

# LARC 2025 VSSS Team Description Paper

Miguel Esteban Martínez Villegas  
*B.S. Mechatronics Engineering*  
A01640781@tec.mx

Ana Sofía Martínez Espinoza  
*B.S. Mechatronics Engineering*  
A01641268@tec.mx

Jesús Andrés Márquez Martínez  
*B.S. Computer Technology Engineering*  
A01641195@tec.mx

Gerardo Maximiliano López Herrera  
*B.S. in Robotics and Intelligent Systems Engineering*  
A01614489@tec.mx

**Abstract**—This paper introduces Robotec's Very Small Size Soccer robots designed for the 2025 Latin American Robotics Competition. Building on previous related competition experiences, our team has transitioned from a differential to an omnidirectional drive system. The robots feature a lightweight 3D-printed frame as well as custom-made omni wheels, enabling free and easy movement in all directions while maintaining stability. The robots are remotely controlled by a central workstation that is connected to the field's camera system; the video feed is parsed for entity coordinates and orientations which are fed to a Deep Reinforcement Learning model trained for providing high-level instructions, which are then converted into individual commands that are wirelessly sent to each robot.

**Index Terms**—IEEE VSSS; LARC; Software; Mechanics; Electronics;

## I. INTRODUCTION

The IEEE LARC VSSS competition aims to foster research and challenge teams to develop cutting-edge miniature robots capable of high-precision movement and intelligent decision-making in a compact field. As such, we present  $\mu$ -*Furbo*, an innovative robotic system engineered for maximum maneuverability and agility. It leverages a four-wheeled omnidirectional drive system with a low center of gravity design so as to quickly and precisely perform all of the movement commands indicated by the central workstation. A specialized computer vision program converts the camera feed into a unified coordinate system which is then feed to a Deep Reinforcement Learning model that acts as a high-level coach for all robots. Several other efficiency-focused implementations are used to improve both the reliability and performance of the robots.

This paper provides a technical overview of  $\mu$ -*Furbo*'s design, development, and implementation.

## II. SYSTEM DESCRIPTION

The robot developed for LARC 2025 represents a significant evolution from our previous iterations, incorporating key improvements based on lessons learned during previous years' related competitions. The new design featured enhanced mobility and stability to overcome the diverse challenges in a fast-paced soccer match scenario. The system description below outlines the hardware and software components.

### A. Mechanical Architecture

For simplicity and modularity, our three robots feature the same design and components. Taking some inspiration from

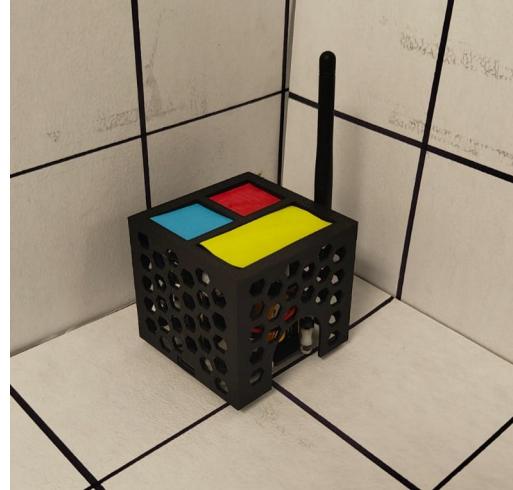


Fig. 1.  $\mu$ -Furbo VSSS robot

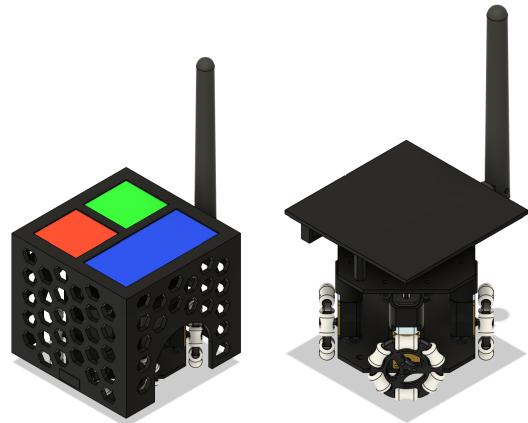


Fig. 2. Robot CAD (with and without uniform)

the RoboCup Small Size League category, they feature a symmetrical chassis with four omnidirectional wheels forming a cross shape.

In order to fit into the dimensions for the VSSS category, we had to create our own omni wheels featuring an innovative single-row design while adapting the overall shape to fit into a square footprint.



Fig. 3. Custom omni-wheel design

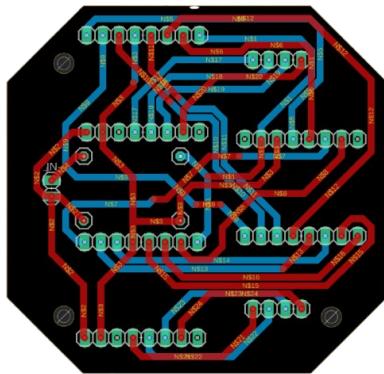


Fig. 4. PCB design overview

- **Dimensions:** 75 mm × 75 mm × 75 mm (80 mm × 80 mm × 80 mm with uniform)
- **Weight:** 0.23 kg
- **Materials:** PLA chassis, bronze support bars, silicone rubber rollers

This configuration allows the robot to achieve fast, precise, and holonomic movements.

#### B. Electronic Architecture

The entirety of the robot's electronic systems are housed on a dedicated, custom-made PCB that also serves as the middle level of the chassis. It includes the ESP32-C3 SuperMini Plus microcontroller, a pair of TB6612FNG dual DC motor drivers, a MP2307 Mini 360 DC step down converter, and input/output connectors for the battery and motors. The PCB significantly reduces wiring complexity, enhances safety and modularity, and provides a standardized approach towards system integration.

- Battery specifications:
  - **Nominal voltage:** 11.1 V
  - **Cells:** 3 cells, 3.7 V/cell
  - **Capacity:** 850 mAh
- Motor specifications:
  - **Model:** GA12-N20
  - **Gear reduction:** 42:1
  - **Rated voltage:** 12 V
  - **Rated current (no load):** 30 mA
  - **Rated speed (no load):** 381 RPM
- DC motor driver specifications:
  - **Model:** TB6612FNG



Fig. 5. 11.1 V, 850 mAh LiPo battery



Fig. 6. Ga12-N20 motor

- **Logic supply voltage:** 2.7-5.5 V
- **Power supply voltage:**  $\leq$  15 V
- **Output current:** 1.2 A (avg.), 3.2 A (peak)

- DC step down converter specifications:

- **Model:** MP2307 Mini 360
- **Input voltage:** 4.75-18 V
- **Output voltage:** 1-20 V
- **Output current:**  $\leq$  3 A

- Microcontroller specifications:

- **Model:** ESP32-C3 SuperMini Plus
- **CPU:** 32-bit RISC-V, 160 MHz clock
- **Supply voltage:** 3-3.6 V
- **Wi-Fi:** IEEE 802.11 b/g/n (2.4 GHz)

All components were chosen for their small size and efficient performance, with the ESP32-C3 in particular offering real-time wireless communication to the central workstation without the need for additional modules.

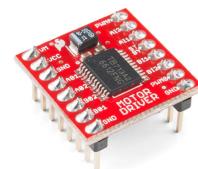


Fig. 7. TB6612FNG DC motor driver



Fig. 8. Mini 360 MP2307 DC step down converter



Fig. 9. ESP32-C3 SuperMini Plus microcontroller and Wi-Fi antenna

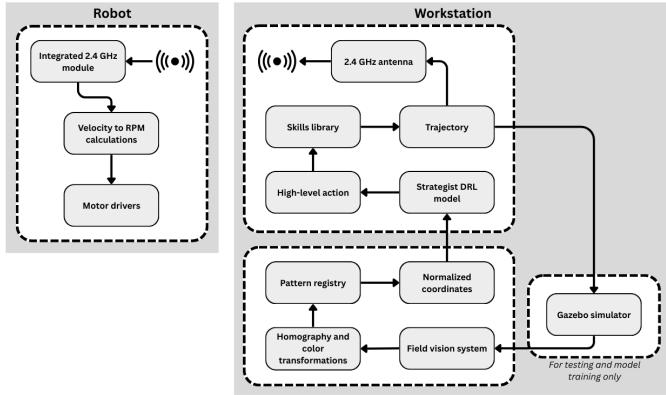


Fig. 10. The control system diagram shows the data flow from the workstation to the robotics system

### III. SOFTWARE ARCHITECTURE

This section presents a top-down view of the decision and control stack: perception and state estimation, strategy and role assignment, skill-level planning, low-level control, communication, and the simulation/training setup with evaluation metrics.

#### A. Overview

An overhead camera provides a video stream rectified to field coordinates. A perception module estimates ball and robot states and publishes a unified field state. A hybrid decision layer combines a learned multi-agent policy with a rule-based arbiter to produce role-conditioned *skill intents* (e.g., *goto*, *intercept*, *mark*). A skill library computes collision-aware trajectories and body-rate references  $(v_x, v_y, \omega)$ , which are tracked by the on-board controller. Intents and parameters are transmitted via a lightweight UDP protocol (micro-ROTA).

#### B. Perception and State Estimation

The camera stream is rectified via automatic homography; the play area is cropped and segmented in HSV space using a trained YOLO Convolutional Neural Network model. Robot identity and orientation are decoded from marker geometry using color masks. States are filtered with a lightweight constant-velocity Kalman filter to reduce jitter and provide velocities. The module publishes a field-

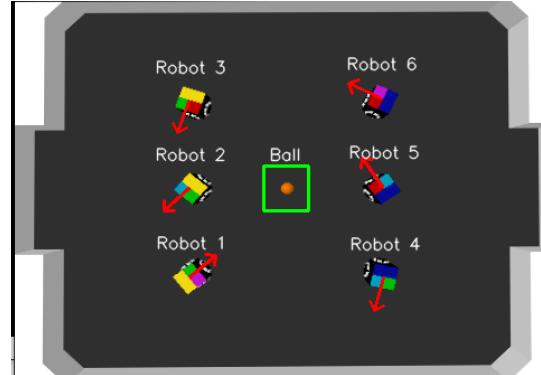


Fig. 11. Robot positions and orientations view

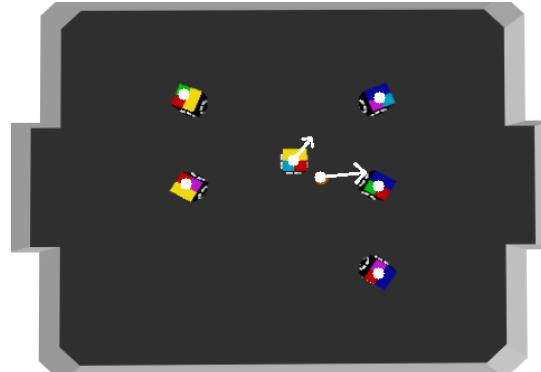


Fig. 12. Kalman filter positions and velocities

centric topic `/field_data` with ball pose/velocity and robot poses/velocities at 50 Hz.

#### C. State Representation

Each agent consumes a normalized observation vector

$$\mathbf{s}_t = [\mathbf{b}_t, \mathbf{r}_t^{(1)}, \mathbf{r}_t^{(2)}, \mathbf{r}_t^{(3)}, \mathbf{o}_t^{(1)}, \dots],$$

where  $\mathbf{b}_t = (x_b, y_b, \theta_b, \dot{x}_b, \dot{y}_b, \dot{\theta}_b)$ ,  $\mathbf{r}_t^{(i)} = (x_i, y_i, \theta_i, \dot{x}_i, \dot{y}_i, \text{role}_i)$  for teammates, and  $\mathbf{o}_t^{(j)}$  for opponents if available. A short history window (5 frames) is stacked to improve velocity/acceleration inference. All features are scaled to  $[-1, 1]$  in field coordinates.

#### D. Decision Layer: Strategy and Role Assignment

We employ a hybrid architecture: a multi-agent policy proposes role-conditioned intents (e.g., attack, support, block, intercept), while a rule-based arbiter enforces safety and tactical constraints (no go zones, spacing, restart procedures). Roles (striker, supporter, defender; goalkeeper if applicable) are reassigned at 20–50 Hz by minimizing a task-assignment cost combining (i) estimated time-to-reach, (ii) policy preference, and (iii) coverage/spacing penalties. We solve the assignment via the Hungarian algorithm and apply a 200 ms hysteresis to avoid role thrashing. Set-plays for restarts (kick-off, free balls) override the policy with scripted intents.

TABLE I  
LATENCY BUDGET

Camera exposure + readout	5–10 ms
Rectify + detect + track	5–10 ms
Policy inference	10–20 ms
Planner + skills	5–10 ms
Wireless transmit	25–50 ms
On-board control	5–10 ms
<b>End-to-end</b>	<b>55–110 ms</b>

### E. Skill Interface and Planning

The policy does not actuate motors directly. It outputs a discrete skill and parameters

$$\mathbf{a}_t = (\text{skill\_id}_t, \phi_t),$$

such as `goto`( $x^*, y^*, \theta^*$ ), `intercept`( $\tau$ ), `mark`( $\text{id}$ , offset), or `clear`( $\ell$ ). A safety layer projects infeasible commands onto the feasible set (speed/acceleration limits, boundaries, contact rules).

**Location Planner (goto).** Starts from a straight line reference, then deforms it using agent-centric repulsive fields (ORCA/potential-field style) to maintain separation; produces a time-parameterized trajectory under ( $v, a$ ) bounds.

**Interception Planner.** Fits the ball with a constant acceleration model over the last 200 ms and selects  $t^* \in [0.2, 1.0]$  s that minimizes the robot-to-ball distance subject to speed limits; covariance inflation under occlusion biases earlier intercepts.

**Mark/Block.** Places the robot on the opponent goal line with an offset; maintains bearing to the ball while respecting restricted areas.

### F. Low-Level Control (Holonomic)

Skill outputs are tracked as desired body rates ( $v_x, v_y, \omega$ ). For a 4-wheel omnidirectional base, wheel speeds  $\boldsymbol{\omega} \in \mathbb{R}^4$  follow

$$\boldsymbol{\omega} = \frac{1}{r} \begin{bmatrix} \sigma_{1x} & \sigma_{1y} & R_1 \\ \sigma_{2x} & \sigma_{2y} & R_2 \\ \sigma_{3x} & \sigma_{3y} & R_3 \\ \sigma_{4x} & \sigma_{4y} & R_4 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix},$$

where  $r$  is the wheel radius,  $(\sigma_{ix}, \sigma_{iy}) \in \{-1, 0, 1\}^2$  encode wheel mounting directions, and  $R_i$  are moment arms. A PID in  $(v_x, v_y, \omega)$  tracks the reference, saturating  $|\omega_i| \leq \omega_{\max}$  and rate-limiting accelerations to protect motors and drivers.

### G. Communication

We use UDP with a compact RTP-style header. Each packet contains a `skill_id`, parameters, and CRC. Heartbeats are sent at 10 Hz ( $\approx 32$  B/packet). Robots are considered "out of play" and stop receiving movement commands if no valid heartbeat is received within 2000 ms. Typical end-to-end latency (camera→robot) is 55–110 ms under field conditions.

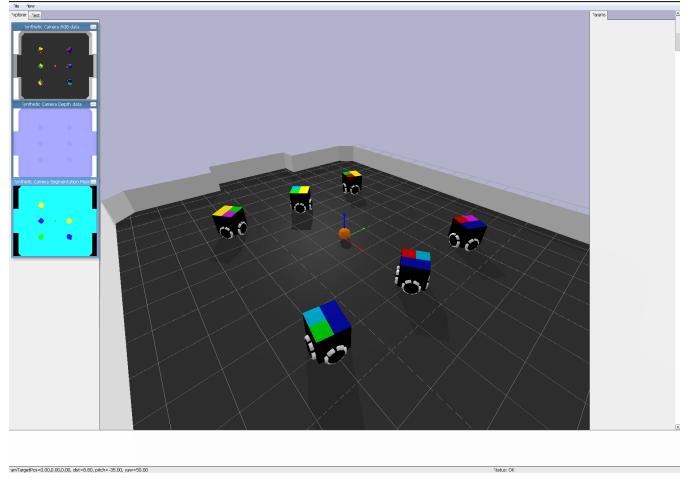


Fig. 13. Simulator view

TABLE II  
TRAINING HYPERPARAMETERS

Algorithm	MAPPO (shared policy)
Optimizer	Adam, lr = $3 \times 10^{-4}$
Discount $\gamma$	0.99
GAE $\lambda$	0.9
Entropy coef.	0.01
Rollout length	2048 steps
Mini-batches	4
Policy net	2×256 MLP (ReLU)
Curriculum	50M steps / stage
Domain rand.	$\mu \in [0.6, 1.0]$ , delay $\in [10, 100]$ ms, noise $\sigma = 0.05$

### H. Simulation, Training, and Sim-to-Real

We train in Gazebo/ROS 2 with curriculum progression (1v0 → 1v1 → 3v3). The algorithm is multi-agent on-policy (e.g., MAPPO). Domain randomization samples wheel/ball friction, mass, communication delay, lighting, and camera noise to reduce sim-to-real gaps. Policies are distilled to a 2-layer MLP (256/256) and executed at 50 Hz on the workstation; outputs are skill intents consumed by the skill library.

### I. Evaluation Metrics

We evaluate (i) robot bench tests, (ii) perception accuracy, (iii) gameplay performance, and (iv) communications reliability:

- **Bench:** max  $v/\omega$ , stop-to-stop time, battery runtime.
- **Vision:** position RMSE (cm), orientation error (deg) vs. ground truth.
- **Gameplay (5-min scrimmage):** time-to-possession, pass success %, shots/min, interception success.
- **Comms:** end-to-end latency, median packet loss %.

Baselines include a rule-based controller and ablated variants (no domain randomization, no role reassignment, no set-plays).

### IV. ACKNOWLEDGMENT

The authors would like to thank Oscar Eleno Carbajal Espinosa and Gerardo José Escobar Del Cid for all the support provided throughout the process.