Barelang63 Team Description 2026

Hendawan Soebhakti, Senanjung Prayoga,Rifqi Amalya Fatekha,
Anugerah Wibisana, Fadhil Tsaqif Ash Shiddiq, Kevin Indrawinata,
Tegar Adhi Nugraha Christ During, Arya Ferdiansyah Putra,
Prhayogo Eko Sumitro, Ichsan Fajar Yudika, Bambang Nurwahid,
Aldi Faber Sulaeman Pakpahan, Faisal Azmi Putra.

[1] Barelang Robotic Artificial Intelligence Labs (Brail)
[2] Politeknik Negeri Batam Jl. Ahmad Yani, Batam Centre, Batam, 29461 Kepulauan Riau, Indonesia
Barelangkrsbi@gmail.com

Abstract. This paper details the development of the Barelang63 Middle Size League (MSL) robot soccer team for RoboCup 2026. The system integrates mechanical, electrical, and software enhancements to optimize autonomous decision-making in dynamic environments. Key improvements include an omnidirectional vision system using YOLO-based detection and color segmentation, optimized with TensorRT for real-time inference. The use of an Oriented Bounding Box (OBB) model. Hardware reliability is bolstered by a redesigned power distribution system, while robot coordination is managed through a Real-Time Database (RTDB) framework.

Keywords. Barelang63, Robot Soccer, Middle Size League

1. Introduction

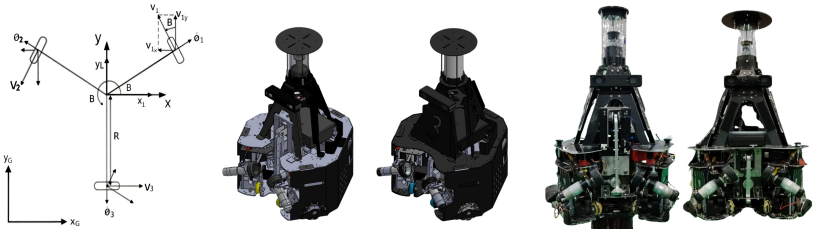
Barelang63 is the RoboCup middle-size league (MSL) football team from the Batam State Polytechnic, Indonesia. The project involved students, staff, and former members of the Barelang63 team for the development of all mechanical components, from hardware to software. The Barelang63 squad was founded in 2016 to carry out developments in the field of robotics, especially in the Middle Size League branch. Barelang63 has participated in several national and international competitions, including the Indonesian robot contest, the RoboCup world championship (5th place in 2022), and the Asia Pacific RoboCup (2nd place in cooperation challenges). This paper describes the robot platform, the technology used in the robot, and the development of the robot from the previous year. Part 2 introduces the Barelang63 soccer robot. Then, section 3 describes the improvement of the robot kick system. Part 4 describes the vision system of the robot platform. Section 5 shows the communication of our robots. Part 6 describes the strategy play of the robots. The last section is the conclusion of this paper.

2. Robot research

2.1 New Striker and Goalkeeper Robot Design

This robot's mechanical system is equipped with three omnidirectional wheels for the striker robot, allowing the robot to move forward, backward, or even diagonally. This robot measures 51.6 cm x 50 cm x 76 cm. The entire body of the robot is made of aluminum and PLA+ filament. The Barelang63 robot uses a Maxon RE 40 Ø40 mm Graphite Brushes 150 Watt motor and a PG45 motor for robot movement, two high-speed DC motors with a diameter of 28 mm for ball gripping, and a solenoid as a ball kicker. The Barelang63 attacking robot uses three omnidirectional wheels with a configuration as shown in Figure 1(a). This configuration allows

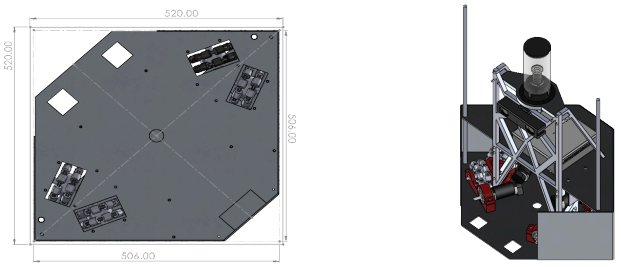
the Barelang63 robot to move in all directions to control the movement system, then kinematic equations are used in local and global coordinates.



A B C

Fig. 1 .(A) Inverse kinematic diagram (B) Barelang 63 Robot Striker Design (C) Application of design to Robots

The goalkeeper robot is designed using four omnidirectional wheels with a configuration as shown in Figure 3, allowing the robot to move left and right quickly. This configuration allows the robot to move in all directions to control the motion system, then the kinematic equations are used in local and global coordinates. We also use four external encoders at the bottom of the robot to determine the position of the robot in Figure 2(a)[1].



A B

Fig. 2.(A) Base design on the robot keeper Barelang 63 (B) Barelang 63 Robot Goalkeeper Design for RoboCup.

2.2 Natural Dribble

Natural dribble is a research project in the program division that began in 2025. We took several references regarding this system and began to find a system that was compatible with our robot, even though there were still failures in performing natural dribbling. For natural dribbling, we took data from the robot's speed as a reference for the speed of the dribble motor with a PID control system, then we took data from the robot's movements to determine whether the robot was moving. After obtaining the data, it then enters the natural dribble logic.

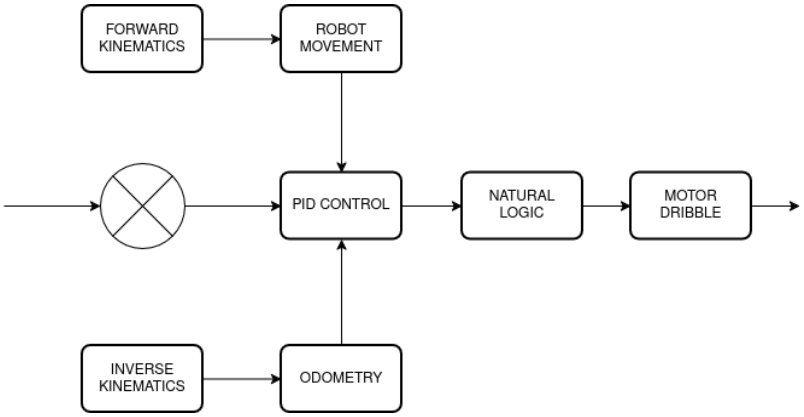
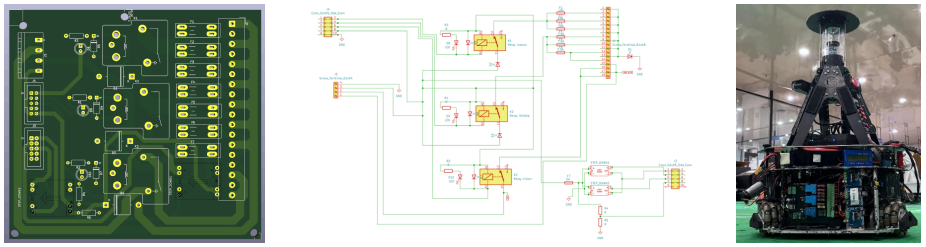


Fig 3. Barelang63 Natural Dribble Flowchart

3. Improved Fusebox System

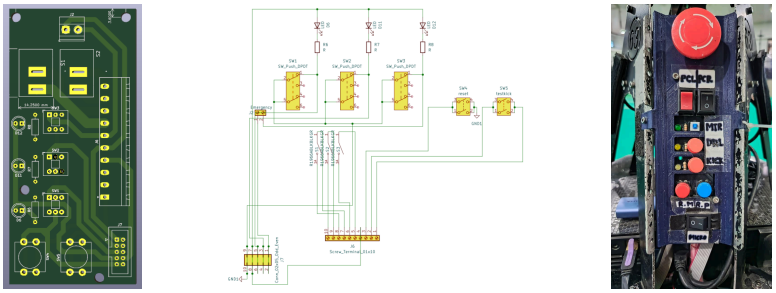
We replaced the old fuse box with a new one. Its functions remain the same, namely controlling the base motor, controlling the dribble, controlling the lifter up and down for the ball kicker, controlling the microcontroller on/off, and controlling the 12v voltage input for the ball kicker system.



A B C

Fig 4.(A) 3D New Fusebox (B) Schematic New Fusebox (C) Application of design to robots

This fuse box is designed to control everything in the robot except for the mini PC, using a relay switch system that is similar to a relay module and has two voltage inputs in the fuse box, 24v and 12v, which are divided differently. With this new fuse box circuit, we have also created a new switch. Within this switch circuit, there is a lower motor switch to indicate the motor relay is active, a dribble motor switch and a lifter up/down switch, a switch to kick the ball, a micro on/off switch, a position reset button and a micro reset button, as well as an emergency stop button to cut off the lower motor voltage. To indicate that voltage is entering the relay, we use LEDs on the switch circuit and the fuse box.



A B C

Fig 5.(A) 3D New Switch (B) Schematic New Switch (C) Application of design to robots

4. Vision

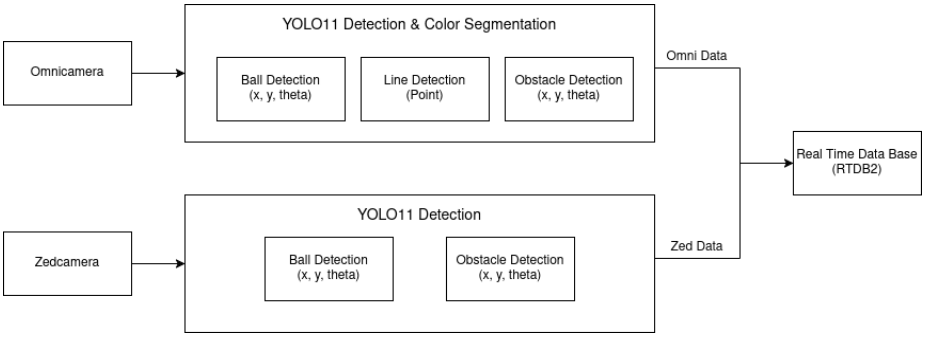


Fig 6. Barelang 63 Vision Flowchart

The 360-degree viewing angle of the Omnicamera enables comprehensive object localization through the YOLO11 Detection & Color Segmentation module, which includes ball detection (x,y,θ), line detection (point), and obstacle detection (x,y,θ). To optimize accuracy, the system implements RGB Image Conversion and a Kalman Filter for more precise state estimation against noise.

The resulting processing output is packaged as Omni Data, which is then transmitted and stored directly into the Real Time Data Base (RTDB2) for real-time robot data synchronization.

4.1 Imu Orientation

In the orientation system here, we are upgrading it using the orientation Imu sensor that is on the Zed stereo camera. The imu sensor on the zed camera has a smalle12r error value compared to the imu GY - 25 sensor. The IMU sensor from the zed stereo camera will certainly be useful to assist movement and reduce errors from the odometry robot.

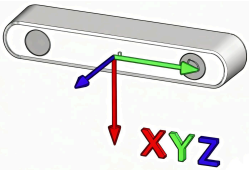


Fig 7. Nine DOF imu zed camera

4.2 OBB Base Model

In our omnidirectional camera system, we use an OBB (Oriented Bounding Box) model for object detection. Unlike a standard bounding box, the OBB can rotate and align with the orientation of the detected object. This allows the bounding box to fit the object more precisely, especially in omnidirectional images where objects can appear at different angles. By using the OBB model, the system can obtain more accurate object position and distance information relative to the robot's center. This improves the precision of localization and helps the robot perform better in navigation and obstacle avoidance.

OBB Based Model;

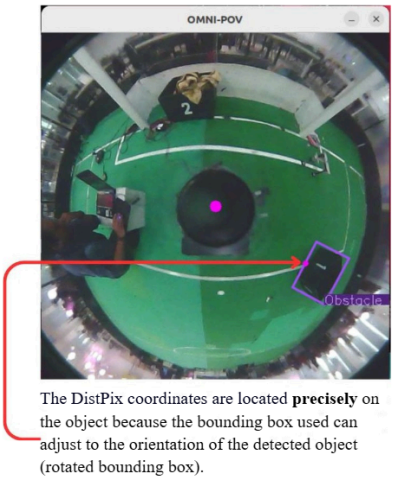


Fig 8. Omni Camera

4.3 TensorRT Optimization for Omnidirectional Vision-Based Object Detection

To achieve efficient real-time inference, our system uses TensorRT as the primary inference engine on the NVIDIA GeForce RTX 2060 GPU. TensorRT optimizes the neural network by converting the trained model into a GPU-specific engine with hardware-aware optimizations such as layer fusion and precision calibration (FP16). This significantly reduces inference latency and improves overall performance compared to our previous implementation using ONNX Runtime.

As shown in Figure X, the **nvidia-smi** monitoring tool confirms that the inference process runs directly on the GPU through the **omni_vision** module, with stable memory usage and active GPU utilization. These results demonstrate that TensorRT enables efficient GPU resource management and ensures reliable real-time object detection, which is essential for robotic perception and navigation.

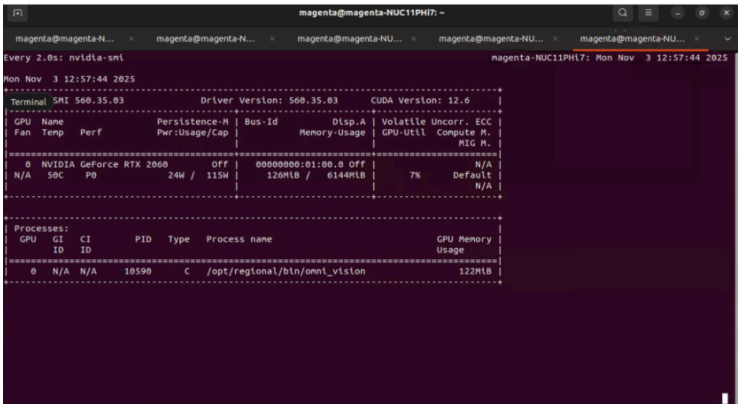


Fig 9. GPU monitoring using the **nvidia-smi** tool

5. Robot Communication

The communication system between robots that we use is RTDB which was developed by the CAMBADA team and the Falcon team. The RTDB communication system functions as middleware, namely for communication between programs in the local agent and communication with other agents[4].

RTDB is easier and more flexible to communicate between robots than using the UDP communication protocol because the data sent is in the form of a serialized structure making it easier to process the data.

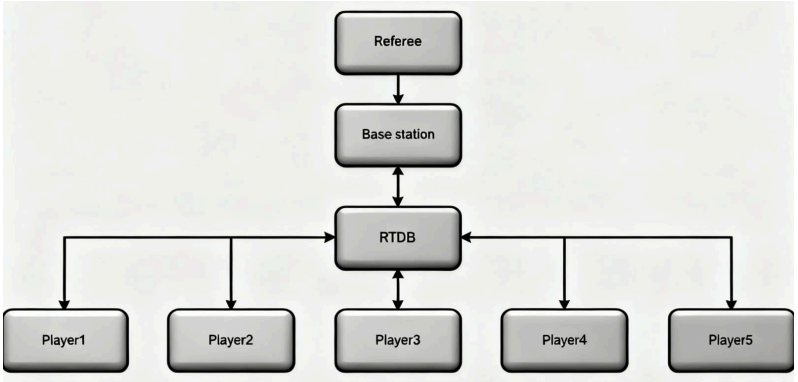


Fig 10. Flowchart Robot communication system using RTDB.

6. Robot Strategy

The Barelang63 robot strategy uses basic strategies such as passing, ball handling, shooting at goal, avoiding obstacles, determining defenders, and determining attackers. Decision - making is done with the behavior tree algorithm, the behavior tree algorithm starts by executing each node sequentially, known as the control flow node, determined by the parent and child relationships. The signal to start execution is given to the root node, with a specified frequency as necessary. After execution, the node returns the running value. If the execution successfully reaches its destination, it returns a success value; otherwise, it fails. The behavior tree uses two control flow nodes, sequence, and fallback, followed by two execution nodes, action, and condition. Sequence nodes are marked with "->" and start by executing child nodes from left to right. Execution continues with a failure or a running value depending on the step taken and succeeds if all child nodes return a success value, similar to the AND logic gate

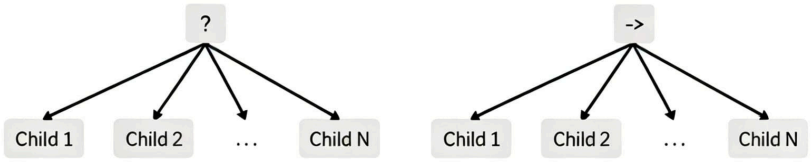


Fig 11. Fallback (left) and sequence (right) node graphical representation.

The figure on the right depicts the sequence node. The action node is on the left, indicated by a "?" mark. Like the sequence node, the action node begins by executing its child nodes from left to right. If a running value is returned, execution continues. If a failure value is returned, the next child node is attempted until one returns a success value, allowing the node to finish. It operates like an OR logic gate.

Fig 12. Action and Condition node graphical representation

The figure depicts the two types of child nodes in use. The action node executes commands and returns success, running, or failure values. The condition node checks conditions and returns either true or false. Both child nodes work together to create a condition check that can be executed in an open or closed loop.

7. Conclusion

This paper describes the latest research on the development of capabilities and systems used in the Barelang63 soccer robot from a mechanical, electrical, and software perspective. Several improvements that have been made and are currently underway include the first part, which is a change in the mechanical design of the goalkeeper robot.

The second change is the input distribution system on a robot drive component using a frog relay, which is useful for safety so that when the button is pressed, the input does not only

pass through the button switch to move the kicking foot to the side. Now, the Barelang63 soccer robot uses an up - down system that uses a longer foot size than before, which is more flexible in selecting kicks.

The third change is an improvement in the robot's vision detection system, and now Barelang63 has replaced the object detection system in the field using the YOLO method, which is highly relevant and powerful, providing fast and stable detection in real-time.

The implementation of YOLO in this system provides a significant improvement in real-time object detection capabilities. Through a training process using a special dataset, the system is able to increase detection speed and ball tracking stability in various lighting and occlusion conditions. The integration of detection results architecture allows the robot to respond to ball movements more quickly and accurately, thereby improving navigation, dribbling, and decision - making performance during matches. The third improvement is in the robot communication system. Previously, the Barelang63 soccer robot used the UDP communication protocol, but now it has switched to the RTDB communication system, where RTDB communication is easier and more flexible for communication between robots compared to using the UDP communication protocol because the data is sent in a serial structure, making it easier to process.

References

1. K. V. Utama, R. A. Fatekha, S. Prayoga, D. S. Pamungkas, and R. P. Hudhajanto: “Positioning and Maneuver of an Omnidirectional Robot Soccer,” in 2018 International Conference on Applied Engineering (ICAE), 2018, pp. 1–5.
2. Bernardo: “CAMBADA 2013 new Platform.”
3. B. P. T. van Goch, M. J. G. van de Molengraft, I. R. J. E. Merry, and R. Verhage: “Optimizing a solenoid for a Robocup kicker,” University of Technology Eindhoven, 2006.
4. L. Almeida, F. Santos, T. Facchinetti, P. Pedreiras, V. Silva, and L. S. Lopes: “Coordinating distributed autonomous agents with a real-time database: The CAMBADA project,” in Computer and Information Sciences-ISCIS 2004: 19th International Symposium, Kemer-Antalya, Turkey, October 27-29, 2004. Proceedings 19, 2004, pp. 876–886.
5. Mohammad Javad Shafiee et al.: “Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection in Video” arXiv, 2017.
6. J. Redmon et al.: “You Only Look Once: Unified, Real-Time Object Detection,” CVPR, 2016.
7. M. Colledanchise and P. Ögren: “Behavior Trees in Robotics and AI,” CRC Press, 2018.
8. R. E. Kalman: “A New Approach to Linear Filtering and Prediction Problems,” Journal of Basic Engineering, 1960.
9. NVIDIA Corporation: “NVIDIA TensorRT Developer Guide,” 2023.
10. S. Thrun, W. Burgard, D. Fox: “Probabilistic Robotics,” MIT Press, 2005.
11. X. Yang et al.: *"Arbitrary-Oriented Object Detection with Circular Smooth Label,"* ECCV 2020.
12. Euston, M., Coote, P., Mahony, R., Kim, J., Hamel, T.: *"A Complementary Filter for Attitude Estimation of a Fixed-Wing UAV,"* IEEE IROS 2008.