Ai Rams - ITESM

Team Description for RoboCup SPL 2025

C. Jaramillo, S. Diener, R. Silva, R. Cisneros, J. Palacios, A. Lucio, D. Sánchez

School of Engineering and Science

Tecnológico de Monterrey Campus Aguascalientes

Avenida Eugenio Garza Sada 1500, 20328 Aguascalientes, Ags.

https://tec.mx/es/aguascalientes

Abstract. This paper holds all the information for the description of the team and the contributions to the program for the RoboCup competition in the Standard Platform League 2025 in Brazil.

Keywords: Strategy Behavior, Decision Making, Statistical Evaluation, RoboCup.

1. Team Information

Team name: Ai Rams

Website: https://ai-rams.github.io/

Email: airams.tec@gmail.com

AiRams is a team formed by students from the Tecnológico of Monterrey Campus Aguascalientes, created by Juan Manuel Campos Sandoval and Julián Mauricio Echeverry Mejía. With the participation of students from 1st and 5th semester of Mechatronic engineering and Computational Sciences.

This year will be the first participation of the team in the RoboCup competition.

Team leaders: Juan Manuel Campos Sandoval**,** Julián Mauricio Echeverry Mejía.

Students in Mechatronics: Sergio Diener Márquez**,** Carlos Javier Jaramillo López**,** Rogelio Silva Galindo**,** Janny Vianey Palacios Prieto**,** Ammy Skarlethe Lucio Reyes.

Students in Computer Science: Diego Sánchez Pámanes, José Rodrigo Cisneros Murillo.

Country: México.

University: Instituto Tecnológico de Estudios Superiores de Monterrey, Campus Aguascalientes

NAO: We already own five robots, and we are aiming to buy two more.

2. Code Usage

We are using the B-human code release from 2024, implementing our own improvements on strategy behavior and decision-making in game situations.

We are using the simulator developed by B-HUMAN SimRobot to test the changes we make in the code to see how the robots behave and interact with two teams actively.

3. Own Contribution

During our analysis of the B-Human's base code, we chose to work on *Ball Detection* and *Behavior Architecture*, which are defined in the original B-Human code as a part of the *Dribble Engine*. We worked on five significant contributions that together significantly improve the strategy and decision making of the Robots. We are in the process of measuring the impact of these changes in the game and look forward to publishing the results in the scientific journal *IEEE Robotics and Automation Letters (RA-L)* or the *International Journal of Humanoid Robotics (IJHR)*.

1. Time to Reach the Ball Estimation

The behavior control relies on time calculations made by the robot to approximate its position to the ball or other robots. These calculations are based on taking the longest distance in a single component (either x or y). As the components approach equal values, the error in time calculations increases.

To measure the impact of this change, we simulate game situations by positioning two robots in the same location, one with the advantage of gaining ball possession. By varying the distances to the ball in the presence of opposing team robots, we can observe the robots' ability to make optimal decisions. We are testing this change statistically to analyze the real improvement it makes to our code and to verify our initial measurements and evaluations. We believe that the results we look forward to obtaining will confirm the impact of the proposed improvement.

Previous Model:

The original method computed the time to reach the ball using a component-wise approach. Specifically, it determined the time by taking the maximum of the time estimates computed separately for the x and y components of the ball's position:

$$t_{\text{reach}} = clamp\left(max\left(\frac{|x|}{v_{\text{max},x}}, \frac{|y|}{v_{\text{max},y}}\right), 1, 2\right)$$

In B-human's original code the positions and speeds $x,\ y$, $v_{\text{max},x}$ and $v_{\text{max},y}$ were previously defined. This method assumes that the motion along each axis is independent, which may not accurately represent the overall displacement when the ball is not aligned with one of the principal axes.

New Model:

We proposed a refined approach based on the Euclidean distance between the robot and the ball. Instead of handling each component separately, we compute:

$$t_{\text{reach}} = clamp\left(\frac{|p_{\text{ball}}|}{v_{\text{avg}}}, 1, 2\right)$$

$$|p_{ball}| = \sqrt{x^2 + y^2}$$

where the average speed is given by:

$$v_{avg} = \frac{v_{max,x} + v_{max,y}}{2}$$

By using the Euclidean norm $|p_{\text{ball}}|$ to measure the straight-line distance from the robot to the ball, this method captures the true magnitude of displacement irrespective of the direction. The average speed v_{avg} serves as a more representative value of the robot's overall translational capability. Clamping the result between 1 and 2 seconds ensures that extreme values are avoided, which is crucial for maintaining predictable behavior. This change leads to a more realistic and robust time estimation, particularly in scenarios where the ball's position lies along a diagonal or off-axis direction, thus enhancing the overall responsiveness of the robot's motion planning. This ensures that the estimation considers both axes distances instead only of the maximum, also considering the average speed related to both axes.

2. Velocity Factor Interpolation for Ball Propagation

To validate the effectiveness of our proposed velocity interpolation model, we have conducted initial tests focusing on the smoothness of the estimated velocity and the robot's reaction time, while additional real-game scenario evaluations are still pending. Our simulations show that the sigmoid model significantly reduces abrupt changes in velocity estimation compared to the linear approach, with estimated variations decreasing from 30-50% to 10-25%. Additionally, reaction times are more stable and, on average, 0.1 to 0.3 seconds faster than the linear model, allowing the robot to adjust more efficiently to ball movement. As the next step, we will conduct real and simulated game tests to further assess whether this improved velocity estimation enhances decision-making and ball interception efficiency. These results provide strong quantitative evidence of the advantages of our approach over the B-Human model, supporting its potential impact in RoboCup competitions.

Previous Model:

The earlier approach applied a simple linear interpolation to adjust the velocity used in ball propagation. The factor "f" was computed as:

$$f = mapToRange(|p_{\text{perceived}}|, d_{\text{min}}, d_{\text{max}}, 0, 1)$$

and the effective velocity was then calculated by:

$$v_{\text{used}} = (1 - f) \cdot \min(v_{\text{ball}}, v_{\text{close}}) + f \cdot v_{\text{ball}}$$

This linear model provided a straightforward blend between a capped velocity and the actual ball velocity, depending on the distance from the ball. All of these parameters are originally defined by B-Human.

New Model:

To achieve a smoother and more responsive transition, we introduce a logistic (sigmoid) function:

$$f_{\text{new}} = \frac{1}{1 + \mathrm{e}^{\left(-k \cdot \left(|p_{\text{perceived}}| - d_0\right)\right)}}$$

where:

$$d_0 = \frac{d_{\min} + d_{\max}}{2}$$

$$v_{\text{used}} = (1 - f_{\text{new}}) \cdot \min(v_{\text{ball}}, v_{\text{close}}) + f_{\text{new}} \cdot v_{\text{ball}}$$

The logistic function is well known for its smooth S-shaped curve. By employing it here, we ensure that the transition from the capped velocity (used when the ball is perceived very close) to the actual ball velocity happens gradually and non-linearly. The parameter k controls the steepness of the transition, and d_0 is the midpoint of the interpolation range. This method mitigates abrupt changes in the effective velocity, which can lead to jerky or unpredictable motion. As a result, the robot's prediction of the ball's future position becomes more stable and reliable, especially under varying distance conditions. This refinement is particularly important during fast ball movements, where even minor estimation errors can significantly affect motion planning.

3. Cosine-Based Interpolation of Offsets in calcBasePose()

Our study demonstrates that implementing cosine-based interpolation significantly enhances motion fluidity and reduces abrupt transitions, which leads to improved stability and energy efficiency. These results are especially crucial as we prepare for RoboCup SPL 2025, where high-performance humanoid robotics are at the forefront of innovation. This breakthrough has the potential to influence both academic research and practical implementations in competitive robotics.

Moreover, sharing our results in a peer-reviewed venue will facilitate knowledge exchange and stimulate further research. It provides the robotics community with robust, adaptable strategies that can be applied to various locomotion systems beyond the current B-Human framework. Ultimately, publishing these results will help set new benchmarks in the field and reinforce the importance of advanced control techniques in high-stakes competitions like RoboCup SPL 2025. If the results, go as expected we will have publishable material to contribute.

Previous Model:

In the original implementation, the robot blended two offset values (one for a kick-forward and one for a turn kick) using a simple linear interpolation:

$$\text{offset} = (1 - \lambda) \cdot \text{offset}_{\text{forward}} + \lambda \cdot \text{offset}_{\text{turn}}$$

This linear combination provides a direct weighted average based on the factor λ, but it can introduce abrupt transitions when λ changes quickly. These parameters were originally defined by B-Human. Below we propose a new model.

New Model:

Our new approach employs a cosine-based interpolation to ensure a smoother transition:

$$\lambda_{cos} = \frac{1 - \cos(\pi\lambda)}{2}$$

offset is then computed as:

$$\text{offset} = (1 - \lambda_{cos}) \cdot \text{offset}_{\text{forward}} + \lambda_{cos} \cdot \text{offset}_{\text{turn}}$$

Cosine-based interpolation is widely used in graphics and robotics to generate smooth transitions. The derived factor λ_{cos} modulates the blending of offsets in a non-linear, sinusoidal fashion, which significantly reduces sudden changes or oscillations that might occur with linear interpolation. This method leads to more natural movements during transitions between kick strategies. The robot can adjust its posture more fluidly, resulting in improved walking precision and turn execution. Additionally, by reducing the likelihood of abrupt adjustments, energy consumption is optimized, as fewer corrective movements are needed during dynamic interactions with the ball. This approach is versatile and can be readily adapted to various robot locomotion systems beyond the current B-Human codebase.

4. Dynamic Threshold in calcInterceptionPosition

The decisions that the robots make are based on all the inputs received by their vision system, which is limited by its field of view, and on the calculations that determine whether obstacles, the ball, or other robots are close enough to trigger different game modes. For this reason, we introduce a dynamic threshold involving ball velocity. Our initial results in these areas are very positive and show a substantial improvement. This proposal and its associated measurements constitute a solid technical advancement with a high probability of success in publishing the results in medium- or high-level journals.

Previous Model:

Originally, the decision to adjust the interception position of the ball was based on a fixed angular threshold of 45°:

$$\theta_{\text{threshold}} = 45°$$

This fixed threshold did not account for varying conditions such as ball speed, which could require sensitivity in the adjustment.

New Model:

We introduce a dynamic threshold that decreases as the ball's speed increases:

$$\theta_{\text{threshold}} = 45° - min\left(\frac{v_{\text{ball}}}{v_{\text{max}}}, 1\right) \times 15°$$

The idea behind the dynamic threshold is to adapt the sensitivity of the interception adjustment based on the current speed of the ball. When the ball is moving slowly, a larger threshold (close to 45°) is acceptable because the ball's trajectory is less volatile. However, at higher speeds, even small angular deviations can result in significant positional errors; hence, the threshold is reduced (down to a minimum of 30° when the ball is at or above maximum speed). This adaptive mechanism ensures that the robot is more responsive during high-speed interactions, making timely and precise adjustments to its interception strategy. By dynamically modifying the threshold, the system can better cope with the inherent uncertainties of fast-moving objects, leading to improved overall control and reduced risk of misalignment during critical maneuvers.

5. Improved Pass Evaluation Model

In this area, we introduce a refined model for pass evaluation. We establish a hierarchy to personalize our preferences in certain game situations through a weighted sum. This is crucial for gaining a more comprehensive understanding of the game, as it allows the robots to develop awareness of the moments when it is best to make a pass, considering all relevant factors. In the previous model, the same factors were considered, but they were merely included as a multiplicative factor, where extremely high or low coefficients led to the discrimination of other factors.

This area of development is measured statistically by analyzing the number of failed passes made by the robots within a certain period of time. Additionally, this new model is also reflected in pass quality, as a higher coefficient in pass evaluation increases the likelihood that the robots will be in a favorable position to execute the pass. This improvement has demonstrated a highly positive impact on the robots' game sense. Moreover, our technical and practical evaluations indicate a strong potential for publishing this change in a journal as one of our key advancements in the game.

Previous Model:

Originally, in B-Human code, the pass evaluation rating was computed using a multiplicative approach, where the overall rating was defined as the product of several individual factors:

$$R_{\text{old}} = \text{passTargetFree} \times \text{passLineFree} \times \text{passTargetInField} \times \text{shotLineFree} \times \text{targetInRange}$$

This straightforward method, however, made the overall rating extremely sensitive to any single factor that approached zero. In practice, if one factor scored very low—due, for example, to an opponent's proximity or a pass target being near the field boundary, the entire pass evaluation would drop sharply, regardless of favorable conditions in other factors. Furthermore, the multiplicative model did not allow for fine-tuning the relative importance of each factor, which limited its adaptability to varying match conditions.

New Model:

To address these limitations, we introduce a new model that combines a weighted linear sum of the evaluation factors with a logistic transformation. In this new approach, a weighted sum S is first calculated:

$$S = w_1 \cdot \mathit{passTargetFree} + w_2 \cdot \mathit{passLineFree} + w_3 \cdot \mathit{passTargetInField} + w_4 \cdot \mathit{shotLineFree} + w_5 \cdot \mathit{targetInRange}$$

We set the weights as $w_1 = 0.3, w_2 = 0.2, w_3 = 0.15, w_4 = 0.15, w_5 = 0.2$, allowing us to reflect the relative significance of each component in determining pass success. The weighted sum S is then passed through a logistic function to obtain a smooth, bounded final rating:

$$R_{\text{new}} = \frac{1}{1 + e^{-k\,(S-b)}}$$

Here, k represents the steepness parameter that we defined as 10, controlling how sharply the rating transitions from low to high, and b is the bias set by us as 0.5. This transformation ensures that the pass evaluation rating changes in a gradual and continuous manner, preventing abrupt shifts and reducing the impact of noisy or uncertain inputs, as we could see it.

The rationale behind the new model is to provide a more robust and flexible framework for pass evaluation. By employing a weighted linear combination, the model allows for partial compensation; a low score in one factor does not completely override the influence of other favorable factors. The logistic transformation further smooths the output, making the rating less susceptible to extreme values and ensuring that small variations in the combined score led to proportionate changes in the final evaluation. This enhanced method thus offers improved decision-making capabilities in dynamic playing conditions by more accurately capturing the nuances of pass quality.

4. Unpublished Results

This is our first participation in the competition, so we haven't had much previous experience or have had any results published. The changes made to the code have increased the robots' organization when passing each other the ball or when changing directions, pass evaluations, and the dribbling engine. Even though we have validated each of these five improvements individually, we are working on a statistical model to measure more accurately the results of these improvements working together, we are planning to publish these results in the near future.

5. Impact

In our campus, we foster enthusiasm for robotics competitions while attracting new students by showcasing the exciting and innovative world of robotics. This project aims to inspire students from Tecnológico de Monterrey, as well as from other universities across Mexico and Latin America, to explore robotics and participate in competitions like RoboCup SPL. By nurturing an interest in robotics and programming and promoting interdisciplinary learning, we are laying the groundwork for the creation of specialized academic programs in these fields.

Competitions such as RoboCup SPL offer invaluable hands-on experience in robotics and programming, enabling students to develop practical skills that are highly sought after in today's job market. These experiences complement the theoretical knowledge acquired in the classroom, resulting in well-rounded and competitive professionals.

Moreover, our vision includes the establishment of tailored academic programs that will further enhance students' skills and prepare them for successful careers in robotics and programming. We also view this project as a catalyst for attracting investment and funding from both the private and public sectors. Such support would not only sustain competitions like RoboCup but also boost student engagement in academic and technological arenas, thereby driving innovation across the region.

6. References

1. *Bhuman. (n.d.). GitHub - bhuman/BHumanCodeRelease at coderelease2019. GitHub. https://github.com/bhuman/BHumanCodeRelease/tree/coderelease2019*
2. *Bhuman. (n.d.-b). GitHub - bhuman/BHumanCodeRelease at coderelease2023. GitHub. https://github.com/bhuman/BHumanCodeRelease/tree/coderelease2023*
3. *Bhuman. (n.d.-c). GitHub - bhuman/BHumanCodeRelease at coderelease2024. GitHub. https://github.com/bhuman/BHumanCodeRelease/tree/coderelease2024*
4. *Hongwei, Q. I. A. N., Chenggang, L. I., Zhaocai, D. U., Meng, L. I., Shijie, D. I. N. G., Pengfei, L. I., & Zhipeng, L. I. (2022). Iterative Identification of Robot Dynamic Parameters Based on Logistic Function. Transactions of Nanjing University of Aeronautics & Astronautics, 39(6).*
5. *Yuan, S., Huang, R., Yu, F., & Niu, H. Research on Robot Trajectory Planning Based on Interpolation with Cosine Function Transition in Joint Space.*