Team Description Paper: ERA-IITK

Debraj Karmakar, Mayank Agrawal, Nikhil Gupta, Shivansh Mangal, Emaad Ahmed, Dhruva Singh Sachan, Kartik Kulkarni, Rishi Agarwal, Aayush Anand, Anirudh Singh, Arbaaz Tanveer, Aruz Awasthi, Joel Bansal, Nayani Om Swarup, Pranav Krishna, Shlok Jain, Vihaan Sapra, 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; mangalshivansh1720@gmail.com;

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

Abstract

ERA-IITK is a team of undergraduates extremely passionate about robotics from the Indian Institute of Technology Kanpur, India. Our team is dedicated to pioneering autonomous solutions for complex robotics challenges. Representing India at the international RoboCup MSL Challenge, we continuously push the boundaries of grassroots engineering through innovation and collaboration. Our journey is marked by rigorous research, creative problem-solving, and a commitment to learning, ensuring that every achievement is both a technical milestone and a celebration of our collective spirit.

1 Introduction

ERA-IITK, founded in 2018, is an autonomous robotics team from IIT Kanpur. We began our journey with the DJI RoboMaster AI Challenge—securing 3rd place in Canada (2019) and consistently ranking in the global top 3 online in 2020 and 2022. Since shifting our focus to RoboCup MSL in 2023, we have made significant progress despite tight budgets. Our approach emphasizes cost-effective hardware and innovative, low-complexity software solutions. We also aim to foster autonomous robotics research in India through RoboCup Open competitions.

2 Mechanical Design

The mechanical structure of the robot has been designed in order to maximize performance during the game by maintaining good speed, strength, agility and overall stability. The length, breadth and height of the robot are 50, 51 and 79 cm respectively. To minimize the weight of the robot, the main frame is made of 20mm x 20mm aluminium extrusions and 2 aluminium plates. The bottom most aluminium plate has a thickness of 5mm while the upper aluminium plate is 3mm thick. Rest all base plates are made of 6mm thick acrylic sheets.

All the heavy electrical and mechanical components have been placed in the bottom section between the aluminium plates to lower the centre of mass and provide a stable motion to the robot at high speeds.

2.1 Drive Motors

The holonomic drive robot uses Rhino 750rpm 10kgcm IG52 planetary gear motors mounted to the base plate via suspensions to move the robot at a max speed of around 4m/s. Bearing type omni wheels of 100mm diameter are attached to each motor using a coupling hub lock-key mechanism.

2.2 Suspension

Each motor is mounted on the bottom plate via 2mm steel suspensions to evenly distribute weight and minimize jitter. KP08 bearings secure the rear, while 10mm $\times$ 10mm aluminum shaft couplers with 50N/cm compression springs support the front, ensuring stable ground contact.

2.3 Dribbling Mechanism

The Dribbling is powered by two 900 rpm, 3 kg-cm Johnson motors paired with 60 mm omni wheels, which deliver excellent grip and performance for fast dribbling. The system employs active feedback control from four encoders (two on the dribbling arms and two on the motors) to manage the dribbling precisely. Additionally, the dribbling control is closely integrated with the bot's primary drive motors and the geometric placement of the dribbling mechanism on the ball, ensuring the ball moves in the same direction as the bot.

2.4 Kicking Mechanism

The kicking mechanism employs a custom built solenoid along with an adjustable angle control mechanism. It can shoot the ball at a max speed of 8-9m/s. Its power circuitry includes a 450v 4700uf capacitor which is charged via a 380v AC inverter through a rectifier. A PCB containing an IGBT controls the firing of the solenoid.

2.4.1 Solenoid

We designed a custom solenoid to generate maximum possible power. Kicking solenoid was made by 18 awg insulated copper wire. Its power was tested at 700, 770, 840 and 910 turns respectively and at 910 turns, the solenoid generated maximum force.The plunger is made of 25mm dia mild steel with 6mm Stainless steel head and has a max movement of 6cm.

Figure 1: Solenoid

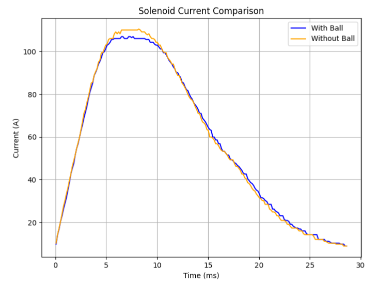

Figure 2: Current recorded at 700 turns

2.4.2 Angle and Power Control

We designed an innovative angle control mechanism using linear bearings on a rod, to which the kicking arm is mounted. A 60 kg-cm DS6150 servo motor moves the kicking assembly up and down, adjusting the impact point on the ball. This allows us to execute both aerial and grounded shots at any angle.
To control the power we are using a high power IGBT of 200 amps to control the solenoid ,it is controlled by a signal from the STM microcontroller using a gate driver. The solenoid reaches a peak current ∼110 amp so a 200 amp IGBT provides a safe margin

2.5 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.

2.6 Ball catching mechanism

To maximize the chances of a successful pass we used two tf luna lidars facing forward ,they detect the ball position and adjust the bot position accordingly when the ball is coming towards the bot so that the bot remains aligned to the ball.

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.

3.3 Emergency Stop

The emergency stop (E-stop) system utilizes an NRF24 module due to its high reliability and long range. A completely isolated circuit, based on an Arduino Nano and NRF24 module, sends a signal to a relay, which cuts power to the motor drivers, ensuring immediate halting of the bot when needed.

4 Self-Localization Approach

The central idea is to align a binary image of white field lines, obtained from processed camera data, with a known reference map of the field. By optimizing both translation and rotation, the algorithm estimates the robot's pose relative to the field. Initially, white field lines are extracted from the sensor data, and then a loss function is optimized—using differential evolution, a gradient-free global optimization method—to find the pose (x, y, θ) that maximizes the overlap between the sensor and field maps. This best alignment provides a global pose estimate that can later be fused with odometry for enhanced robustness.

4.1 Optimization Using Differential Evolution

This function takes candidate parameters (x, y, θ) and returns the negative overlap score, since the optimizer minimizes the function. The overlap is measured by counting the number of overlapping white pixels; a higher count indicates a better alignment. Differential evolution is employed to avoid local minima and ensure robust convergence without requiring differentiability.

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

This part of the motion planning module implements RRT* algorithm provided by OMPL [3] to plan the path from current location to target location, keeping a safe distance of 70 cm from the obstacles. If any obstacle comes in vicinity of the path, the path is replanned. If no such path exists then the robot is stopped gracefully at its current location. C2 Cubic Spline Approximation is used to smoothen the generated path. Then points are sampled at equal distances from this smoothened path to generate low level control.

5.2 Model Predictive Controller

This part handles the low-level controls of the motion planning module. It implements linear Model Predictive Control (MPC) [4] using the ACADO [1] toolkit to minimize the time taken by the robot to move from one point in the path to another while adhering to speed and acceleration constraints.

It turns out 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.

5.2.1 Kinematic Equations

The kinematic equations governing the MPC module are given by:

$$\dot{x} = v_x, \quad \dot{y} = v_y, \quad \dot{\theta} = \omega, \quad \dot{v_x} = a_x, \quad \dot{v_y} = a_y, \quad \dot{\omega} = \alpha.$$

These equations are followed by constraints on $v_x, v_y, \omega, a_x, a_y$, and α.

5.2.2 Optimization Formulation

Let $\hat{p} = (x_p, y_p, \theta_p, v_{x_p}, v_{y_p}, \omega_p)$ be the current state and $\hat{q} = (x_q, y_q, \theta_q, v_{x_q}, v_{y_q}, \omega_q)$ be the next target state point.

be the next target state. These state vectors, $\hat{p}$ and $\hat{q}$, are provided as inputs to the optimizer, which then solves the optimization problem to minimize the time required to transition from $\hat{p}$ to $\hat{q}$.

The optimizer outputs the optimal values of v_x, v_y, and ω. The individual angular velocities of the wheels are then computed using the inverse kinematic equation:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = J^{-1} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}$$

where J is the Jacobian matrix of the robot's kinematic model.

6 Vision System

The robot is equipped with an omni-vision system comprising four wide-angle IMX 179 USB cameras connected to a central computing device powered by a Jetson Orin nano, achieving a frame rate of 18 fps. These cameras operate concurrently on four separate threads to maintain synchronization. Within each thread, a Yolov11 model runs continuously to detect key objects, including enemy robots, our team's robots, the ball, and the enemy goal post. Once an object is detected, it is isolated from the image, and its real-world coordinates are computed . This is accomplished using a calibration map which maps the image pixel coordinates to real world coordinates, ensuring accurate and reliable conversion. Additionally, an array is made to temporarily store the positions of the bots; these positions are periodically compared to mitigate occlusion issues and to differentiate instances when the same bot appears in two camera views.

7 Decision Algorithm

This module is responsible for planning the game strategy by analyzing the current field state, which includes the positions of all objects on the field. It continuously evaluates various heat maps [2] to determine the optimal positioning for each robot and utilizes a structured decision tree to compute the next move, ensuring effective coordination and tactical execution. The module is written in python and integrated with all other domains using ROS2.

7.1 Heat Map Equations

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

7.1.1 Repulsion Map

Robots Repulsion: For each grid point (x, y), the repulsion from all robots is given by

$$H_{\text{rep}}(x, y) = -\sum_{p \in \mathcal{P}} \left(1 - \exp\left(-\frac{\|[x, y] - p\|^2}{2\sigma^2}\right)\right),$$

where $\mathcal{P}$ is the set of all robot positions and σ is a scaling parameter.

7.1.2 Attraction Maps

Ball Attraction:

$$H_{\text{ball}}(x, y) = A \exp\left(-\frac{\|[x, y] - b\|^2}{2\sigma^2}\right),$$

with b denoting the ball's position, A a scaling factor, and σ controlling the spread.

Goal Attraction: Let $h = (h_x, h_y)$ be the ball holder's position and $g = (g_x, g_y)$ the goal. Define: Then,

$$H_{\text{goal}}(x, y) = \cos\big(\theta(x, y) - \theta_g\big) \cdot \frac{p}{d(x, y) + \epsilon} \cdot \exp\left(-\frac{(d_g(x, y) - IGD)^2}{2\sigma^2}\right),$$

where IGD is the ideal goal distance, p is a scaling factor, and ϵ is a small constant.

Vertical Center Attraction:

$$H_{\text{vert}}(y) = 1 - \frac{|y|}{\text{field\_width}/2}.$$

Horizontal Right Attraction:

$$H_{\text{horiz}}(x) = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}},$$

with x_{min} and x_{max} as the field's minimum and maximum x-coordinates.

Ball Holder Circle:

$$H_{\text{circle}}(x, y) = \begin{cases} 1.0, & \text{if an enemy holds the ball and } \|[x, y] - h\| \leq r, \\ -0.5, & \text{if our team holds the ball and } \|[x, y] - h\| \leq r, \\ 0, & \text{otherwise,} \end{cases}$$

where h is the ball holder's position and r is the radius of influence.

Ideal Pass Distance:

$$H_{\text{pass}}(x, y) = A \exp\left(-\frac{(\|[x, y] - h\| - r_0)^2}{2\sigma^2}\right),$$

where h is the ball holder's position, r_0 is the ideal passing distance, A a scaling factor, and σ controls the spread.

7.2 Decision Tree

This part of the module evaluates the game state and makes strategic choices by first determining which team currently possesses the ball—whether it is the enemy, our own team, or neither. Based on the possession state, robot positioning is assigned by considering both the distance of each robot from the nearest clustered point on the field and the number of clustered points, which varies with ball possession. For instance, if the ball is not with our team, the robot closest to the ball is designated to move towards it. Additionally, if any robot is within a predefined threshold distance from the goal, that robot is tasked with taking a shot to goal. This structured approach ensures that decisions are made efficiently and responsively based on the dynamic state of the game.

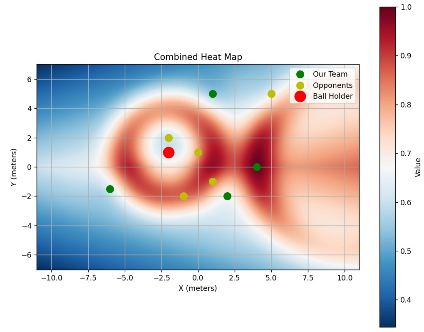

Figure 3: Combined Heat Map

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] António Fernando Alcântara Ribeiro, Ana Carolina Coelho Lopes, Tiago Alcântara Ribeiro, Nino Sancho Sampaio Martins Pereira, Gil Teixeira Lopes, and António Fernando Macedo Ribeiro. Probability-based strategy for a football multi-agent autonomous robot system. *Robotics*, 13(1), 2024.

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

[4] 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.