Sysmic Robotics
Team Description Paper 2025

Daniela Moya, Claudio Tapia, Gabriel Arcaya, Nicolás Mansilla, Matias Faundes, Gerson Marihuan and Nelson Cortés

Universidad Técnica Federico Santa María
Av. España 1680, Valparaíso, Chile

Abstract. This paper presents the latest advancements in the fifth generation of Sysmic Robotics' robots, focusing on improvements since RoboCup 2024. The mechanical structure has been redesigned with 3D-printed PLA components, enhancing durability and flexibility while optimizing component placement. In hardware, the kicker board was refined by eliminating redundant components, integrating a level shifter, and correcting voltage errors. Additionally, a new base station PCB was developed to improve communication reliability. The software transitioned from third-party systems to an in-house architecture, incorporating advanced motion planning and the STP framework for strategic decision-making. Finally, the firmware was enhanced with a rigorously defined kinematic model, ensuring better velocity calculations and movement control.

Keywords: RoboCup · Small Size League · Mobile robots

Introduction

Sysmic Robotics is a team of engineering students from the Technical University Federico Santa María. The team first participated in the Small Size League (SSL) during the 2018 RoboCup in Montreal under the name *AIS*, competing in Division B. In 2023, the team made its second RoboCup appearance in Bordeaux. Since then, the team has worked to upgrade their robots, culminating in their qualification for the 2024 RoboCup in the Netherlands.

For RoboCup 2025, our team has focused on key improvements in four main areas: mechanics, hardware, software, and firmware. These enhancements aim not only to meet RoboCup standards but also to lay the foundation for future developments. The robot's structure was redesigned using 3D-printed PLA components; the kicker board was optimized by removing redundant components and integrating a level shifter while correcting power supply errors; a new PCB for the base station was developed to improve communication reliability; a critical transition from third-party systems to our own architecture was completed, incorporating advanced motion planning and the STP framework for strategic decision-making; and a rigorously defined kinematic model was implemented,

enabling more precise velocity calculations and better motion control.

This document details these advances, organized into sections that cover each of the mentioned areas. Our goal is not only to compete in RoboCup 2025 but also to contribute to the robotics community with a scalable, efficient, and well-documented system that serves as a foundation for future innovations.

1 Mechanics

Since our last submission for the 2024 RoboCup, the robot has undergone several slight structural changes.

1.1 Structure

Our previous version, described in our 2024 TDP [1], featured a structure held together by two 3 mm-thick MDF discs. One of these served as the base of the robot, supporting all the components, while the other, located at the top, held the mainboard. From this point onward, these will be referred to as the *bottom base* and the *top base* for clarity. Although this technique worked well for us for years, issues arose as the repeated assembly and disassembly of the robot caused the screw holes to wear out (ass seen in Fig 1 and 2), necessitating the replacement of the bases. This problem, compounded by the breakdown of our university's laser cutter, led us to explore new approaches.

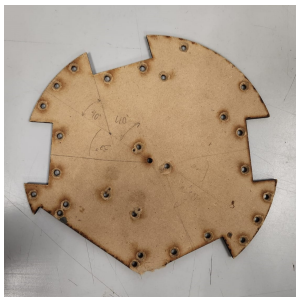


Fig. 1: Bottom base with worn-out holes.

Fig. 2: Top base with a missing section due to worn-out holes.

Moving from MDF discs to fully 3D-printed discs had always been an idea for our team, so it was logical for us to use PLA to print both, the bottom base and the top base. This approach was not only affordable, as we used materials that were already available to us, but it also expanded the possibilities for redesign, as we are now able to include the countersunk depth, which was not possible with MDF. The 2024 version of the bottom base featured straight holes for bolts

only, as shown in Fig. 3. In contrast, the 2025 version includes countersinks that allow screw heads to sit flush with the surface, four cutouts to lower the wheel motors, and a holder designed to fit the battery, as shown in Fig. 4.

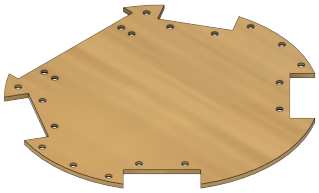


Fig. 3: 2024 version of the Bottom Base, made of MDF.

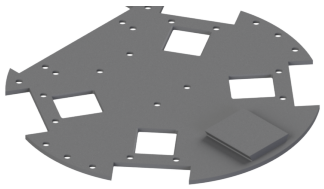


Fig. 4: 2025 version of the Bottom Base, made of PLA filament

The top base faced similar problems: the straight holes wore down, and small holders were required to secure the mainboard. These holders were screwed into both the top base and the mainboard, which also needed its own holder. Seeing as the mainboard holders and the small holders were always connected to the top base, it is logical to combine everything into a single piece. The result is the 2025 version shown in Fig. 6. The flower design on the 2024 version was purely aesthetic; however, the space in the 2025 version is a result of avoiding printing unnecessary structure, as the mainboard is stiff enough.

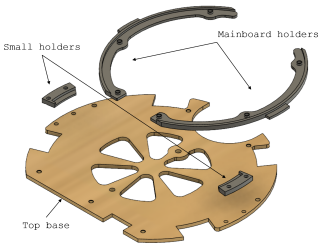


Fig. 5: 2024 version of the Top Base, featuring two holders for the mainboard and two small holders to secure the Top Base to the mainboard.

Fig. 6: 2025 version of the Top Base.

Another redesign involved the structure that held the capacitors. In an effort to create more room inside the robots, the capacitors, which were positioned next to the wheels, were left with no space. This led us to design a holder for the

capacitors using the lateral structures. Now, the capacitors are held up high, not touching the wheels or the motors.

Fig. 7: 2024 version of the Lateral Structure

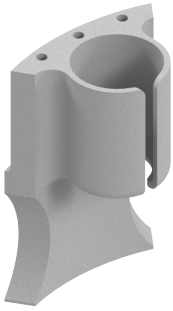


Fig. 8: 2025 version of the Lateral Structure

The front structure also underwent a redesign. During testing, we noticed that if the robot collided directly with another structure, the pillars of our front structure absorbed all the impact energy, consistently breaking at the same point, as seen in Fig 9. We identified that switching the material from PLA to TPU would introduce elasticity, effectively preventing breakage. To validate this, we deliberately subjected the robot to collisions with other structures. The results clearly demonstrated that the TPU pillar exhibited superior performance and a notable improvement in durability compared to the PLA pillar.

Fig. 9: The front structure with a broken pillar due to collision.

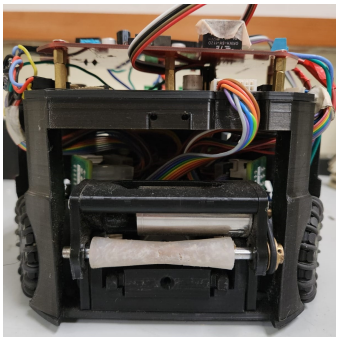


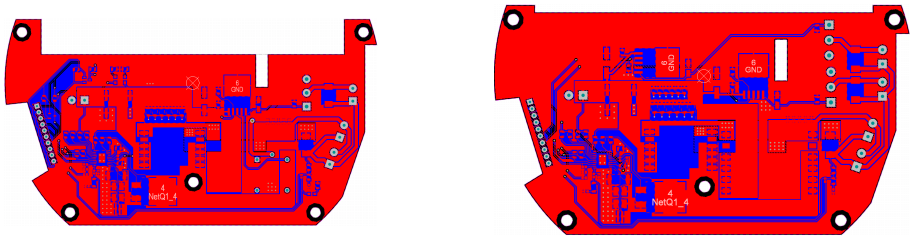
Fig. 10: The front structure made of TPU installed.

2 Hardware

This year, hardware modifications focused mainly on developing the kickboard, which underwent extensive testing to finalize its implementation, the development of this board began as part of the 2023 TDP [5]. In addition, a printed circuit board (PCB) was designed for the base station, which is used to send information to robots.

2.1 Kicker

We conducted a thorough review of the kickboard, and we identified numerous unnecessary and redundant components, as well as several issues present in the first version of the board, which was introduced in the 2023 TDP [5]. Based on these findings, we designed a new board that eliminates these superfluous components and addresses the errors identified in the initial version, the new board schematics can be seen in Fig. 12, and a comparison between the layouts of the current and older version of it in 11. This optimization not only saved space within the robot but also enhanced its overall design and efficiency.



(a) Kicker 2025. (b) Kicker 2022.

Fig. 11: Kickboards, different versions.

The most important improvements are as follows:

- A larger transformer was removed from the circuit board after determining that a smaller transformer could adequately meet the system's requirements. This modification optimizes space utilization while preserving functional performance.
- The capacitor array spanning from C3 to C8 was removed from the circuit board, as these components were deemed redundant for the system's operation. This adjustment simplifies the design and reduces the unnecessary density of components.
- All circuitry related to the discharge of the capacitors through a resistor was removed, as this function is now managed externally.

– Taking advantage of the removal of components, the board's dimensions were adjusted, resulting in a more compact design and improved space utilization.

– The level shifter, responsible for amplifying the signals sent from the mainboard to the kickboard, was integrated into the circuit board. This required the addition of new footprints, 3D models, and libraries corresponding to the newly introduced components.

– A critical design flaw was identified on the kicker board, where the VCC pin of the chip was incorrectly connected to 5V instead of 24V. This issue has been addressed and corrected in the new design.

– Three new LEDs were integrated into the design: one to indicate that the board is powered on, another to signal when the capacitors are charging, and a third to indicate when the capacitors are fully charged.

– The outside layout of the PCB was modified, reducing the overall area usage of the board, allowing this way more clearence for the connection of the debugging cable.

– A relay was added to the PCB design in parallel of the MOSFET in charge of the kick action to add redundancy.

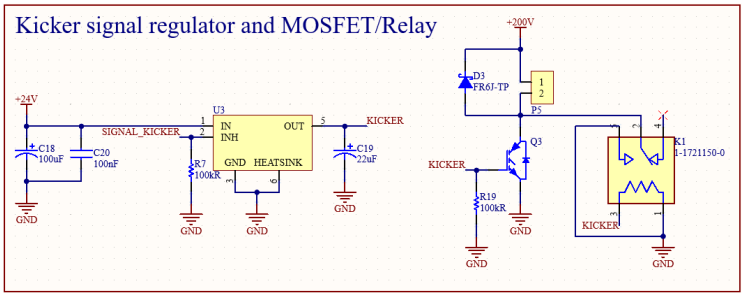


Fig. 13: Kicker activation relay

A relay was integrated into the PCB to replace the previously external switching component. Initially, a MOSFET was used for this function; however, the specific MOSFET selected failed to trigger the pulse reliably. As a result, a relay was adopted as an alternative solution. Since its implementation, the relay has consistently provided reliable performance, which justified its full integration into the current design.

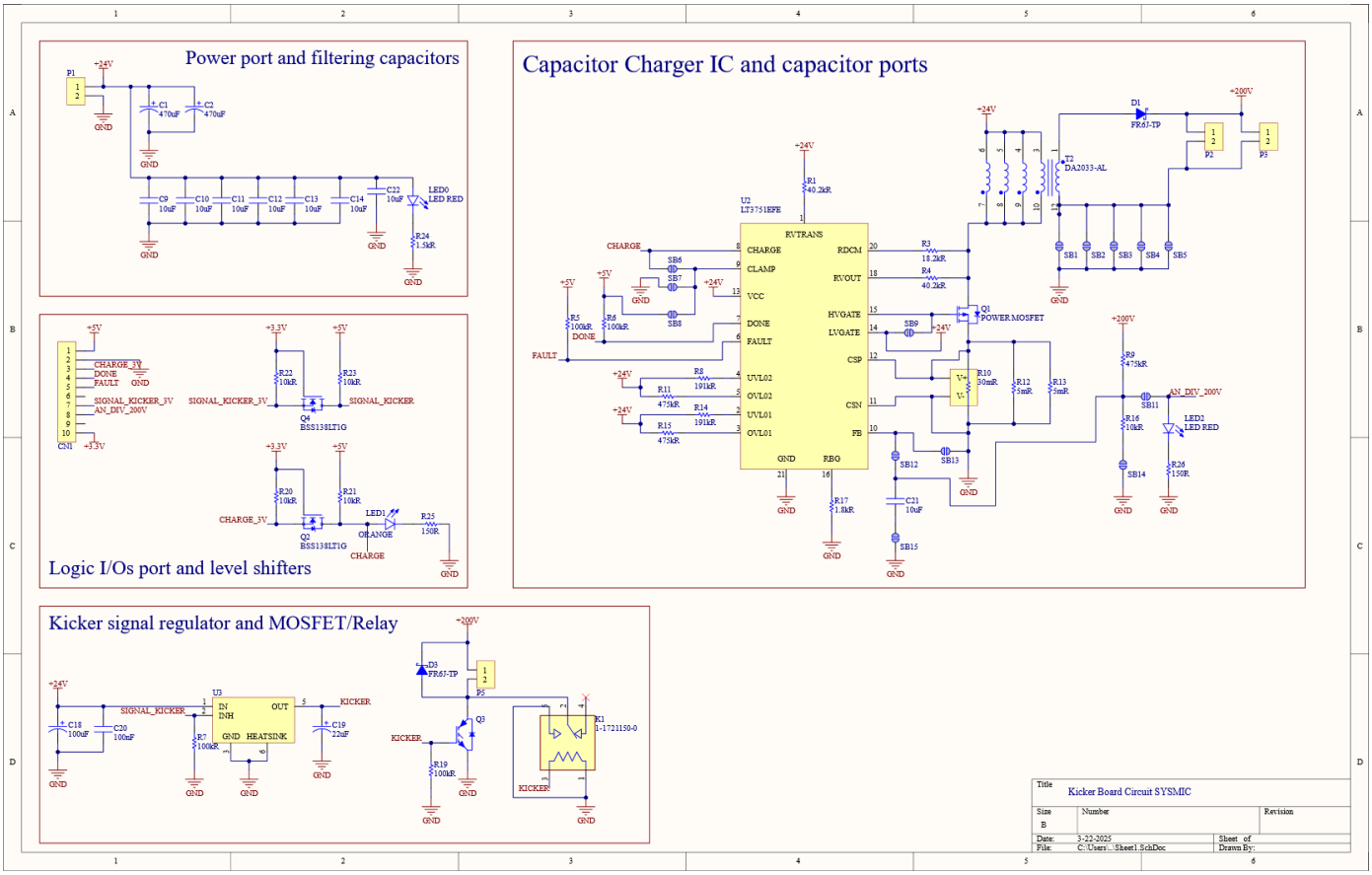


Fig. 12: Kickboard 2025 Schematic

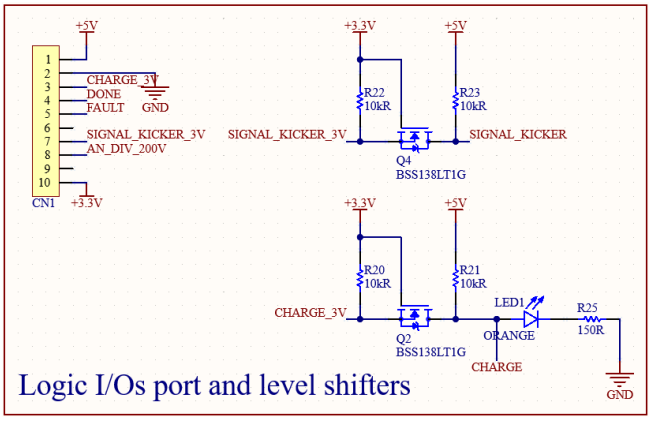


Fig. 14: Logic input and output

Previously, the `CHARGE` signal line was connected directly to the main integrated circuit, which required external voltage level management. To improve integration and reduce dependency on external components, a voltage regulator was incorporated directly into the PCB, with level shifters. This modification simplifies the overall design and enhances reliability. Furthermore, an indicator `LED1` was added to provide a visual signal when the capacitor is charging, aiding in monitoring and diagnostics during operation.

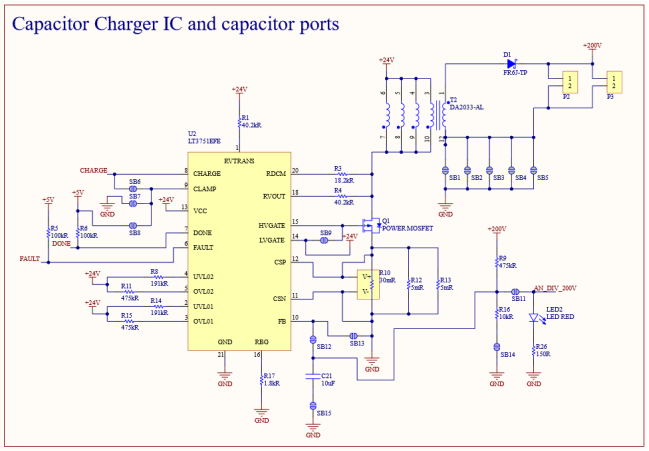


Fig. 15: Main integrated circuit

The same configuration used in 2022 was retained, with the addition of a red `LED2` and its corresponding current-limiting resistor on the `AN_DIV_200V` line. This LED provides a visual indication that the capacitor is charged and ready for operation, thereby facilitating diagnostics and system verification.

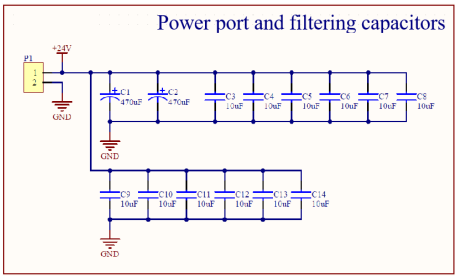


Fig. 16: 2022 power port

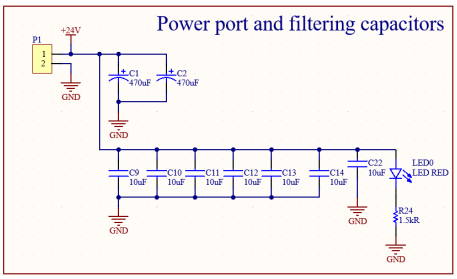


Fig. 17: 2025 power port

The number of decoupling capacitors was reduced because many were redundant and did not significantly improve filtering performance. The remaining capacitors (C1, C2, and a few 10μF units) are sufficient for power stabilization under the system's requirements. This change reduces component count, board space, and cost without affecting functionality.

Additionally, a power-on indicator `LED0` with a current-limiting resistor was added to provide visual confirmation of the 24V supply status, enhancing usability and diagnostics.

2.2 Base Station PCB Development

As part of our communication system, we developed the first version of the Base Station PCB, a key component designed to facilitate wireless data ex- change between the control system and the robots. The board is built around an STM32 microcontroller, which interfaces with both an FTDI module for USB communication and an NRF24L01 radio module via SPI. Below, we present images of the Base Station PCB, showcasing both the top and bottom views of the board, where the key components and their layout can be observed. One of the main improvements in this design was the transition from a pre- perforated board, which was prone to failures due to weak solder joints and inconsistent connections, to a custom-designed PCB. The previous setup occa- sionally failed, causing communication issues that impacted system reliability. With this new PCB, we ensured a more robust and reliable platform. Addition- ally, the modular design of the board allows us to replace faulty components, such as the radio module or the FTDI interface, quickly and efficiently. This flexibility minimizes downtime and simplifies maintenance, making the system more adaptable and resilient in real-world conditions. This PCB represents a significant milestone in the team's development efforts, providing the foundation for a scalable and efficient communication system within the robotics platform.

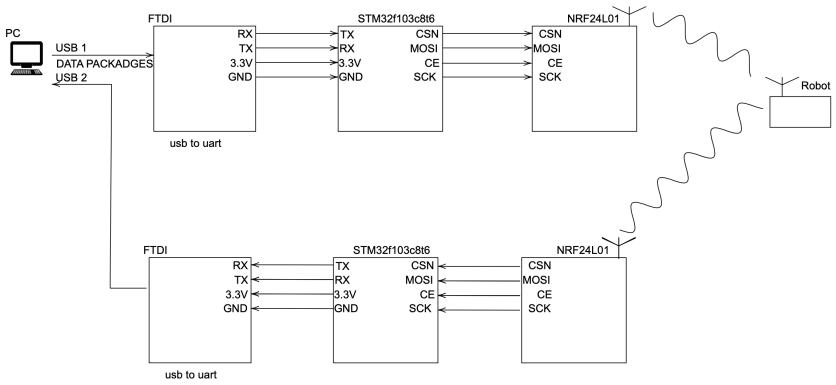
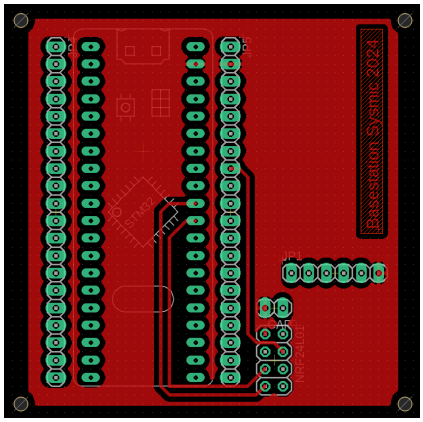


Fig. 18: Schematic diagram of the Base Station PCB, showing connections between the PC and Robots trough the FTDI, STM32 and NRF24L01 modules.

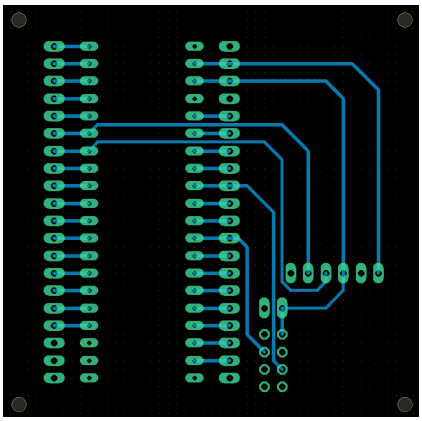
Design Improvements The transition from a pre-perforated board to a custom-designed PCB addressed critical reliability issues, such as weak solder joints and inconsistent connections. The new design features:

– **Modularity**: Faulty components (e.g., radio or FTDI modules) can be replaced without redesigning the entire board.

– **Scalability**: The layout supports future expansions, such as additional sensors or communication interfaces.



(a) Top view of the Base Station PCB.



(b) Bottom view of the Base Station PCB.

Fig. 19: Physical implementation of the Base Station PCB.

This PCB represents a significant milestone in the team's development efforts, providing a robust foundation for scalable and efficient communication within the robotics platform.

Future Hardware Development (2025 and Beyond)

– **Modular Mainboard**: Redesign to isolate functional sections, preventing total system failure due to localized issues.

– **Integrated Battery Monitoring**: Add voltage indicators and alarms directly to the PCB.

– **Rigorous Testing**: Validate the kickboard and base station designs post-manufacturing to ensure compliance with performance standards.

3 Software

In previous years, we participated in competitions using software developed by other teams, specifically RoboJackets' software [4]. While this approach allowed us to compete, it limited our ability to customize and fully understand the underlying mechanics of the system. Therefore, our first major challenge is to develop our own software from scratch. This will provide us with greater control

over the system, enable deeper insights into crucial aspects of the competition such as robot control, strategy, the graphical user interface (GUI), and a more structured and well-documented codebase. A robust codebase will make it easier for new team members to onboard, understand the system, contribute effectively, and become productive more quickly.

We selected Python for the initial prototype due to its flexibility and efficiency in developing. Our initial focus was on establishing fundamental software components essential for developing more advanced functionalities, such as equipping the robot with basic autonomy capabilities—executing commands like moving to specific field position or face target positions. After validating this first prototype, we migrated the logic to C++ for enhanced runtime performance and incorporated Lua as an embedded scripting language for building and managing our high-level strategies.

Software Architecture Our software development was inspired by the architecture of CMDragons [2]. We adopted key features such as modular task division and efficient state management. Figure 20 presents a diagram illustrating system architecture and explained in the followings points:

1. **Vision:** This module receives and processes packets asynchronously from SSL-Vision. It employs a Kalman filter to reduce measurement noise and enhance positional accuracy, ensuring reliable and precise input for other system components. Due to the asynchronous nature of SSL-Vision communication, this module operates in a dedicated single thread.

2. **GameController:** Similarly, the GameController module handles asynchronous communication with the SSL-GameController, interpreting game events and referee commands. It ensures compliance with game rules and enables appropriate responses to changes in the game state. To effectively manage these asynchronous communications, this module also runs in its own dedicated single thread.

3. **World:** The World module receives processed data from the Vision module, including ball and robot positions and orientations. It also computes angular and linear velocities for each robot using the backward difference method. After processing, the World module updates and stores this information as the centralized repository for the current game state.

4. **Control:** This component uses data provided by the World Module to enable robots to act autonomously. Specifically, it manages robot behaviors related to movements through simplified commands like "move to point" or "face to point," which abstract away complex underlying computations. To archieve these commands we employs the following algorithms:
   (a) **Path Planning:** Responsible for generating obstacle-free paths. Initially, we employed the RRT (Rapidly-exploring Random Tree) algorithm. However, the large number of edges in the generated paths caused

issues for the control system. To address this, we adopted the *Fast Path Planning* [6] algorithm, which efficiently resolves this problem by producing smoother paths.

(b) **Motion Control:** Responsible for actually executing movements along planned paths. Initially, trapezoidal velocity profiles were employed, but these caused the robot to pause at intermediate points, leading to non-smooth movement. The team transitioned to *BangBang Control* [3], enabling continuous, fluid movements without unwanted stops. Additionally, we implemented a PID controller specifically for angular movements, greatly improving orientation accuracy (especially when executing commands such as "face to point").

5. **Strategy:** For strategic decision-making, we adopted the Skills, Tactics, and Plays (STP) [2] . This structured, hierarchical framework allows clear, modular, and efficient strategy implementation. The Strategy module integrates directly with the Control module to execute low-level robot actions, referred to as "Skills." All the strategy logic is written in Lua.

6. **Communication Module:** This module is responsible for transmitting the instructions generated by the control module to their respective destinations, whether to the simulator (grSim) or the radio system.

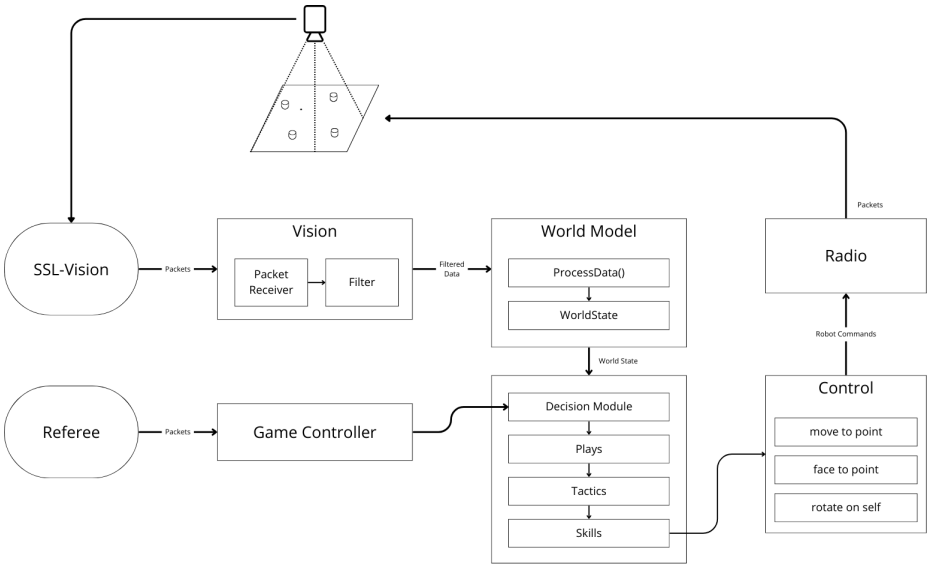


Fig. 20: Sysmic Robotics software architecture

4 Firmware

4.1 Kinematic model

In recent years, we have consistently faced challenges in understanding certain models and design decisions implemented by former team members. These complexities were often difficult to articulate or clarify, even for those familiar with the project, as they appeared in the firmware code as a matrix that '*just works*'. To address this issue, we developed a comprehensive and rigorous description of the robot's kinematic model based on the book *Introduction to Autonomous Robots by Siegwart et al.* [7], specifying the methodology by which the robot calculates the required velocities for each of its wheels, making this topic easier to explain and understand. In this section, we present a summary alongside diagrams.

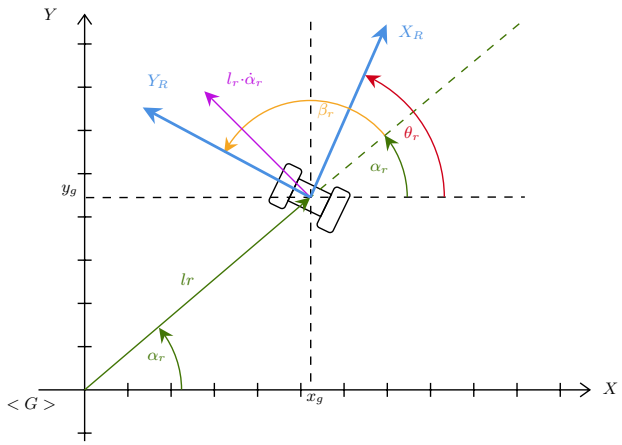


Fig. 21: Differential robot in a global reference

In Figure 21, the playing field is considered the global reference space [X, Y], with a two-wheeled robot positioned within it. Within this framework, a state vector $\dot{\xi}$ is defined, consisting of the global position x_g, y_g and angle θ_r, leading to the following expression for the velocities of interest:

$$\dot{\xi} = \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta}_r \end{bmatrix} \tag{1}$$

however, the velocity state of the robot in the global reference is determined by the robot's velocity in its own coordinates; thus, the following equation is defined for the state vector of the robot $\dot{\xi}_r$.

$$\dot{\xi}_r = \begin{bmatrix} v_{X_R} \\ v_{Y_R} \\ v_{\theta_R} \end{bmatrix} = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} \tag{2}$$

This is affected by each wheel of the robot. A model of one wheel with respect to the robot's center is shown in Fig. 22, and the state vector for the wheel (3), where r is its radius and $\dot{\phi}$ is its angular velocity, is as follows:

$$\dot{\xi}_1 = \begin{bmatrix} v_{X_1} \\ v_{Y_1} \\ v_{\theta_1} \end{bmatrix} = \begin{bmatrix} r_1 \cdot \dot{\phi}_1 \\ v_{Y_1} \\ 0 \end{bmatrix} \tag{3}$$

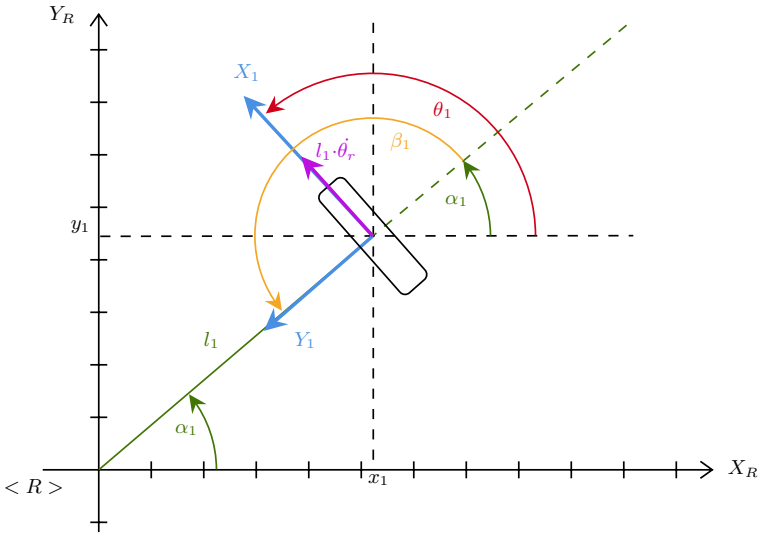


Fig. 22: One wheel model

Then, considering the rotation matrices (4) and its inverse (5):

$$R(\theta) = \begin{bmatrix} cos(\theta) & sen(\theta) & 0 \\ -sen(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$R(\theta)^{-1} = \begin{bmatrix} cos(\theta) & -sen(\theta) & 0 \\ sen(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

we are able to express velocities in any coordinate system, as follows:

$$\dot{\xi} = R(\theta_r)^{-1}\dot{\xi}_r = R(\theta_r)^{-1}R(\theta_1)\dot{\xi}_1 \tag{6}$$

$$R(\theta_r)\dot{\xi} = \dot{\xi}_r = R(\theta_1)\dot{\xi}_1 \tag{7}$$

In Figure 23, a representation is provided where the velocity of each wheel is taken into account as part of the robot at the center of the coordinate system.

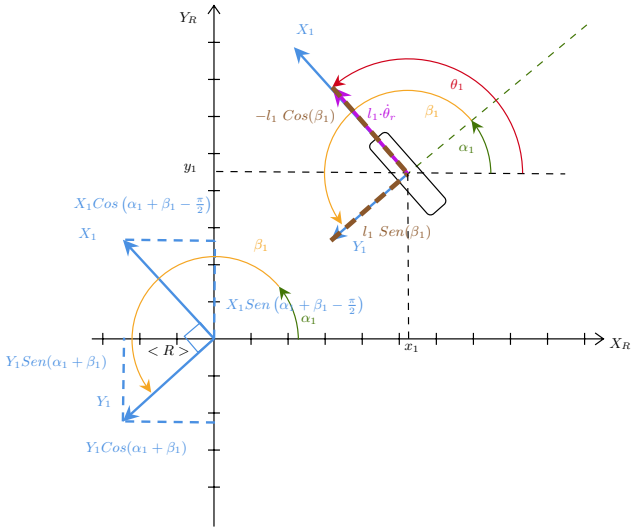


Fig. 23: One wheel transmitted velocity

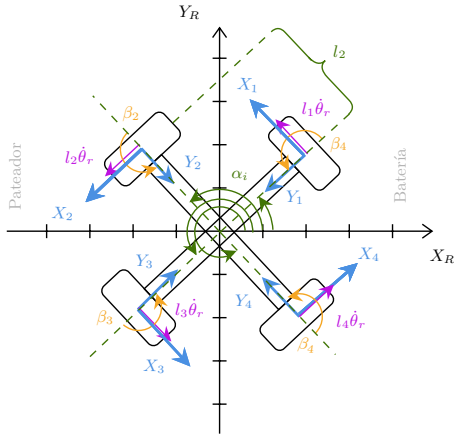


Fig. 24: SSL model

Considering that our SSL model consists of four wheels, a representation of the effect of each wheel being taken into account is shown in Fig. 24.

The final forward kinematic model, considering fixed wheels equidistant from the center of the robot, with the same radius $\beta_i = \pi$, $l_i = l$ and $r_i = r$ with $i \in \{1, 2, 3, 4\}$ is shown in equation 8:

$$\frac{1}{r}\begin{bmatrix} -Sen(\alpha_1)\ Cos(\alpha_1)\ l \\ -Sen(\alpha_2)\ Cos(\alpha_2)\ l \\ -Sen(\alpha_3)\ Cos(\alpha_3)\ l \\ -Sen(\alpha_4)\ Cos(\alpha_4)\ l \end{bmatrix} \dot{\xi}_r = \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} \tag{8}$$

Considering the J matrix defined by 9, simplifies the 8 equation into 10; then the left pseudo-inverse is defined by J^+ as shown in 11.

$$J \triangleq \frac{1}{r}\begin{bmatrix} -sen(\alpha_1)\ cos(\alpha_1)\ l \\ -sen(\alpha_2)\ cos(\alpha_2)\ l \\ -sen(\alpha_3)\ cos(\alpha_3)\ l \\ -sen(\alpha_4)\ cos(\alpha_4)\ l \end{bmatrix} \tag{9}$$

$$J \cdot \dot{\xi}_r = \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} \tag{10}$$

$$J^+ \triangleq (J^T \cdot J)^{-1} \cdot J^T \tag{11}$$

$$J^+ \cdot J = I \tag{12}$$

Leading to the inverse kinematic model 13:

$$\dot{\xi}_r = J^+ \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} \tag{13}$$

References

1. Arriagada, A., Cortes, N., Jimenez, I., Marihuan, G., Moya, D., Richards, I., Tapia, C., Valenzuela, C.: Sysmic Robotics - Team Description Paper 2024 (2024)
2. Browning, B., Bruce, J., Bowling, M., Veloso, M.M.: STP: Skills, Tactics and Plays for Multi-Robot Control in Adversarial Environments (6 2018). https://doi.org/10.1184/R1/6561002.v1, https://kilthub.cmu.edu/articles/journal_contribution/STP_Skills_Tactics_and_Plays_for_Multi-Robot_Control_in_Adversarial_Environments/6561002
3. Purwin, O., D'Andrea, R.: Trajectory generation and control for four wheeled omnidirectional vehicles. Robotics and Autonomous Systems **54**, 13–22 (01 2006). https://doi.org/10.1016/j.robot.2005.10.002
4. RoboJackets: Georgia Tech RoboJackets software for the RoboCup Small Size League (SSL) https://github.com/RoboJackets/robocup-software
5. Rodenas, T., Torres, D., Reyes, P., Peña, J., Tapia, C., Moya, D.: Sysmic Robotics - Team Description Paper 2023 (2023)
6. Rodríguez, S., Rojas, E., Pérez, K., López, J., Quintero, C.A., Calderón, J.M.: Fast path planning algorithm for the robocup small size league. In: Robot Soccer World Cup (2014), https://api.semanticscholar.org/CorpusID:12886519
7. Siegwart, R., Nourbakhsh, I., Scaramuzza, D.: Introduction to Autonomous Mobile Robots, second edition. Intelligent Robotics and Autonomous Agents series, MIT Press (2011), https://books.google.cl/books?id=4of6AQAAQBAJ