

# SRBL Team Description Paper

Small Size League, RoboCup 2025

Ali Shavakhi<sup>1</sup>, Alireza Rashvand<sup>2</sup>, Fatemeh Rashnozadeh<sup>2</sup>, Fateme Birashk<sup>3</sup>,  
Nima Najafpour<sup>3</sup>, Mohammadreza Nikkhoo<sup>3</sup>, Mohammadreza Karimi<sup>4</sup>,  
Melina Mehrani<sup>3</sup>, Maryam Marjani<sup>3</sup>, Amir Mohammad Jafari Harofte<sup>3</sup>,  
Romina Sahari Moghadam<sup>3</sup>, Mahdi Mohammadi<sup>3</sup>, Yas Meshkin<sup>3</sup>,  
Elena Heidary<sup>3</sup>, Arman Barazandeh Motlagh<sup>3</sup>, and Alireza Jafari Neysiani<sup>4</sup>

<sup>1</sup>Padova University, Padova, Italy

<sup>2</sup>I. Azad University, Qazvin Branch, Qazvin, Iran

<sup>3</sup>I. Azad University, Science and Research Branch, Tehran, Iran

<sup>4</sup>Sharif University of Technology, Tehran, Iran  
alishavakhi99@gmail.com

**Abstract.** This paper presents the SRBL Robotics Team’s advancements in designing and developing robots for the RoboCup 2025 Small Size League (SSL). Our approach integrates cutting-edge mechanical engineering, electronics, and artificial intelligence to optimize gameplay performance. The team has focused on enhancing motor control, ball-handling mechanisms, and strategic decision-making algorithms to create agile and efficient robots. Key innovations include real-time opponent prioritization, region-based movement optimization, and GPU-accelerated game planning. Additionally, we have developed a robust hardware architecture featuring an STM32 microcontroller, precision motor drivers, and a low-latency wireless communication system. The mechanical design incorporates lightweight materials, an omni-directional drivetrain, and an adaptive ball-handling system to ensure dynamic maneuverability. This paper details our design methodologies, technical implementations, and future enhancements to further improve competitive performance in SSL robotics.

## Introduction

The RoboCup Small Size League (SSL) is a competitive robotics domain that challenges teams to develop autonomous multi-agent systems capable of performing in dynamic, high-speed environments. The SRBL Robotics Team, established in 2024, is dedicated to advancing robotic soccer through the integration of innovative mechanical design, intelligent control algorithms, and real-time decision-making frameworks. As a first-time participant in RoboCup 2025 SSL, the team aims to enhance robotic performance by focusing on key aspects such as precise motion control, efficient ball-handling mechanisms, and optimized strategic planning.

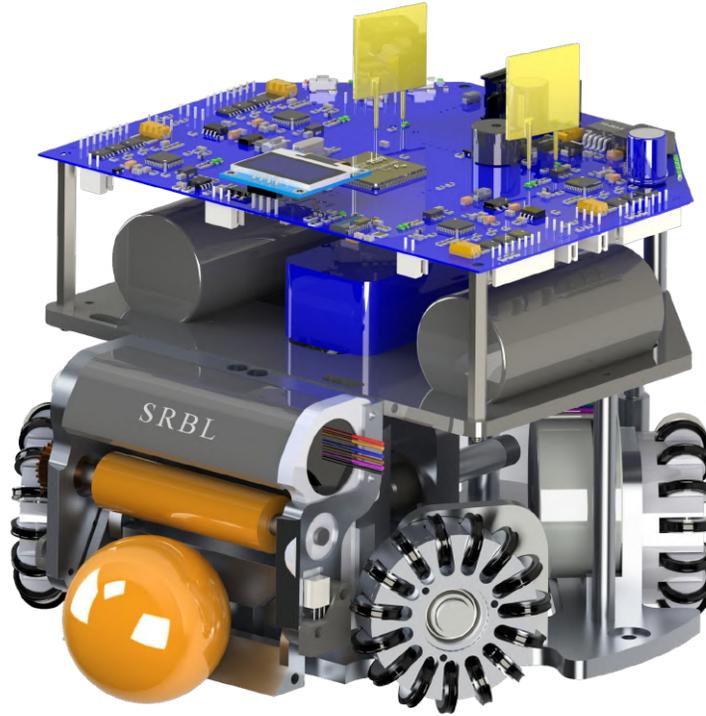


Fig. 1: SRBL Robot

To achieve these objectives, the team has employed a multidisciplinary approach, leveraging expertise in mechanical engineering, electrical engineering, and computer engineering. The development process emphasizes three core areas: (1) the design and implementation of a robust hardware system, including an STM32 microcontroller, high-efficiency motor drivers, and a low-latency wireless communication framework; (2) the integration of AI-based decision-making algorithms that facilitate real-time opponent analysis, field segmentation, and gameplay optimization; and (3) the implementation of a lightweight, omni-directional drivetrain with an adaptive ball-handling system to improve maneuverability and precision in gameplay.

This paper presents the systematic development of the SRBL robotic system, as shown in Fig. 1, detailing the methodologies adopted in hardware and software design, computational efficiency enhancements, and game strategy formulation. The remainder of this paper is structured as follows: Section 1 discusses the software architecture, including strategic decision-making frameworks, motion planning, and real-time visualization tools. Section 2 details the electronic components, highlighting the communication protocols, motor control strategies, and sensor integration. Section 3 describes the mechanical system, including drive-

train optimization, ball-handling mechanisms, and structural enhancements. Finally, the paper concludes with an evaluation of performance metrics and future research directions aimed at further refining robotic capabilities in SSL.

## 1 Software

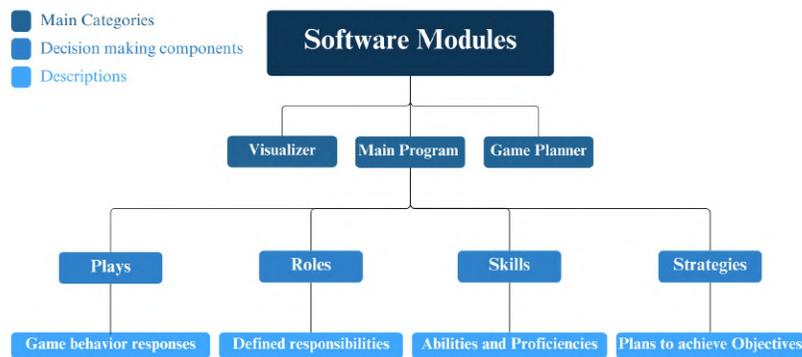


Fig. 2: Software layer and part's

### 1.1 Main program

This is the core decision making system. It has four parts that are connected and work together to decide what each robot should do in current game situation (see Fig. 2).

**Plays:** Each play may be triggered by specific behaviors or actions in the context of the game. For example, a defensive play dictates how robots should position themselves when preventing an attack. Each play represents a tactical approach to the current situation.

- A play determines how the team should act as a whole, balancing individual robot behavior with collective goals.
- It selects and executes strategies depending on the game situation.
- It ensures adaptability, allowing the system to respond to real-time changes.
- It assigns a singular role to each individual robot.

**Roles:** Each play should assign some role to all robots in the team. For example, one robot might be assigned the role of defense while another focuses on offensive actions. The roles are flexible and can be adapted based on the evolving gameplay scenario.

- Roles are dynamic; a robot can switch between them as needed based on the specific play assigned.
- Each role operates under different conditions, with its own decision-making process.
- The overall gameplay scenario determines how roles are distributed among robots.

**Skills:** in this system, individual actions or tasks a robot can perform is called skill, such as kicking, passing, or moving to a specific point. Skills are usable among all parts of the system. a role may consist of a set of skills required to perform its function. skills are modular and reusable, meaning that multiple roles can use the same skills.

**Strategies:** Each play can be executed in multiple ways, and these different execution methods are referred to as strategies. A strategy defines the approach, decision-making process, and tactical adjustments for a play. The AI selects the best strategy based on factors such as opponent positioning, ball movement, and teammate availability.

- Strategies define the flow of decision-making, helping the system transition between different states (e.g., offense to defense).
- Strategies ensure coordination, making sure individual robots work toward a common goal.

## 1.2 Game Planner – Strategic Decision Support

The Game Planner is responsible for analyzing the game environment and supporting strategic decision-making. It provides high-level game context that influences how robots perform both offensively and defensively.

**Prioritizing Opponent Robots** This component plays a crucial role in defensive tactics. It evaluates the importance of each opponent robot and calculates a dynamic score based on factors such as the location of the controlling robot (ball carrier) and the relative positioning of other opponents. This score helps the system determine which opponent to prioritize for defense or attack.

**Regioning** Regioning involves analyzing the field for empty spaces to optimize strategy selection. By simulating light blocking from obstacles, the Game Planner divides the field into regions that are critical for making decisions about passes, positions, and off-ball movements.

**Computational Efficiency** Due to the computational complexity of its algorithms, the Game Planner leverages GPU-based computing (e.g., CUDA) to speed up real-time processing. Using technologies such as OpenCL and Direct-Compute, it can process data more efficiently, ensuring that real-time game decisions happen without delays.

### 1.3 Visualizer – Monitoring and Debugging Tool

The Visualizer is a graphical tool designed to monitor and debug the system during live matches or simulations. It offers real-time insights into the behavior of the robots and their interactions on the field, as shown in Fig. 3.

**Real-Time Insights** The Visualizer displays various elements of the game, including robot positions, ball location, and strategy execution, enabling the team to understand how the system behaves in different scenarios.

**Debugging Support** It is also a critical tool for troubleshooting. You can track the performance of individual systems, such as the robot’s sensors or controllers, making it easier to identify and fix issues in real time.

**Tuning and Calibration** The Visualizer includes tuning tabs for high-level control parameters and 'Active Role' settings, providing easy access for adjustment during debugging and system calibration.

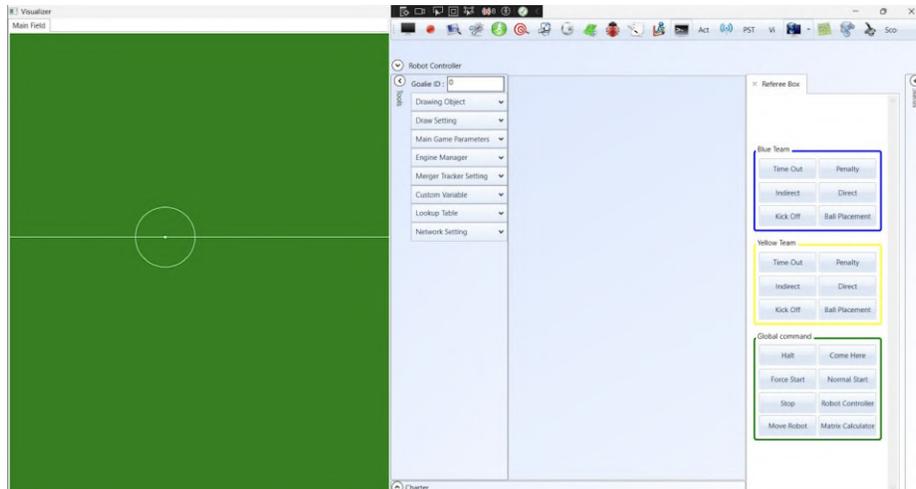


Fig. 3: Visualizer and Debugging Tool

## 1.4 Supporting Tools

**Kalman Filter:** The Kalman filter implemented to serves as a critical tool for real-time state estimation and prediction in motion tracking systems. It combines two primary steps—prediction and correction—to provide accurate estimates of the system’s state over time, even when faced with noisy sensor data.

- The filter estimates not only the position of the robots and the ball but also their speed, acceleration, and direction.
- Despite the challenges of noisy sensors and fast movements, the Kalman filter ensures data accuracy, helping robots respond quickly and effectively during a match.

**ERRT Algorithm** The ERRT algorithm is a motion planning system designed to navigate robots in dynamic and constantly moving environments. It builds a tree of possible robot positions, expanding from the starting point toward the goal while avoiding obstacles [1].

- The algorithm moves the robot step by step toward its goal while checking for obstacles.
- If it encounters an obstruction, it adjusts its direction accordingly.
- The algorithm runs in the background, ensuring it does not interfere with other processes.

**grSim** grSim is a 3D simulator that replicates SSL gameplay, allowing for testing and refinement of AI strategies without deploying them on a physical robot.

- grSim is a real-time 3D simulator that replicates SSL gameplay.
- Its network-based architecture ensures seamless AI integration.
- With real-time configurability and SSL-Vision compatibility, grSim has become an indispensable tool for AI performance enhancement.

## 2 Electronics

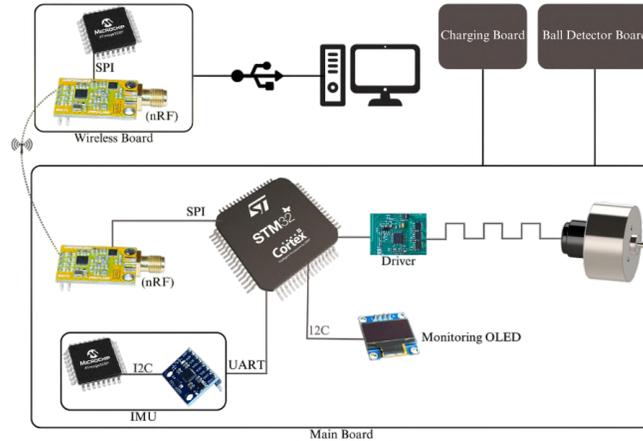


Fig. 4: An overview of electronics

### 2.1 Main board

**Power supply** Each robot is powered by a 4-cell Li-Po battery with a capacity of 3800mAh and a discharge rate of 45C. The MEW1022 DC-DC regulator is used to separate the two main power sections: one for the DC brushless motors and voltage booster, and the other for the STM32F646ZGTx microcontroller, which is operated at 216 MHz, and any related sensitive voltage components. To ensure safe transmission between these sections, the ADuM7440 optocoupler is used.

**Wireless communication** The nRF24L01+ PA/LNA module on the main board is responsible for wireless communication between the robot and external systems. Instead of the standard nRF24L01+, we chose the power amplifier (PA) and low noise amplifier (LNA) version, which provides stronger signal transmission, improved reception, and increased range to ensure stable and reliable data exchange. This module communicates with the STM32F7 microcontroller on the main board via the SPI protocol, enabling efficient two-way data transmission. It allows the robot to receive commands and send feedback in real-time, ensuring low-latency and robust wireless communication.

**IMU** To achieve precise motion control in the robot, the MPU-6050 module is used. This 6-axis motion tracking sensor integrates a 3-axis accelerometer and

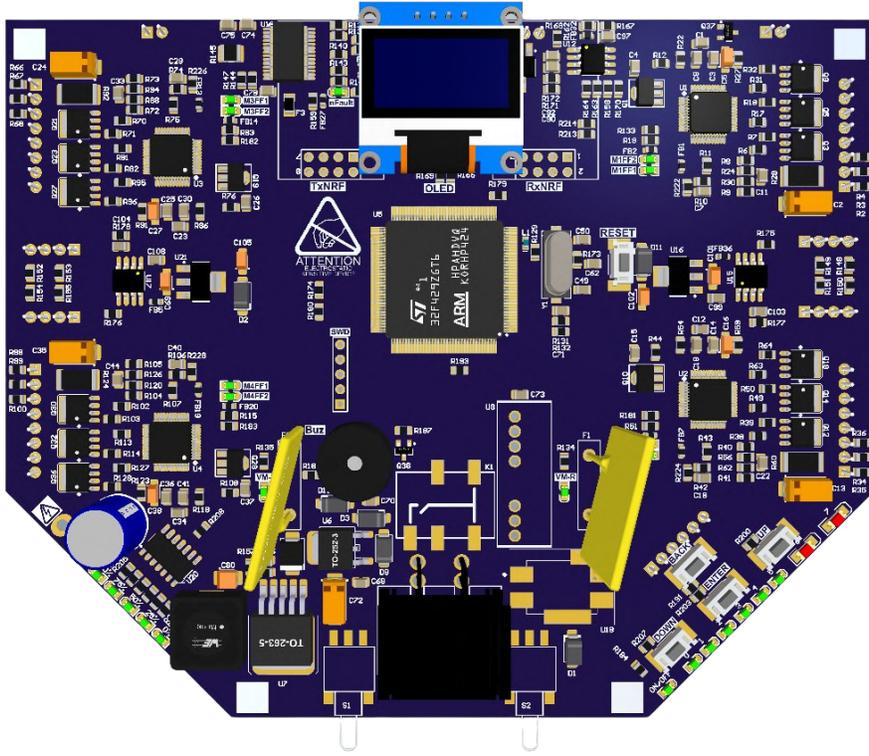


Fig. 5: Main board

a 3-axis gyroscope, along with a Digital Motion Processor (DMP) that enables onboard motion processing. The sensor allows measurement adjustments based on acceleration and angular velocity. The output data from the MPU-6050 is first processed through an ATmega328 microcontroller, which collects the sensor data and transmits it via UART to the STM32F7 microcontroller. The STM32F7 then applies Kalman filtering to refine motion control for the robot.

**Driver** The system employs the A3930, an MOSFET-based automotive 3-phase BLDC controller, to drive brushless motors efficiently. This driver manages motor operation by precisely controlling the switching of MOSFETs, ensuring smooth and stable performance. Each robot integrates the A3930 for motor control, enabling accurate motion handling. The driver interfaces with the main board through PWM signals for speed control and GPIO signals for direction and enable or disable functions. The MOSFETs used in the motor driver circuit include IRLZ44N and IRFR3806, ensuring efficient power delivery and switching performance. Additionally, the dribbler module operates with a separate motor driver system, ensuring independent and efficient functionality (see Fig. 5).

## 2.2 Charger board

The main board controls the charger board by sending a charge enable signal and selecting the kick mode. The charger regulates power, manages current sensing, and ensures proper capacitor charging before activation. Feedback signals report errors and temperature status to the main board for safety. Key components include LM393M for signal comparison, TC4427 for MOSFET driving, and voltage regulators for power stability. The capacitor takes approximately 9 seconds to fully charge before the kicker can be activated.

## 2.3 Ball-detector board

The main board supplies power and control signals to the sensor board, ensuring proper operation. It provides a voltage source and ground connection while also sending possible control signals for activation. The sensor board processes infrared signals to detect objects, using LM393M for signal comparison and LMC555CM for generating modulation signals. An opto-isolator processes the received infrared reflections, and the sensor output is sent back to the main board via a GPIO or an interrupt pin for real-time object detection.

## 2.4 Wireless board

The wireless board is designed to act as a bridge between the PC and the robot, ensuring stable and efficient data transmission. It features an STM32F405RGT6 microcontroller and an nRF24L01+ PA/LNA module, which work together to receive data via USB from the PC and then transmit it wirelessly to the robot. The USB-to-serial interface (FT232RL) allows seamless communication between the PC and the STM32, converting USB signals into a format that the microcontroller can process. The nRF24L01+ PA/LNA module enhances communication stability by providing higher transmission power and improved signal reception, reducing data loss and ensuring reliable two-way communication. The SPI protocol is used to exchange data between the STM32F4 and the nRF24L01+, while additional interfaces, such as UART and Ethernet, allow flexible connectivity options. This system ensures low-latency wireless data transmission, allowing the robot to receive control commands in real time and send feedback efficiently [2].

### 3 Mechanics

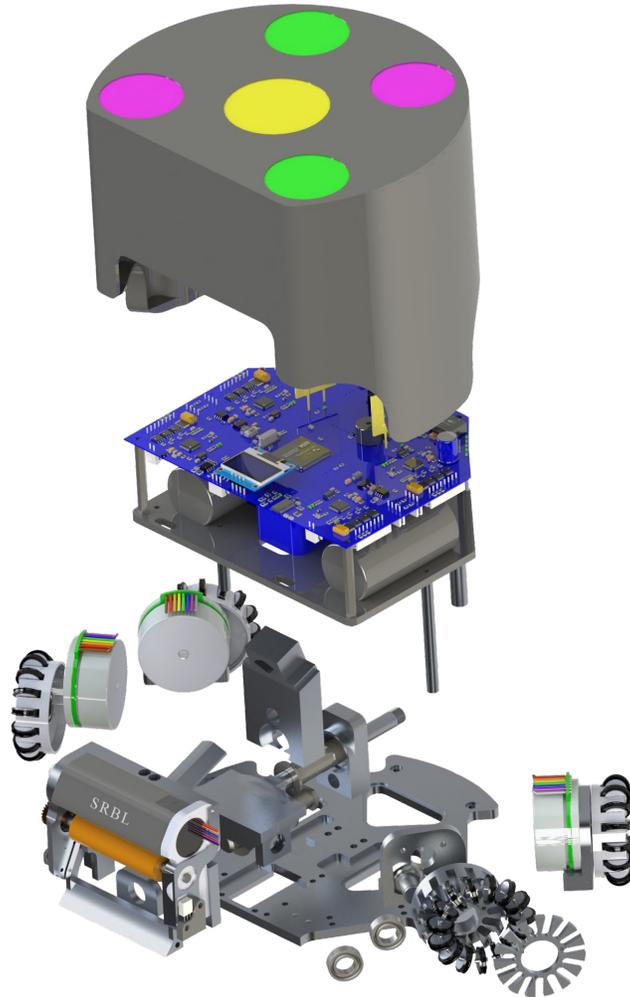


Fig. 6: Robot Design

#### 3.1 General Overview

Our mechanical design ensures that our robots are adaptable for diverse game strategies. Key objectives include enhanced ball control through our custom designed dribbling system and omni-directional drive systems [3]. we have used efficient materials and we have developed precise kicking mechanisms through a solenoid-based kicker for dynamic shooting power in varied game scenarios. Our competitive edge lies in several mechanical features including a lightweight

chassis. As first year participants, we have drawn inspiration from the experiences of other teams in developing our mechanical design and we have followed the SSL guidelines accordingly [1,3,4,5].

Table 1: Specifications of the robot

Parameter	Specification
Dimension	180mm diameter, 140mm height
Weight	2.8kg
Power Supply	14.8V
Driving Motor	Maxon EC45
Wheel Diameter	52.76mm
Dribbling Motor	Maxon EC16

### 3.2 Drivetrain

**Motors** Due to the SSL requirements, we selected Maxon EC45 brushless motors for their high torque (90.5mNm), efficiency, and precision, further enhancing performance with the US Digital E4P360 encoder. Each motor is mounted on a custom CNC-milled aluminum holder engineered for durability and precision, featuring a central hole for exact alignment.

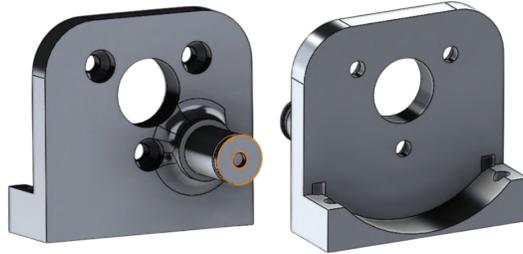


Fig. 7: The Motor Holder's Front and Back

The motor holder displayed in Fig. 7 plays a crucial role in securing the motor while maintaining structural integrity; it features countersunk screws with

conical holes, which are specifically designed to accommodate flat-head screws so that they sit flush with the surface. These conical shaped holes prevent possible movement of surrounding components.

**Chassis Design** For the chassis we used aluminum alloy because in addition to being lightweight it is strong enough to withstand potential impacts. The chassis employs a compact and aerodynamic design optimized for ball handling. To further enhance maneuverability, the robot's geometry was designed to maximize internal space efficiency, ensuring all components fit seamlessly. Motors, wheels, and other components are securely attached using screws and threaded inserts, ensuring precise alignment and easy replacement as illustrated in Fig. 8.

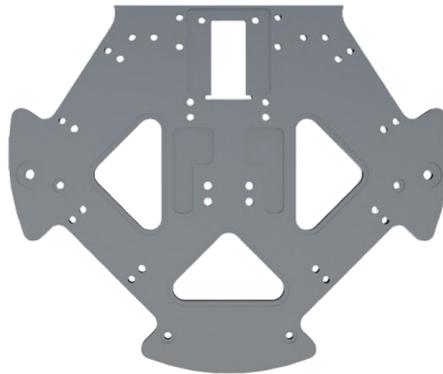


Fig. 8: Robot's Chassis

**Wheels** The drivetrain utilizes an omni-wheel configuration with high-precision wheels, featuring 15 sub-wheels which allows smoother transitions and movement in any direction without the need to rotate the robot's body. Our team uses wheels with a diameter of 52.76mm and we selected a  $45^\circ$  angle for the back wheels, as it provides a better balance between lateral and axial forces. Additionally, our wheels incorporate an aluminum ring and a custom designed pulley, to enhance transmission efficiency [3].

### 3.3 The Dribbling system

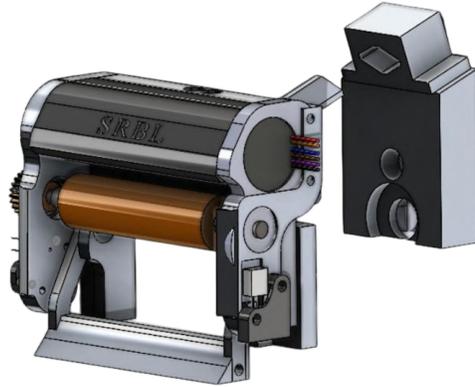


Fig. 9: Robot Interface Module

**Dribbler** In our initial design, a significant challenge we faces was that the dribbler permitted more than 20 percent of the ball’s surface to enter the robot during motion, violating competition regulations. this was because the dribbler is engineered with a degree of flexibility to reduce inconsistencies in ball handling. Inspired by the the MRL team’s robot design [6], we refined the design to ensure that exactly 20 percent of the ball is covered while maintaining reliable ball retention. our moving dribbling system incorporates a foam component that functions as a damper, mitigating the dynamic forces during operation and enhancing overall performance (see Fig. 9) [6].

**Roller** The roller material was carefully chosen to provide both grip and flexibility. We used TPE, but due to excessive softness, we switched to TPU, which provides better ball handling while preventing excessive deformation. The roller surface is CNC-machined and slightly textured to optimize friction balance. The roller’s width is 55.5mm; its outer diameter is 7.5mm and its inner diameter is 2.9mm.

### 3.4 Kicker

The kicker system provides powerful, and controlled kicking capabilities, allowing the robot to execute precise passes and powerful shots with efficiency and

reliability. The chip operates via an encoder-assisted rotating mechanism, allowing dynamic shot trajectory adjustments. The controlled launch angle ensures adaptability in different game scenarios.

The solenoid-based kicking system generates a strong magnetic field to propel the plunger forward, delivering controlled and powerful strikes and its managed by a switching circuit that regulates current flow, ensuring consistent shot performance. The solenoid is securely placed at an optimized position which maximizes energy transfer using an aluminum bracket with vibration-damping rubber pads, ensuring stability and reducing energy loss during impact. The control system dynamically adjusts the voltage to regulate shot intensity. The kicker assembly is securely positioned using an aluminum mounting bracket with vibration-damping components which ensure mechanical stability and precise alignment with the ball's resting position[5].

### 3.5 Robot Cover

The robot cover, made from PLA and manufactured using 3D printing, protects all electronic components. A Butterfly Pattern marking has been incorporated onto the cover for vision detection following standard competition regulations.

## References

1. D. Ferguson, N. Kalra, and A. Stentz, "Replanning with rrts," 2006.
2. M. K. Naeini, A. G. Poudeh, A. Rashvand, A. Dalvand, A. R. Doost, M. A. Keivanani, S. Razavi, S. Esmaelpourfard, E. Fathi, S. Mosayebi, and A. Adhami-Mirhosseini, "Mrl extended team description," 2020, extended team description document.
3. T. Röfer, A. Löttsch, and G. Martius, "Omnidirectional drive," 2004. [Online]. Available: <http://robocup.mi.fu-berlin.de/buch/omnidrive.pdf>
4. M. Sobhani, M. Bagheri, A. Hosseinikia, H. Shirazi, S. Mosayebi, S. Esmaelpourfard, M. K. Naeini, and A. Adhami-Mirhosseini, "Mrl extended team description 2017," 2017.
5. A. G. Poudeh, V. K. Nejad, A. Dalvand, A. R. Doost, M. A. Keivanani, H. Shirazi, S. Esmaelpourfard, F. Rashnozadeh, S. Mosayebi, M. K. Naeini, and A. Adhami-Mirhosseini, "Mrl extended team description 2017," 2017.
6. M. T. et al., "Mrl extended team description paper 2019," 2019. [Online]. Available: [https://ssl.robocup.org/wp-content/uploads/2019/03/2019\\_ETDP\\_MRL.pdf](https://ssl.robocup.org/wp-content/uploads/2019/03/2019_ETDP_MRL.pdf)