Team Description Paper: ERA-IITK

Arbaaz Tanveer, Debraj Karmakar, Mayank Agrawal, Nikhil Gupta, Emaad Ahmed, Aayush Gajeshwar, Arnab Datta, Sushil Krishna K, Pranesh S, Rattandeep Singh Puar, Dhruva Singh Sachan, Kartik Kulkarni, Rishi Agarwal, Nishi Mehta, Aditya Jain, Shubham Mittal, Tejas Chikoti, Yukkta Seelam

Contributing authors: arbaaztanveer6173@gmail.com; debraj2003jsr@gmail.com; mayank.agrawal.532004@gmail.com; nikhilgupta0364@gmail.com; aayushgajeshwar06@gmail.com; arnab.datta.dbpc@gmail.com; sushilkkrish@gmail.com;

Institute: Indian Institute of Technology, Kanpur, India
Team Website: https://era.sntiitk.com/

Abstract

ERA-IITK is a team of undergraduates from Indian Institute of Technology Kanpur, India who are extremely passionate about robotics. Our team is dedicated to pioneering autonomous solutions for complex robotics challenges. Representing India at the international RoboCup MSL Challenge, we consistently strive to extend the frontiers of grassroots engineering through innovation and engineering. Our journey is characterized by rigorous research, creative problem-solving, and a dedication to learning, ensuring that each accomplishment serves as both a technical milestone and a testament to our collective spirit.

1 Introduction

ERA-IITK is an autonomous robotics team based at IIT Kanpur, founded in 2018 with the mission of advancing intelligent ground robotics in India. The team began its journey in the DJI RoboMaster AI Challenge, securing 3rd place in Canada in 2019 and consistently achieving global top-3 online finishes in 2020 and 2022. In 2023, we transitioned to the RoboCup Middle Size League (MSL), where we have demonstrated substantial technical progress, successfully qualifying for RoboCup MSL in both 2024 and 2025. Our development philosophy emphasizes affordable, modular hardware and efficient, low-complexity software architectures to build high-performance autonomous systems. Recently, the team was awarded the Hyundai Hope Scholarship 2025 for proposing an innovative solution in the automobile industry, further reflecting our commitment to impactful engineering and advancing autonomous robotics research in India.

2 Mechanical Design

Engineered for high-speed stability, the robot features a lightweight $52 \times 51 \times 79$ cm frame constructed from 20×20 mm aluminum extrusions. The chassis utilizes a hybrid structure, pairing 5mm and 3mm aluminum plates with 6mm acrylic sheets to balance structural rigidity with reduced mass. Mobility is delivered by a 4-wheel omni-drive system, with each motor independently suspended to ensure consistent ground contact and minimize jitter. By centralizing all heavy components within the lower aluminum assembly, the design maintains a low center of gravity, optimizing the robot's performance during aggressive acceleration and lateral maneuvers.

2.1 Dribbling Mechanism

The robot employs a parametric, closed-loop dribbling mechanism powered by dual 900 RPM Johnson motors. By utilizing 60 mm omni-wheels with a reconfigured vector orientation, the system maintains superior ball retention during high-speed lateral strafing—a common failure point in standard designs.The control logic is driven by an STM32-based PID loop that fuses two critical data streams: real-time elevation angles from dual encoders and the robot's chassis velocity.This synergy of adjustable geometry and predictive software ensures the ball remains "locked" to the bot during aggressive tactical maneuvers.

2.2 Kicking and Angle Control System

The robot features a high-energy solenoid kicker capable of ballistic launch speeds up to 9 m/s. Power is managed by a high-voltage circuit featuring a 450V, 4700μF capacitor bank, charged via a 380V AC inverter and rectifier. Firing is precisely modulated by a 200A rated IGBT, providing a robust 45% safety margin over the solenoid's 110A peak current.Precision targeting is achieved through an innovative 2-DoF angle mechanism. A 60 kg-cm DS6150 servo drives a kicking assembly mounted on linear bearings, allowing the impact point on the ball to shift vertically. This enables the bot to dynamically switch between grounded drives and aerial lobs by adjusting the launch angle in real-time via the STM microcontroller.

2.3 Goalkeeper's Shield

The goalkeeper's front shield is a 50cm x 78cm grid made completely of 20cm x 20cm aluminium extrusions. Additionally, it has 3 arms that can extend (one at a time) up to 10 cm to prevent the ball from entering the goalpost. The arms are powered by solenoid having a similar structure as the kicking solenoid. It has 900 turns of 20 awg insulated copper wire. It uses a 20mm mild steel plunger with a 6mm stainless steel head.

3 Electrical Design

The bot's electrical system is designed for reliability and modularity, consisting of the primary control PCB, power distribution system, capacitor charging and discharging circuit, emergency stop mechanism, and motor control electronics.

3.1 Central Control PCB

The central control system is built around an STM Nucleo-144 microcontroller, which acts as the interface between software and hardware, managing all bot functionalities. It handles eight encoders, six motors, one angle control servo, charging and firing commands, ball alignment LiDAR sensors, and an onboard compass.

3.2 Power Distribution System

The power system uses buck and boost converters fed by two batteries—an 11.1V Li-ion (yielding 15V, 12V, and 11.1V) and a 22.2V Li-ion (providing 24V, 12V, 8.5V, and 5V)—to supply various bot components. This isolation is ensured by using a 380V AC inverter through a DC rectifier.

3.3 Emergency Stop

The E-stop system is built around an Arduino Nano and HC-12 (SMD) RF transceiver, chosen for its superior signal penetration and long-range reliability. Operating on a completely isolated logic circuit, the module drives a high-speed relay that physically disconnects power to the motor drivers.

4 Self-Localization Approach

We pose the localization problem as an optimization problem using genetic algorithms, as done by Watanabe et al. [4]. We build upon this approach with two major improvements: (a) the use of Gaussian kernels to make the ground truth map smoother for faster convergence, and (b) fusing odometry to reduce the search space and noise.

4.1 Methodology

Ground Truth Map: We first precompute a binary ground truth map $M(x, y)$ of the white lines and apply a Gaussian blur to it.

Feature Extraction: White field lines are extracted from camera frames using traditional computer vision algorithms. These extracted lines are then made sparse to increase computational efficiency.

Genetic Algorithm: A genetic algorithm is used to find the optimal robot coordinates (x^*, y^*, θ^*) by maximizing the value of the fitness function F:

$$F(x, y, \theta \mid L_{in}, M_G) := \sum_{x', y'} T(L_{in} \mid x, y, \theta)(x', y') M_G(x', y'),$$

where $M_G(x, y) = M(x, y) * G(x, y, \sigma)$, G is the Gaussian kernel, and T performs a translation by x, y and a rotation by θ.

Fusion with Odometry: A weighted average of the genetic algorithm-based and odometry-based measurements is calculated, with the odometry weight being negatively proportional to the odometry buffer integrated over a short preceding time window (e.g., the past 100 milliseconds). This is done to overcome the noise of the genetic algorithm solution when the robot is stationary, as well as the noise of the odometry when the robot is moving.

5 Controls and Motion Planning

As the name suggests, this module is responsible for planning the trajectory of the robot from its current state to the desired target state (given by the decision module) in a dynamic environment. This is done by optimally planning the motion whilst simultaneously satisfying all constraints and avoiding collision with obstacles. The module uses ROS2 to communicate between different modules. It sends the generated angular velocities to the STM microcontroller via serial communication.

5.1 Path Planner and Controls

To plan the path from current location of robot to target location, we have implemented RRT* algorithm provided by OMPL [2]. To smoothen out the jitteriness of planned path (if it exists), we have used C2 Cubic Spline Approximation.
The low-level controls of the motion planning module implements linear Model Predictive Control (MPC) [3] using the ACADO [1] toolkit to optimize the time taken by the robot to move along the path while adhering to speed and acceleration constraints.
We observed that an optimal strategy for the robot is to always face the ball in order to quickly receive or block passes. Thus, θ_{target} is always set such that the robot maintains this orientation.

5.2 Goalkeeper Controls

Given predicted interception point from vision system, the goalkeeper trajectory controller evaluates the robot's positional error, $d = |y_{\text{intercept}} - y|$, and required direction, $s = \text{sgn}(y_{\text{intercept}} - y)$. The maximum robot acceleration is a_{acc} and

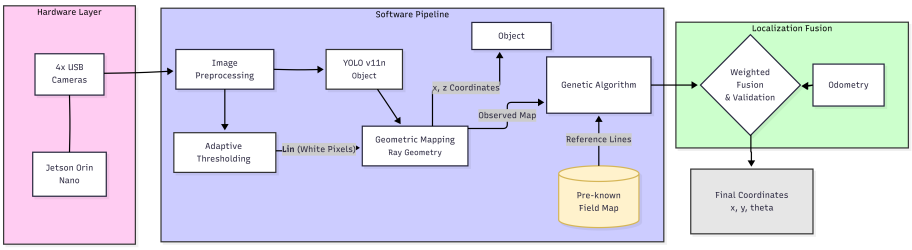

Figure 1: Overview of the Vision and Localization System

maximum robot deceleration is a_{dec}. We calculate a critical braking distance based on its current velocity v,

$$d_{\text{brake}} = \frac{v^2}{2a_{\text{dec}}}$$

When the robot enters this braking zone ($d \leq d_{\text{brake}}$), it determines the time required to reach the target under constant deceleration:

$$t_{\text{reach}} = \frac{v - \sqrt{v^2 - 2a_{\text{dec}}d}}{a_{\text{dec}}}$$

By comparing t_{reach} against the predicted target arrival time t_{ball}, the system makes a binary control decision $u \in \{0, 1\}$ (where $u = 1$ denotes acceleration and $u = 0$ denotes deceleration). The velocity command is then updated via Euler integration over time step Δt:

$$v_{k+1}^* = \begin{cases} v_k + s \cdot a_{\text{acc}} \cdot \Delta t & \text{if } u = 1 \\ v_k - s \cdot a_{\text{dec}} \cdot \Delta t & \text{if } u = 0 \end{cases}$$

These velocity commands are strictly bounded to prevent unintended direction reversal and absolute actuator saturation.

6 Vision System

The robot utilizes an omni-vision setup composed of four wide-angle IMX 179 USB cameras, all linked to a central computing unit based on a Jetson Orin Nano, achieving a frame rate of 18 fps.

All four cameras operate concurrently on four separate threads to preserve synchronization. In each thread, a YOLOv11 model executes continuously to identify key elements such as opponent robots, our robots, the ball, and the opponent's goal post. When an object is detected, it is segmented from the image, and its corresponding real-world coordinates are calculated. This is achieved using a calibration map that converts image pixel coordinates into real-world coordinates, providing accurate and consistent localization.

In addition, an array is used to temporarily store robot positions; these stored positions are periodically compared to reduce the impact of occlusions and to recognize cases where the same robot is visible in multiple camera views.

6.1 Goalkeeper Bot

The Goalkeeper Bot is fitted with Xbox Kinect Depth Camera for accurate 3D ball tracking. A YOLO11n model was trained on the data recorded and annotated by our team. The mean depth of bounding box is used as depth of the ball. As the range of the depth camera is limited, we have also incorporated a three monocular RGB camera system in the goalkeeper bot. We use this setup as default and shift to depth camera when ball is within range. We assume x_b, the mean position of the ball over five latest frames, and take v as the mean of the four v_i estimated as $(v_{xi}, v_{yi}, v_{zi}) = (\frac{\Delta x_i}{\Delta t}, \frac{\Delta y_i}{\Delta t}, \frac{\Delta z_i}{\Delta t})$ We solve for the time η when the ball's x-coordinate reaches the goalkeeper's x:

$$\eta = \frac{x_g - x_b}{v_x}$$

From this we calculate the point of interception of ball with the bot

$$(x_{\text{int}}, y_{\text{int}}, z_{\text{int}}) = (x_g,\ y_b + v_y\eta,\ z_b + v_z\eta + \tfrac{1}{2}g\eta^2)$$

7 Decision Algorithm

Over the previous version of decision making module, we have improved the defensive strategy algorithm introducing two new heatmaps and we also improved the ball passing logic.

7.1 Defensive Heat Map Equations

Multiple heat maps are generated to quantify field positions based on different factors for the defensive strategy. Each map is normalized to $[0, 1]$ and merged into a combined map for clustering optimal positions. The new maps introduced are defined as follows:

7.1.1 Defensive Opponent Influence Map

Defensive Opponent Influence: For each grid point (x, y), the defensive influence from all opponents is given by

$$H_{\text{def}}(x, y) = \sum_{\substack{o \in \mathcal{O} \\ \|[x,y]-o\| < r_{\max}}} \cos(\alpha_o)\big(r_{\max} - \|[x, y] - o\|\big)$$

where $\mathcal{O}$ is the set of all opponent positions, $r_{\max}$ is the maximum influence radius, and α_o is the angle between the vector from opponent o to the ball b and the vector from o to the grid point $[x, y]$.

7.1.2 Pass Block Map

Pass Block Influence: For each grid point (x, y), the pass blocking influence is determined by the highest-threat passing lane covering that point, given by

$$H_{\text{pass}}(x, y) = \max_{p \in \mathcal{P}} \Big(w_p \cdot \mathbb{I}\big(d([x, y], \overline{o_c p}) \leq \tau\big) \Big), \tag{1}$$

where o_c is the position of the closest opponent (the passer), $\mathcal{P}$ is the set of potential receiver positions, $w_p \in [0, 1]$ is the goal probability (threat level) associated with receiver p, and τ represents the line thickness threshold (half of the pixel thickness). The function $d([x, y], \overline{o_c p})$ calculates the shortest Euclidean distance from the grid point to the line segment connecting o_c and p, and $\mathbb{I}(\cdot)$ is the indicator function which equals 1 if the condition is met and 0 otherwise.

7.2 Passing Strategy

7.2.1 Micro-Level: Single Pass Probability Logic

The kinematic and spatial viability of a pass is computed as a probability score $P_{\text{pass}} \in [0, 1]$ by synthesizing three factors:

Distance Factor: Passing accuracy peaks at an optimal distance d_{ideal} and decays following a Gaussian distribution, where d is the Euclidean pass distance and σ controls the strictness of the decay:

$$F_{\text{dist}} = \exp\left(-\frac{(d - d_{\text{ideal}})^2}{2\sigma^2} \right) \tag{2}$$

Obstruction Factor: Opponent interference is evaluated based on orthogonal distance $d_\perp$ to the passing lane and longitudinal projection $t \in [0, 1]$. Because opponents closer to the passer have a wider interception angle, their threat is scaled by $(t + 0.2)$:

$$F_{\text{obs}} = \max\left(0, \min\left(1, 1 - \sum_i (t_i + 0.2) \cdot \max\left(0, 1 - \frac{d_{\perp i}}{N} \right) \right) \right) \tag{3}$$

(where N is a normalization factor).

Forward Progression Factor: Passes advancing toward the opponent's goal ($\Delta x > 0$) are incentivized using a hyperbolic tangent function (with steepness k) to provide a smooth tactical gradient:

$$F_{\text{fwd}} = \max\left(0, \min\left(1, 0.5 \cdot (\tanh(k \cdot \Delta x) + 1.25)\right)\right) \tag{4}$$

Final Micro-Probability: These spatial components are merged into a final weighted score:

$$P_{\text{pass}} = \max\left(0, \min\left(1, (w_1 F_{\text{dist}} + w_2 F_{\text{obs}}) \cdot F_{\text{fwd}}\right)\right) \tag{5}$$

7.2.2 Macro-Level: Multi-Step Pass Evaluation (DFS)

To evaluate multi-step passes, the system uses a bounded Depth First Search (DFS) to maximize long term tactical value.

Temporal Discounting: Future passing sequences are exponentially discounted to account for environmental entropy at recursion depth d:

$$\gamma(d) = \exp(-\lambda \cdot d) \tag{6}$$

Recursive Value Maximization: If a receiver secures a high probability shooting opportunity ($P_{\text{goal}} >$ threshold), the sequence terminates with the score:

$$\text{Path Score} = P_{\text{current\_pass}} + \gamma(d) \cdot (P_{\text{goal}} \cdot S_{\text{goal}}) \tag{7}$$

If no immediate shot is viable, the receiver becomes the new passer, and the algorithm recursively maximizes the single link pass probability across all available teammates.

8 Conclusion

In summary, ERA-IITK's TDP has outlined a holistic approach to RoboCup MSL, integrating a robust mechanical platform, modular electrical architecture, precise self-localization, advanced motion planning, and strategic decision-making. The emphasis on cost-effective, efficient hardware and innovative software solutions aims to enable faster, more agile, and tactically aware robots. With ongoing refinements and rigorous testing, we strive to achieve a higher level of autonomy and performance on the field, reflecting our commitment to advancing both the competitive and scientific frontiers of robotic soccer.

References

[1] B. Houska, H.J. Ferreau, and M. Diehl. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[2] Ioan A. Sucan, Mark Moll, and Lydia E. Kavraki. The open motion planning library. *IEEE Robotics Automation Magazine*, 19(4):72–82, 2012.

[3] Chengcheng Wang, Xiaofeng Liu, Xianqiang Yang, Fang Hu, Aimin Jiang, and Chenguang Yang. Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy. *Applied Sciences*, 8(2), 2018.

[4] Kaori Watanabe, Yuehang Ma, and Hidekazu Suzuki. Real-time self-localization using model-based matching for autonomous robot of robocup msl. *Journal of Robotics, Networking and Artificial Life*, 7(1):1–4, 2020.