RoboUnited 2026
Team Description Paper

Taehun Lee[1], Dohun Kwon[2], Hyojee Jin[1], and Sangyoung Lee[1]

[1] Taejae University, Jongno-gu, Seoul, 03051, Republic of Korea
[2] Kangwon National University, Chuncheon-si, Gangwon-do, 24341, Republic of Korea
`robounitedssl@gmail.com`

Abstract. This paper presents the system design of RoboUnited, a RoboCup Small Size League team targeting RoboCup 2026. The design of the omni-directional robot platform and kicking mechanism emphasizes consistent performance and manufacturability. The motion control integrates inverse kinematics with a hybrid vision architecture to handle real-time uncertainties tracking. Rather than relying on learning-based methods, explicitly structured control and perception pipelines are adopted to ensure interpretability and robustness. On the strategic level, a role-based offensive framework is presented, built around triangular passing networks. The Commitment Inducer (CI) role is also introduced, drawing the defensive reactions and creating spatial opportunities for teammates. This approach manipulates opponent positioning rather than simply reacting to it. The development methodology prioritizes explainable architectures over data-intensive approaches, demonstrating a practical path toward competitive performance in resource-constrained environments. This work provides insights for teams facing similar developmental constraints while pursuing high-level competition in dynamic adversarial scenarios.

1 Introduction

RoboUnited is a RoboCup Small Size League (SSL) team formed by students and graduates from Taejae University, Kangwon National University, the University of Utah, and Seoul National University of Science and Technology. Several team members have prior experience in RoboCup Junior Soccer, which motivated the team to investigate a systematic transition from the Junior League to SSL. Compared to the Junior League, SSL demands significantly higher levels of motion speed, ball handling accuracy, and real-time multi-robot coordination.

In recent years, many SSL teams have adopted learning-based strategies that rely on precise state estimation and extensive experimental data. While effective, such approaches often require substantial development resources and mature hardware platforms. This poses challenges for newly formed teams and those transitioning from RoboCup Junior League Soccer, where control systems are designed for stability and simplicity under limited sensing. Consequently, Junior

League solutions cannot be directly extended to SSL, and a structured redesign is required to bridge this gap.

To address this challenge, RoboUnited adopts a system framework that emphasizes explicit hardware design, model-based control, and rule-based strategy instead of learning-dependent methods. The system employs a three-wheeled omnidirectional platform with inverse kinematics–based motion control and a capacitor-based solenoid kicking mechanism to ensure stable locomotion and consistent kicking performance. At the strategic level, a role-based offensive framework using a triangle passing structure is proposed, maintaining multiple offensive options through predefined conditions. This work presents a practical system design that supports teams transitioning from RoboCup Junior Soccer to SSL.

2 Hardware

2.1 Initial Hardware Setup

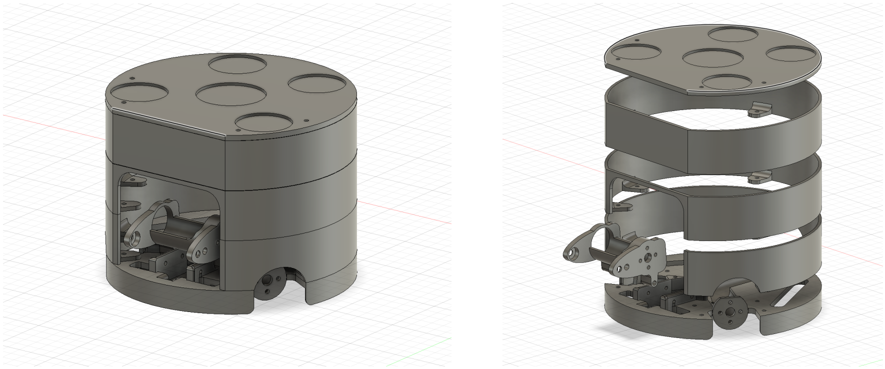
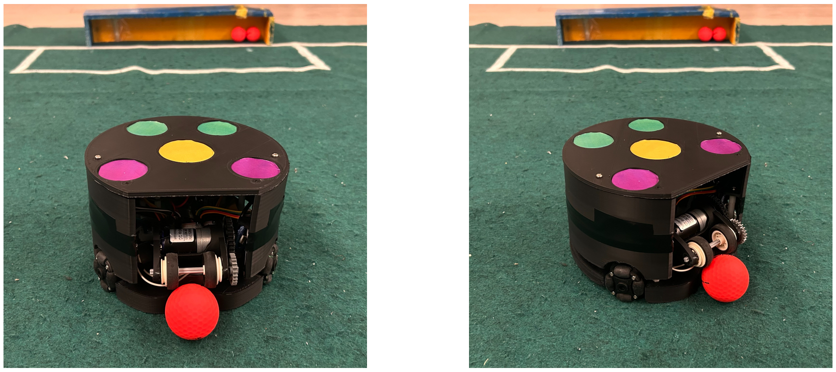

(a) (b)

Fig. 1: Overview of the 3D designed robot: (a) assembled view, (b) exploded view

The hardware structure of the robot is designed with a primary focus on maintaining a low center of gravity to enhance driving stability. The main structural components are fabricated using 3D printing with PLA as the primary material. To balance structural strength and weight reduction, the internal infill density of the 3D-printed parts is set between 30 % and 40 %. The motor control board is mounted with an approximately 12.5 mm clearance from the ground to prevent friction during operation. In addition, the mechanical layout includes a dedicated space for mounting a solenoid-based kicking mechanism.



(a) (b)

Fig. 2: Prototype robot: (a) front view, (b) side view

To ensure stable ball possession, a custom-designed dribbler system is implemented. While many existing RoboCup Small Size League teams adopt a single cylindrical dribbler, the proposed system employs a dual tire-based dribbler configuration to improve ball trapping and manipulation stability. By supporting the ball from both sides, this structure increases the effective contact area and reduces ball slippage during high-speed motion and rapid directional changes. As a result, the robot achieves more reliable ball control, enabling stable offensive execution and smoother transitions during dynamic gameplay.

Table 1: Summary of hardware specifications

Type	**Detail**
Dimension	178*114 (mm)
Weight	1372 (g)
CPU	ESP32 C5
Drivetrain Motor	Chihai motor CHR-GM24-37, 980RPM
Dribbler Motor	Chihai motor CHR-GM24-37, 2000RPM
Motor Driver	BTS7960
Wheel	38mm Omni Wheel
IMU	BNO055
Solenoid	Takaha Kiko, CB1037 10Ω
Kicker Board	Custom Circuit board
On-Board Camera	Pixy 2.1
Battery	LiPo 11.1V 1100mAh

The system employs DC motors for both locomotion and ball manipulation. For robot movement, 980 rpm DC motors are used to achieve stable and controlled driving performance. The dribbler mechanism utilizes a 2000 rpm motor combined with a geared acceleration structure, with a gear ratio of 40:24 to provide sufficient rotational speed and torque for reliable ball handling. Robot heading correction is performed using a BNO055 IMU sensor, enabling accurate orientation estimation. In addition, an on-board vision system is implemented using a Pixy2.1 camera to support visual perception during gameplay.

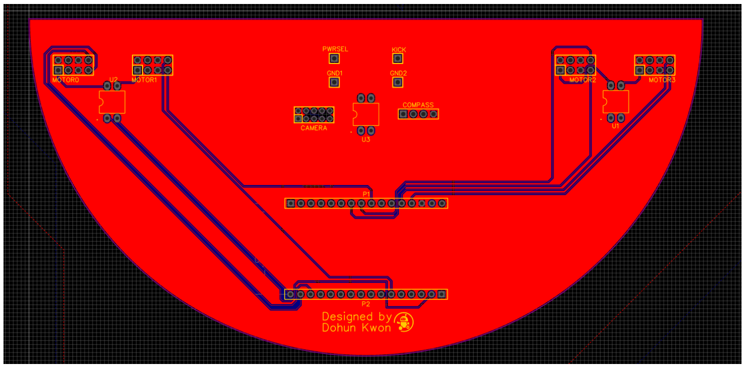
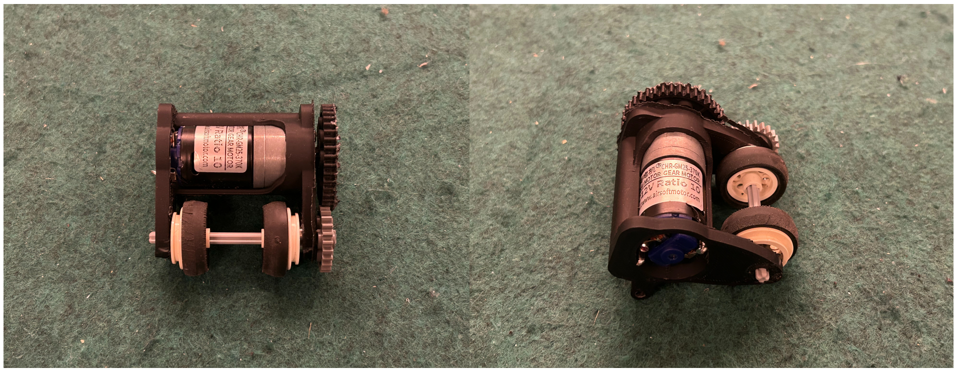


Fig. 3: Motherboard 3D design overview

The dribbler system is designed to achieve stable ball trapping and active manipulation during dynamic game play. The dual tire-based dribbler configuration supports the ball from both sides, increasing the effective contact area and friction compared to conventional single cylindrical dribblers. This structure significantly reduces ball loss during high-speed motion, abrupt direction changes, and immediately after pass reception. In addition, the rotational speed is carefully tuned relative to the robot's translational velocity to prevent excessive backspin and ball bouncing. This design contributes to improved posture stability during dribbling and enables smooth and continuous offensive execution.

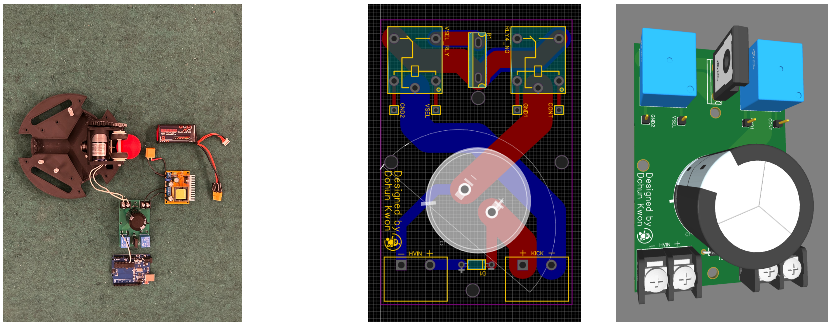


(a) (b)

Fig. 4: Dribbler design: (a) front view, (b) side view

2.2 Design and Optimization of a High-Voltage Solenoid Kicking System

A solenoid drive system requires the supply of instantaneous high current, making stable voltage maintenance and sufficient current availability essential. This study proposes a method to overcome the limitations of the conventional XL6009-based boost circuit by using high-voltage capacitors to enhance solenoid driving performance.



(a) (b) (c)

Fig. 5: Kicking system overview: (a) hardware configuration, (b) PCB layout, and (c) PCB 3D model

2.3 Necessity of Adding Capacitors

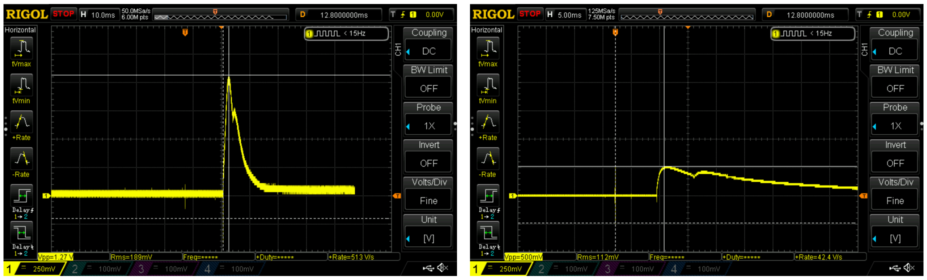
The conventional XL6009 boost circuit could not adequately handle the solenoid load, resulting in voltage drops and insufficient current. To address this, a

390 V boost module was applied and the converted voltage was stored in a capacitor to strengthen the ability to supply instantaneous current. Using the charge/discharge characteristics of the capacitor, a stronger driving force was achieved compared to that of the existing circuit.

2.4 Output Strength Control Design

Controlling output strength is essential depending on strategic requirements. In this study, a 50 Ω resistor was inserted into the discharge path to limit the current flowing into the solenoid, allowing for output strength control. In addition, a relay was incorporated to allow reconfiguration of the circuit, making it possible to control the strength in various operating modes.

Oscilloscope measurements confirmed the effectiveness of this approach. When the discharge path did not include the 50 Ω resistor, the solenoid current rose sharply, producing a large current peak, as shown in Fig. 5(a). In contrast, when the resistor was inserted, the current peak was limited, resulting in a more stable discharge profile, as illustrated in Fig. 5(b). A direct comparison of the two waveforms in Fig. 5(c) clearly demonstrates that resistor insertion enables effective control of the solenoid driving strength. solenoid driving force.



(a) (b)

Fig. 6: Discharge current waveform: (a) without 50 Ω resistor, (b) with 50 Ω resistor

2.5 Basis for Capacitor Specification Selection

The maximum output voltage of the boost module is 390 V. To ensure stability, a capacitor rated at 400 V (330 μF) was selected. This provides a breakdown voltage 10 V higher than the module's maximum output, ensuring safety. Discharge tests showed that compared to a 200 V 56 μF capacitor, a capacitor capable of applying higher voltage demonstrated superior performance. Therefore, this study concludes that voltage characteristics have a greater impact on performance than capacitance. Based on this finding, a cost-effective 400 V 330 μF capacitor was ultimately adopted.

2.6 Adoption of Relay-Based Driving Method

In kicker circuits operating at high voltage (390 V), MOSFETs offer fast response speed. However, power MOSFETs capable of withstanding such voltage are expensive and have poor direct compatibility with MCUs. For this reason, this study adopted a relay-based driving method. Relays are relatively inexpensive, operate reliably in high-voltage environments, and experimental results confirmed that their performance exceeded expectations relative to cost.

3 Software

3.1 Initial Software Setup

Table 2: Summary of software specification

Type	**Detail**
WIFI Communication	ESPNOW
Languages	C, Python
Simulator Communication	UDP Multicast
Vision System	Custom Vision System with Pixy 2.1 Camera

The software environment of RoboUnited is designed to support real-time communication, modular development, and system-level extensibility. Wireless communication between robots is implemented using the ESP-NOW protocol, enabling low-latency and peer-to-peer data exchange without reliance on a centralized network infrastructure. The system is primarily developed using C for embedded control and low-level hardware interaction, while Python is utilized for higher-level tasks such as strategy logic, simulation interfacing, and debugging. Communication with external simulators and vision systems is achieved through UDP multicast, allowing efficient broadcast of shared state information to multiple agents. For visual perception, a customized vision system incorporating a Pixy2.1 camera is employed to provide on-board object detection under constrained sensing conditions.

ESP-Now ESP-Now is a wireless communication protocol developed by Espressif. In this research, ESP32-C5 was implemented as a CPU. It provides direct, fast, and lightweight communication through low-power data transmission. It operates Wi-Fi and Bluetooth LE without a router. The ESP-NOW model compresses five layers of the OSI model, which eliminates the need to transfer data via application, presentation, session, transport, and network layers and it leads to a quick response without long delay. Therefore, it qualifies to use in RoboCup SSL requires real-time signal transmission.[2]

3.2 Inverse Kinematics on an Omnidirectional Structure

While multiple RoboCup Small Size League teams employ four-wheeled omnidirectional robots, RoboUnited adopted a three-wheels configuration. Reflecting on the experience of applying four-wheeled robots in the RoboCup Junior League, instability issues on immediate velocity reduction after high-speed motion were repeatedly observed. To improve the limitations on 4-wheels structure, the robots are made with a three-wheel structure. The three-wheel configuration provides greater postural stability during rapid acceleration and deceleration. Three omniwheels are placed at 120-degree intervals around the center of the robot. This configuration enables the robot to perform translational motion in any direction while simultaneously rotating, with the overall movement direction and angular velocity determined by the combined velocities of the individual wheels. The inverse matrix calculation process is as follows [1]:

$$\begin{bmatrix} a_x \\ a_y \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\alpha_1 & \cos\alpha_2 & \cos\alpha_3 \\ \sin\alpha_1 & \sin\alpha_2 & \sin\alpha_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}, \tag{1}$$

where a_x and a_y denote the desired linear velocities in the x and y directions, respectively, ω represents the desired angular velocity, and F_1, F_2, F_3 are the wheel velocitie commands. The angles $\alpha_1, \alpha_2, \alpha_3$ indicate the orientation of each wheel with respect to the robot frame.

To obtain the wheel velocities from the desired robot motion, the inverse matrix A^{-1} is used:

$$A^{-1} = \frac{1}{\det(A)} \operatorname{adj}(A) = \begin{bmatrix} -0.333 & 0.578 & 0.333 \\ 0.667 & 0 & 0.333 \\ -0.333 & -0.578 & 0.333 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} -0.333 & 0.578 & 0.333 \\ 0.667 & 0 & 0.333 \\ -0.333 & -0.578 & 0.333 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ \omega \end{bmatrix} \tag{3}$$

Thus, the final inverse kinematics equations for each wheel are expressed as:

$$\begin{gathered} \mathrm{F}_1 = -0.333\, a_x + 0.578\, a_y + 0.333\, \omega, \\ \mathrm{F}_2 = 0.667\, a_x + 0\, a_y + 0.333\, \omega, \\ \mathrm{F}_3 = -0.333\, a_x - 0.578\, a_y + 0.333\, \omega. \end{gathered}$$

For omnidirectional robots, directly controlling each wheel independently does not guarantee accurate movement. Therefore, RoboUnited adopted an inverse kinematics model that first defines the robot's desired motion and then converts it into the velocity of each wheel. The target motion of the robot is represented by three components: the x-axis and y-axis translational components, and the rotational component ω. These three components are transformed into individual wheel velocities through a 3×3 inverse kinematics matrix.

3.3 Linear Distance-Based Kick Velocity Control Model

The kick velocity is defined as a linear function of the distance between the ball and the target robot:

$$v_{kick} = a \cdot d + b$$

In the above equation, the distance between the ball and the target robot is defined as the Euclidean distance between the two points, while the distance scaling coefficient and the minimum kick offset are defined as the variables ‘a’ and ‘b,’ respectively.

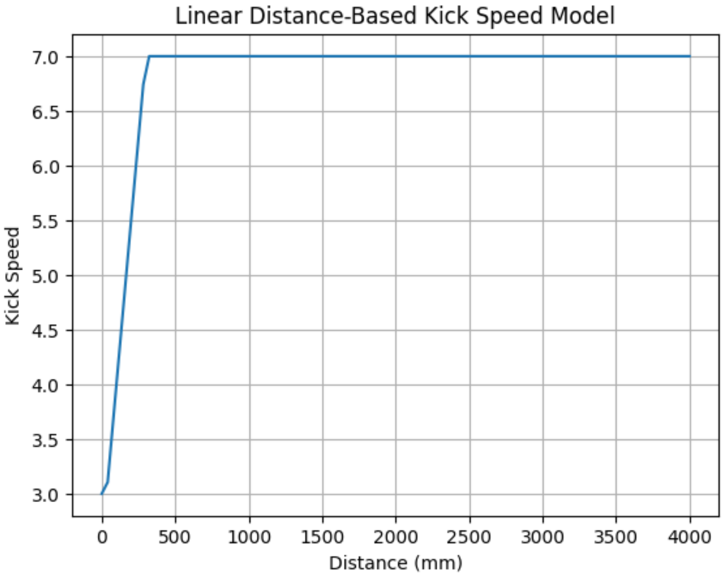


Fig. 7: Linear Distance-Based Kick Speed Model

Figure 7 shows the relationship between the distance and the kick velocity according to the proposed linear distance-based kick velocity control model. As shown in the figure, the kick velocity increases proportionally with the distance.

Theoretically, the required initial velocity of the ball should be proportional to the square root of the distance between the ball and the target robot because of the deceleration caused by friction forces. However, there are some discrepancies in the grSim environment compared to the ideal state. Specifically, the omniwheel-based robot’s drive mechanism generates anisotropic friction forces according to the direction of the robot’s movement, and non-linear dynamics are generated during the process of collision between the ball and the robot.

Moreover, the ball is not in the state of rest immediately before the robot kicks the ball due to the continuous state of contact between the ball and the dribbler, the non-linear dynamics generated by the omniwheel-based robot's mechanism, and the possible errors in the robot's controller.

Therefore, this study concluded that the application of the theoretical physical model might impair the control's performance due to the above factors. Hence, the control's stability and usability were prioritized over the precise reproduction of the physics. Therefore, the linear model was chosen.

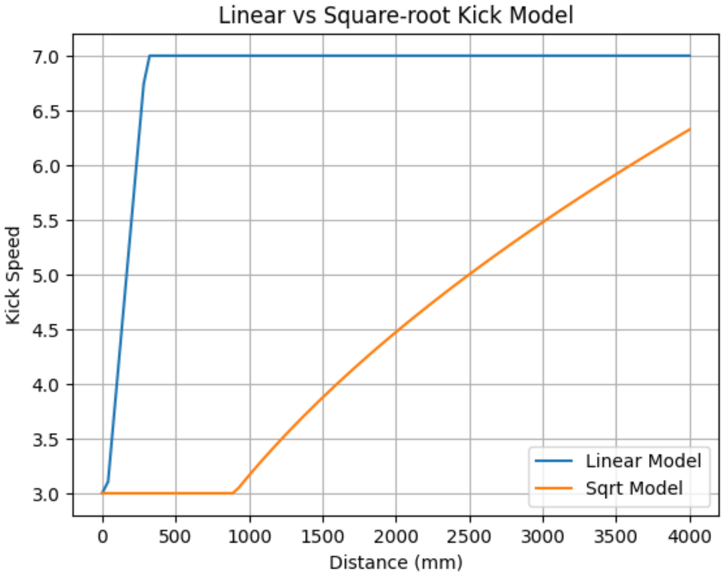


Fig. 8: Linear vs Square-root kick Model

Figure 8 illustrates the linear model's comparison with the square-root-based model. The latter demonstrates high sensitivity at small distances. However, this sensitivity gradually decreases with increasing distance. Conversely, the linear model maintains constant sensitivity over the entire range.

Since the velocity changes linearly with the distance in the linear model, the output is linear and thus predictable. The control's reaction is consistent across all ranges. In the presence of noise in the measured distance, the velocity's change is linearly constrained. Therefore, the output's fluctuations are constrained. Thus, the control's stability is very high. Conversely, the square-root-based model demonstrates reduced sensitivity with increasing distance. Thus, it might react too sensitively to noise at small distances and insufficiently at long distances. Therefore, the control's stability is low in this case.

The velocity of the kick was constrained to fall in the range of the minimum and maximum velocities. The constraint was implemented in the following way:

$$v_{kick} = \text{clip}(a \cdot d + b, v_{min}, v_{max})$$

In the above equation, v_{min} and v_{max} were set to 3.0 and 7.0, respectively. The adopted linear model is appropriate for real-time control due to the low computational cost, enabling the tuning of the model for experiments, thus ensuring robustness to various changes in the environment. Therefore, despite the limitations of the model in the explicit representation of non-linear factors such as friction, rotation, and collision, the model can be considered an efficient solution in terms of system performance.

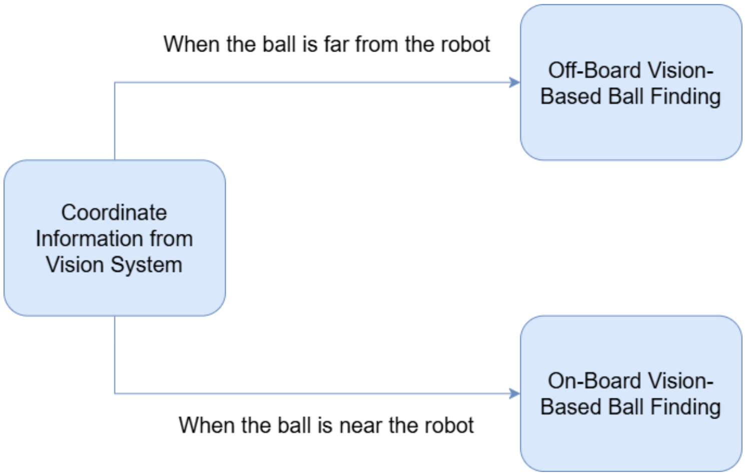


Fig. 9: On-Board Vision System Overview

4 Strategy

In the RoboCup Small Size League (SSL), offensive strategies have gradually shifted from executing a single optimal action toward maintaining multiple tactical options while inducing and exploiting opponent defensive responses [4, 6, 7]. Recent studies emphasize the quantitative evaluation of pass success probability [5], opponent-aware planning that explicitly incorporates defensive behavior [6], and the application of team sports tactical principles to robotic soccer[8].

In this study, the offensive strategy is designed to avoid reliance on learning-based models and instead adopts an explicitly structured, role-based cooperative framework. The proposed approach integrates a triangle-based passing network with a *Commitment Inducer* role, aiming to preserve multiple attacking alternatives while deliberately influencing the timing of opponent defensive commitment. The overall objective is to construct an offensive framework that remains robust, explainable, and executable under constrained experimental and system conditions.

4.1 Related Work

Cooperative Offensive Structures and Passing Networks Cooperative offensive structures involving multiple robots have become a fundamental component of SSL gameplay [4]. Several studies have demonstrated that quantifying pass selection based on distance, interception risk, and the feasibility of subsequent actions contributes to more stable offensive progression [5, 7]. The CM-Dragons team introduced zone-based coordination, enabling role distribution and improved ball possession stability through region-based field control [9].

Opponent-Driven Offensive Planning Biswas et al. [6] proposed opponent-driven planning strategies that intentionally induce defensive reactions and exploit the resulting spatial changes. This perspective frames offensive behavior as an interactive process that incorporates opponent perception and decision-making, rather than treating defense as a static constraint.

Play-Level Tactics and Team Sports Principles Structuring strategies at the level of plays, rather than individual robot actions, has been shown to improve tactical extensibility and reusability [8]. In addition, research applying tactical principles from human team sports such as space creation, numerical advantage, and timing disruption provides a theoretical foundation for offensive strategy design in robotic soccer [10, 6, 11].

5 Overview of the Offensive Strategy

The offensive strategy is composed of three core modules:

1. **Triangle-based passing network positioning**
2. **Opponent perception and timing disruption through the CI**
3. **Utility-based role assignment with integrated control**

Each module is designed independently, but they are dynamically combined during gameplay according to ball possession state, opponent defensive density, and passing network stability. This modular structure allows the strategy to maintain multiple offensive scenarios rather than converging to a single fixed attack pattern.

5.1 Triangle Passing Network and Dynamic Formation

The five field robots, excluding the goalkeeper, form a pentagon-based formation, while each robot is positioned to participate simultaneously in multiple triangular passing structures. This configuration ensures that the ball-holding robot

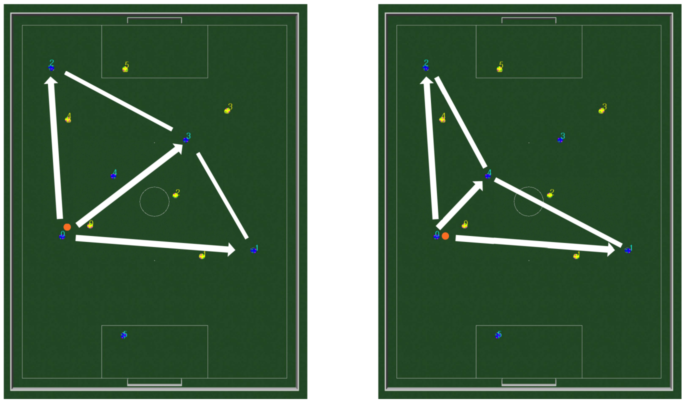


Fig. 10: Triangle Pass Route Decision Making Examples

maintains at least two valid passing options at all times, allowing immediate switching when a specific passing route is blocked.

This structure is inspired by the zone-based coordination approach of CMDragons cmdragons2015. However, instead of maintaining static field zones, dynamic repositioning is allowed based on passable angles and opponent defensive positioning. The maintenance of the triangular passing network is determined by pass safety. Pass safety is evaluated by jointly considering the passing angle, passing distance and required time, and the intercept potential of opponent robots.

The following control variables are defined during pass execution. Robots perform passes from a stationary state, and the ball is assumed to move in a straight line with a certain initial velocity. Since the initial velocity of the ball may vary depending on the game situation, it is defined as a variable parameter n, rather than a fixed constant. The change in the ball's velocity is approximated using a uniformly accelerated linear motion model. For time $T > 0$, the velocity–time relationship is expressed as follows:

$$V_{\text{ball}}(T) = -T^2 + n \tag{4}$$

This formulation represents a simplified model in which the ball starts with an initial velocity n and decelerates over time. The corresponding displacement–time relationship is derived from the equations of uniformly accelerated motion, enabling a connection between the passing distance and the time required for the pass.

The motion of opponent robots is modeled as starting from a stationary state and accelerating over time. The velocity of an opponent robot increases as a function of time and is represented by the following velocity–time relationship:

$$V_{\text{opp}}(T) = T^2 \tag{5}$$

By integrating the velocity–time relationship, the maximum displacement that an opponent robot can reach during the pass duration is estimated. This displacement is used as the radius of the opponent's interceptable region. The passing distance between two robots is defined with their position coordinates (X_1, Y_1) and (X_2, Y_2) as follows:

$$S = K_p\left((X_2 - X_1)^2 + (Y_2 - Y_1)^2\right) \tag{6}$$

Here, K_p denotes a proportional constant used to account for control errors, ball friction, and directional alignment errors that occur in real game environments.

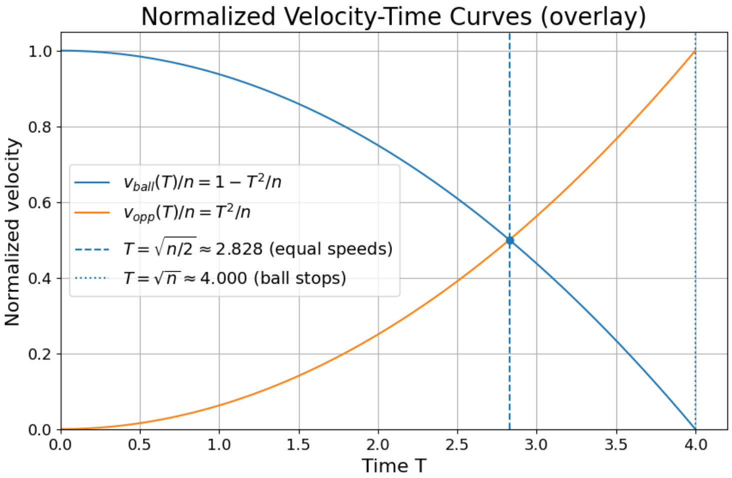


Fig. 11: Normalized Velocity-Time Curves

Pass safety evaluation is performed based on the relationship between the region reachable by opponent robots during the ball's travel time and the passing path itself. The reachable region of an opponent robot is approximated as a circular area, while the passing path is modeled as a straight corridor with a finite width. The degree of overlap between these two regions is defined as the *intercept risk*. If the risk remains below a predefined threshold, the pass is classified as safe.

Based on this safety evaluation, the connectivity of the triangular passing network is periodically re-evaluated. If the possibility of a pass is reduced, robots are repositioned to more conservative locations to increase the robustness of the passing network.

5.2 Commitment Inducer-Based Perception and Timing Disruption

Concept of Commitment Inducer The Commitment Inducer (CI) is defined as a role-based behavior designed to induce early movement or defensive switching by opponent robots. This is achieved through actions that are easily perceived as threats, such as forward acceleration without executing an actual shot or breakthrough attempt. Rather than attempting to predict opponent responses through explicit models, the CI follows an opponent-driven planning approach that focuses on inducing observable reactions and exploiting them once they occur biswas2014. In this sense, the CI functions as a perceptual trigger rather than a direct attacking action.

CI Activation Conditions The CI is activated only when all of the following conditions are satisfied:

Ball State When the ball enters the opponent half of the field, defined as the half containing the opponent goal, the situation is considered likely to be perceived as an attacking phase by the opponent defense.

Opponent Defensive Response When at least two opponent robots, particularly defenders, begin to reorient toward their own goal or initiate forward movement, the defensive system is interpreted as entering a commitment-ready state.

Availability of Passing Alternatives At least one stable triangular passing network must be maintained. This condition ensures that alternative passing options remain available both during and immediately after CI execution.

Success Criteria The CI is considered successful if at least one of the following conditions is met:

Space Creation The free space around the intended pass-receiving robot increases compared to the state at the moment of CI initiation.

Defensive Commitment At least one opponent robot exhibits a clear directional change toward the CI robot, with the distance between them decreasing continuously over a short time interval.

Offensive Transition Within three passes after CI termination, the play results in a stable pass, valid shot, or goal.

Failure Detection and Termination Conditions The CI is immediately terminated when one of the following conditions occurs:

Passing Network Collapse If the triangular passing structure becomes unstable or collapses during CI execution, the CI is aborted, and the strategy transitions to a stabilization mode.

Timeout If more than three opponent robots do not respond even though the position holding state persists, the CI is judged ineffective, and the strategy switches to a pass-oriented stabilization mode.

Excessive Pressure If the ball-holding robot is simultaneously pressured by two or more opponent robots, defined as both opponents encountering the ball holder while reducing distance, the strategy immediately performs ball circulation passing to relieve pressure.

5.3 Role Assignment and Integrated Control Architecture

The five field robots are dynamically assigned roles such as Attacker, Commitment Inducer, Supporter, Defender, and None. Role assignment and transitions follow a utility-based strategy that considers distance to the ball, relative positioning of opponents, and current passing network connectivity [8, 4]. Through this structure, the high-level strategy module provides explicit target positions and action types to the lower-level path planning and control layers. When path generation fails or collision risk increases, the strategy layer immediately transitions to a conservative mode, prioritizing stability over aggressive progression.

6 Conclusion

The software system proposed in this study is designed to be accessible to teams entering the RoboCup Small Size League (SSL), particularly those transitioning from RoboCup Junior soccer, rather than relying on high-performance computing environments or complex external infrastructures. The control framework, based on inverse kinematics for an omnidirectional drive and sensor-based feedback control, enables stable motion using relatively simple computations, reflecting continuity with control strategies commonly employed in Junior competitions. In addition, an onboard camera system is utilized to enhance ball approach accuracy in close-range situations, compensating for the limitations of conventional vision systems in near-field perception. This software architecture reduces system complexity while allowing incremental expansion, thereby providing a practical and scalable guideline for beginner SSL teams and teams migrating from RoboCup Junior to SSL.

Despite the effectiveness of the proposed system, several limitations remain. The absence of motor encoders reduces the accuracy of path planning and limits precise motion estimation, particularly during high-speed maneuvers. In addition, the restricted field of view of the onboard camera weakens its integration with the off-board vision system, resulting in reduced robustness in dynamic game situations. As future work toward RoboCup 2026, the vision framework will be upgraded from the current Pixy camera based system to the official SSL Vision to achieve more reliable global perception. Furthermore, alternative Wi-Fi communication methods beyond ESP-NOW will be explored to improve data throughput and system scalability. These improvements aim to enhance both performance and compatibility with standard SSL environments while maintaining accessibility for emerging teams.

References

1. Lee, T.: Sensor-Based Feedback Control in Omnidirectional Systems. National High School Journal of Science (2025)
2. Espressif Systems: ESP-NOW wireless communication protocol. `https://www.espressif.com/en/solutions/low-power-solutions/esp-now`, Last accessed 2026 31 March 2026
3. Huang, Z., Wang, Y., Chen, L., Li, J., Chen, Z., Xiong, R.: Mechatronic design of a dribbling system for RoboCup small size robot. arXiv preprint arXiv:1903.9934v1 (2019)
4. Behnke, S., Stückler, J., Schreiber, M.: Design and implementation of dynamic offensive strategies in the RoboCup Small Size League. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2014: Robot World Cup XVIII, pp. 257–268. Springer (2015)
5. Leipscher, U.: Rework of the coordination of offensive actions in the RoboCup Small Size League. Study Report, Baden-Wuerttemberg Cooperative State University Mannheim (2018)
6. Biswas, J., Mendoza, J.P., Zhu, D., Choi, B., Klee, S., Veloso, M.: Opponent-driven planning and execution for pass, attack, and defense in a multi-robot soccer team. In: Lomuscio, A., Scerri, P., Bazzan, A., Huhns, M. (eds.) Proc. of the 13th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2014), pp. 493–500. IFAAMAS (2014)
7. de Koning, L., Mendoza, J.P., Veloso, M., van de Molengraft, R.: Skills, tactics and plays for distributed multi-robot control in adversarial environments. In: RoboCup 2017 Symposium Proceedings (2017)
8. Chen, H.L., Beumer, R., Torta, E., Bruyninckx, H., van de Molengraft, R.: Designing offensive robot soccer strategies from principles in invasion team sports. *J. Intell. Robot. Syst.* **112**(2) (2026). https://doi.org/10.1007/s10846-025-02334-0
9. CMDragons: CMDragons 2015 extended team description paper. RoboCup Small Size League (2015)
10. Nakanishi, R., Bruce, J., Murakami, K., Naruse, T., Veloso, M.: Cooperative 3-robot passing and shooting in the RoboCup Small Size League. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006: Robot Soccer World Cup X. LNCS, vol. 4434, pp. 459–470. Springer (2007) https://doi.org/10.1007/978-3-540-74024-7_41
11. Tech United Eindhoven: STP framework for multi-robot soccer. Technical Report (n.d.)
12. Baez, S.: Predicting opponent team activity in a RoboCup environment. arXiv preprint arXiv:1503.01446 (2015). Available at: `https://arxiv.org/abs/1503.01446`