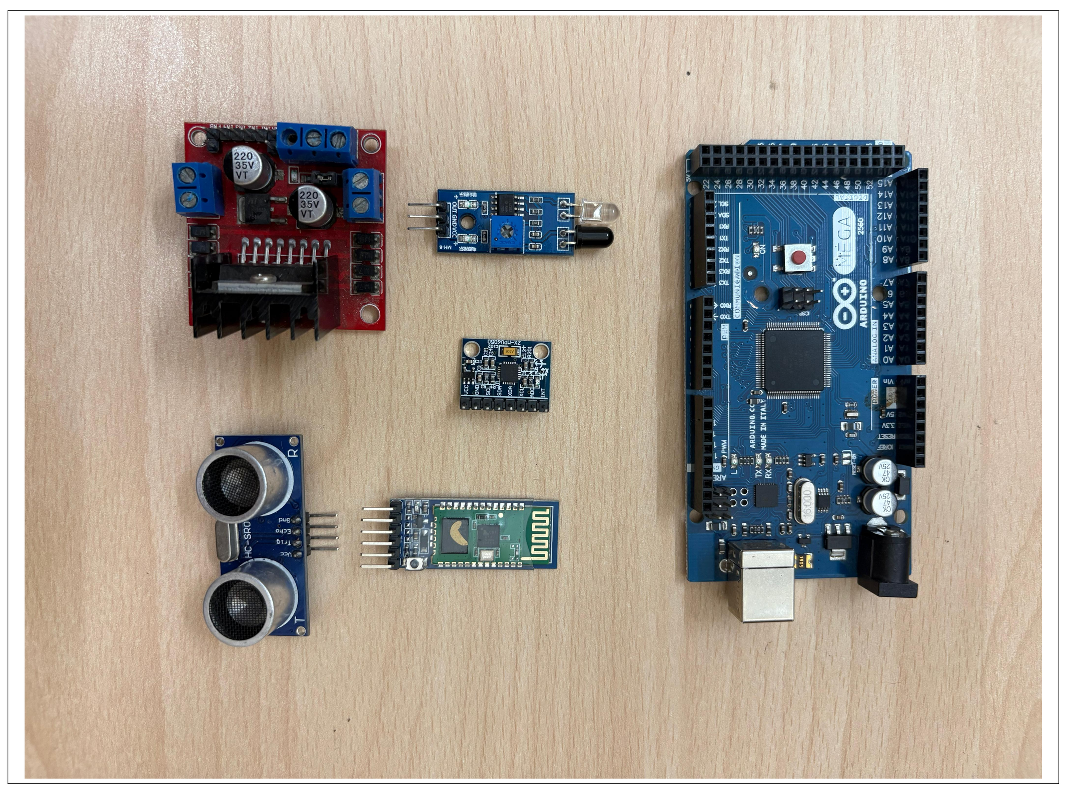

Figure 3: Arduino-based cascaded control logic

Each robot executes onboard control using cascaded PID loops:

- **Position control**: An outer loop computes target wheel RPMs based on desired displacement.
- **Velocity control**: An inner loop stabilizes actual wheel speed via PWM.
- **Directional control**: A similar cascaded loop uses the MPU6050's yaw rate to adjust heading with feedback.

This architecture enables smooth movement and accurate turns without centralized supervision. Communication is limited to basic signals (like intent or location) to simulate decentralized cooperation. Collision detection and obstacle avoidance are under development and will be integrated post-demo.

The integrated design supports autonomous navigation in structured, multi-agent settings such as indoor mapping or field exploration.

Fully Assembled Robot:

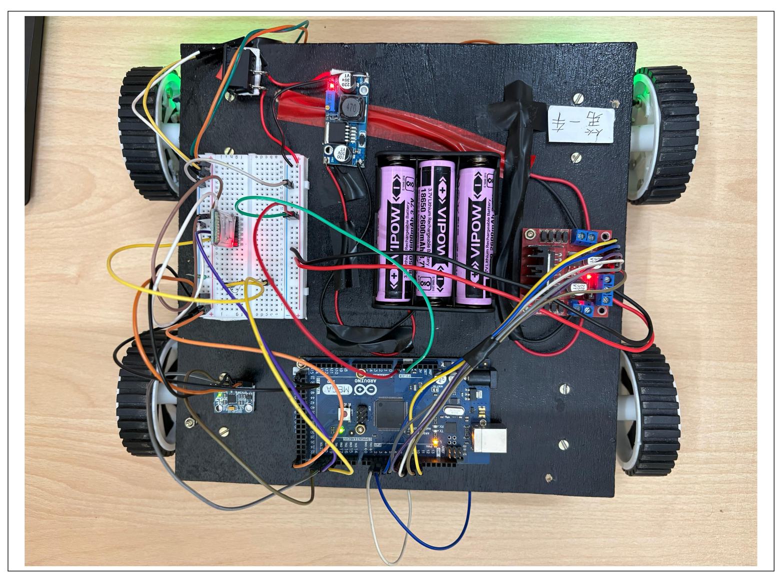

Figure 4: Top view of robot after assembly

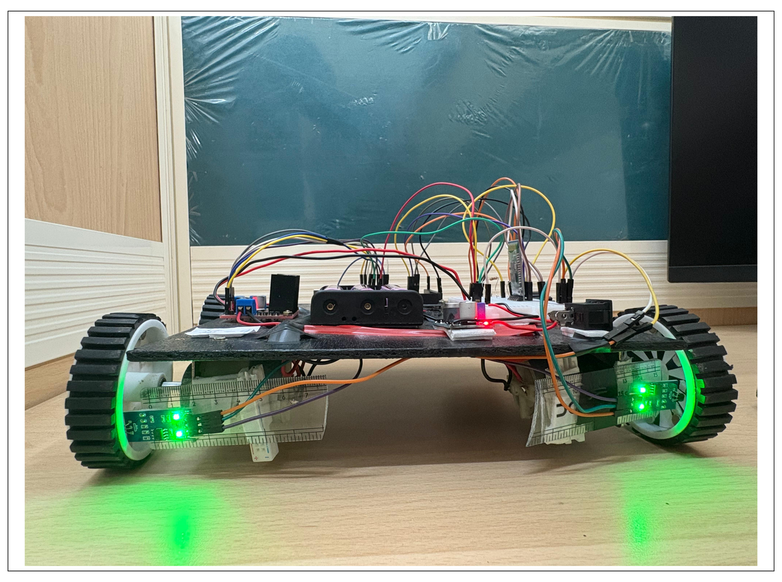

Figure 5: Front view showing sensor and encoder placement

4 Results and Observations

- Robots successfully navigated and performed position tracking using encoder-based feedback.
- Bluetooth enabled low-latency command updates from the central learner.
- Cascaded PID control allowed smooth, coordinated motion with minimal interference.
- The system gained reliable angular velocity from the gyroscope, enabling differential yaw control.

5 Conclusion and Future Work

This hardware-grounded MARL implementation provides a strong proof-of-concept for scalable, distributed robotic systems using affordable, resource-constrained platforms. Despite offloading the learning module to a central laptop, the design preserves local autonomy in sensing, actuation, and control. Bluetooth communication served as an effective medium for decentralized coordination.

In conclusion, this project demonstrates that key principles of Multi-Agent Reinforcement Learning can be realized on low-cost embedded systems with minimal hardware. The modular design, onboard control stack, and real-time responsiveness make this setup a strong baseline for further exploration of embedded MARL.

As the future part of the project, we aim to:

- Expand the system to support a larger team of robots with dynamic task allocation.
- Integrate camera-based vision modules for perception-driven decision-making.
- Incorporate real-time policy switching and on-the-fly learning for adaptive behaviors.

Acknowledgments

This internship project was completed under the SPARK 2025 Program at the Indian Institute of Technology Roorkee, under the mentorship of Prof. Rajdeep Niyogi. The authors thank the CSE department and lab staff for their support.