Robocup Junior Soccer: Lightweight Division

Our Website

HYPERION

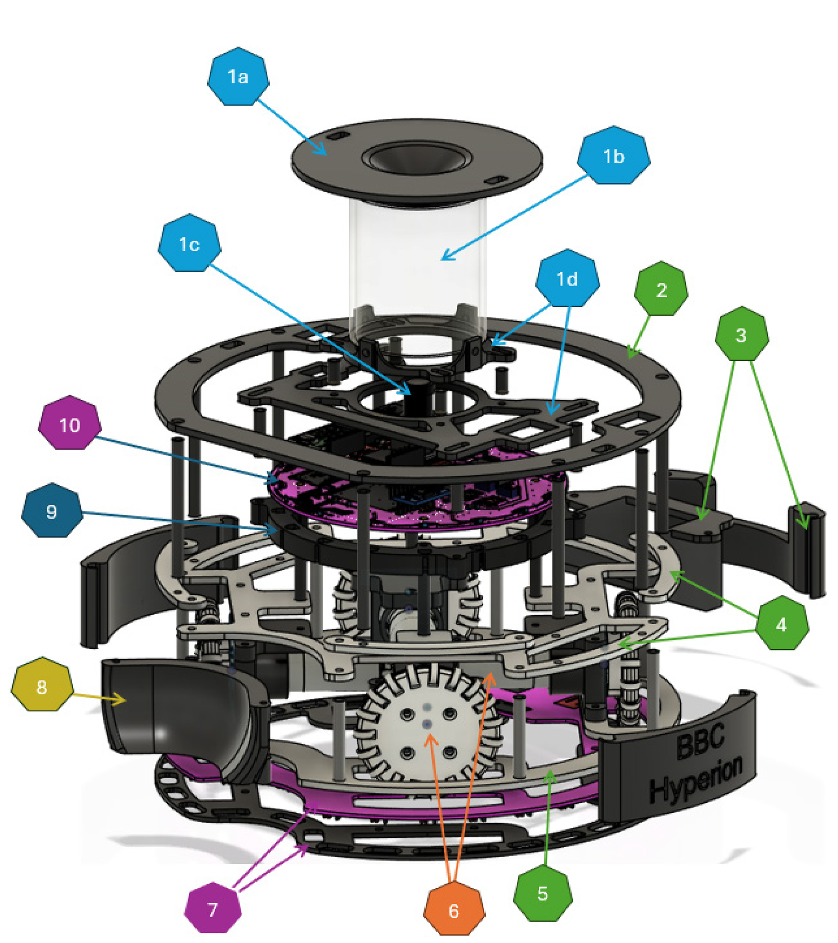
BRISBANE BOYS COLLEGE

Github Link

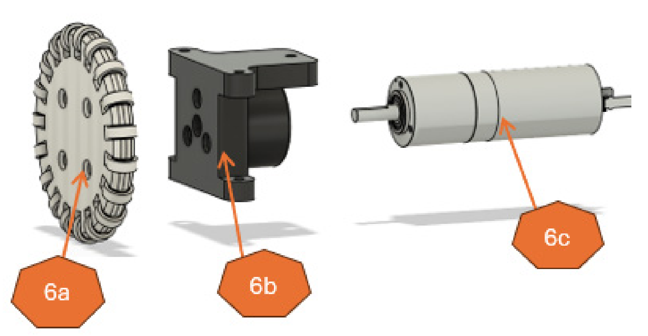
Method / Production / Design

> Robot Hardware Overview [Methods/Production/Design]

Both of our robots are built on a shared design. Each robot features four motors, providing power, speed, and precise control for 360-degree movement. The goal of our robot design was to achieve a low centre of mass to improve acceleration, deceleration, and reduce tipping during turns. The frame was structured to compactly house all components without wasted space. We maintained a simple and focused design, prioritising only the components essential for consistent, reliable performance.



CAD 3D Exploded view of overall Robot Structure In AutoDesk Fusion 360



3D Exploded view of Motor Mount In AutoDesk Fusion 360

Drive system (6: 6a, 6b, 6c):
The drive system consists of custom-made omni-directional wheels (6a), motor brackets (6b) and high-performance motors (6c)

The omni-wheels were designed with silicon-coated steel rollers mounted on ABS 3D-printed hubs for improved traction and durability.

Each wheel is powered by a DCX19 Maxon 9V motor (6c), due to its compact size and high torque output. The motors are fixed in place using custom brackets (6b), for alignment and structural stability.

Materials:
Three materials were used for the majority of the robot's design: CNC-machined aluminium 6061, 3D-printed PLA, and 3D-printed ABS. The core structural plates for the robot were constructed from 3 mm thick CNC-machined aluminium, offering excellent strength, durability, and ductility, ensuring the robot's structure remains intact after many collisions. The majority of other designed components were made using 3D-printed PLA due to ease of printing, decent strength and heat-resistant properties, and accessibility. 3D-printed ABS was used to construct the wheels, as, due to their close proximity and connection to the motors, they require very high heat resistance and greater durability compared to other printed components. Thus, ABS was used—as although it is more difficult to acquire and print, its superior performance in these properties make it favourable for this component.

Vision System (1):
The vision system consists of a cone-shaped reflective mirror (1a), a clear acrylic tube (1b), an upward-facing OpenMV camera module (1c), and adjustable PLA mounting plates (1d).

- Mirror (1a) The mirror reflects a full 360° horizontal field of view down into the camera, allowing the robot to continuously track the goal from any position or orientation without needing multiple cameras.

- Tube (1b) The transparent acrylic tube connects the mirror and camera while allowing unobstructed light to pass through—unlike standoffs, it avoids introducing blind spots.

- Camera (1c) The OpenMV camera is programmed using MicroPython and processes the reflected image to determine the angle to the goal.

- Mount (1d) The adjustable plates allow for calibration of the mirror's position to correct for alignment inconsistencies during manufacturing.

General Frame Design (2, 3, 4, 5):
The robot's frame was designed around two main structural components:

- Top Plate (2) The top plate is 3D-printed in PLA and houses key switches and a handle for ease of access during matches.

- Battery Containment (3) The battery holder and a rear support wall between the mid and base plates, which secures the battery and prevents hardware failures caused by movement or impact.

- Mid & Base Plates (4 & 5) The mid plate and base plate, both made from CNC-machined aluminium to centralise mass low to the ground and improve stability during rapid movement.

Capture Zone (8):
The PLA-printed ball capture zone improves contact area with the ball, enabling more consistent control during movement and turns. Its curved design better matches the ball's shape, reducing sudden contact and improving handling.

PCBs (7 & 10):
The PCB's are designed in AutoDesk Fusion 360 and ordered through JLBPCB for fast and good quality boards. (More information in the Eletrical Overview)

- Light Sensor Board (7) The Light Sensor Board allows the robot to detect the boundary line and the Light sensor Board protector protects the Light Sensor Board from any debries left on the field.

- Main Board (10) The Main Board consists of the power supply, microcontrollers, motor controllers, bluetooth & camera.

> Electrical Overview [Methods/Production/Design]

We use 2 customly designed PCBs which are an Intergreated Main Board and a Light Sensor Board.

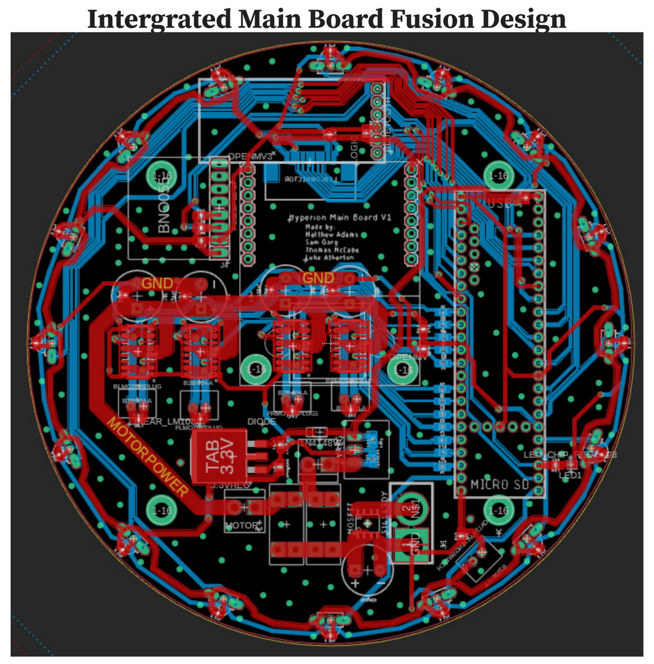
1. Intergreated Main Board
The integrated main board includes a power supply circuit, Teensy 4.1, OpenMV H7 Plus, BNO055 IMU, Bluetooth HC-05 module, and a Light Sensor Board connected via an FFC connector. The power supply circuit is driven by a 3-cell 1300 mAh battery, which powers four VNH7070 motor controllers. It also includes 5V and 3.3V regulators that supply power to the microcontroller and 16 TSSP58038 sensors, respectively. The TSSP58038 sensors are arranged in a circular layout to provide a 360-degree view, allowing accurate detection of the ball's position. The OpenMV H7 Plus was selected for its high detection speed, resolution, and frame rate.

2. Light Sensor Board
The Light Sensor Board is comprised of 32 clusters of components, including phototransistors and LEDs arranged concentrically, and two 4-bit MUX506IPW to reduce Teensy pin usage. The Light Sensor Board allows our robot to distinguish boundary lines of the field (out of bounds, goal box) and other specific edge cases where cross-shaped Light Sensor Boards cannot handle.

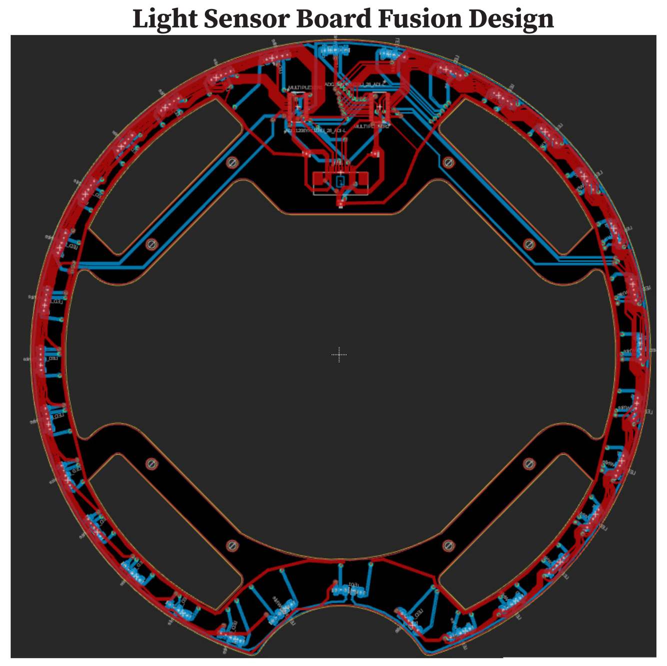
Innovations from 2024:
The main board was changed from multiple different PCBs to a single integrated PCB for easy access to all components and to limit the amount of space needed.

The light sensor board was changed from a small ring with 16 groups of components to a ring surrounding the wheels of the robot with 32 groups of components to adapt to the different field and ruleset from the Australian Open in 2024.

Intergrated Main Board Fusion Design



Light Sensor Board Fusion Design



Meet the Team:

Thomas McCabe	Luke Atherton	Samaksh Garg	Matthew Adams
Software & Documentation	Hardware & Structural	Software & Strategy	Hardware & Electrical

Abstract

Hyperion is an Australian robotics team from Brisbane Boys' College, competing in the 2025 RoboCup Junior Soccer Lightweight League. The team consists of four secondary school students who developed two fully autonomous soccer robots for the international competition. Building on experience from the 2024 season, our goal was to enhance the mechanical reliability, sensor accuracy, and gameplay consistency of our robots.

Each robot features custom-built omni-wheels for precise omni-directional movement, a vision system with a conical mirror and OpenMV camera for 360° goal tracking, and a light sensor array for field boundary detection. The robots are controlled using a Teensy 4.1 microcontrollers programmed in C++ via PlatformIO, with MicroPython used for the OpenMV module. The robots were designed in Autodesk Fusion 360 and fabricated using 3D printing, CNC machining, and powered by custom-soldered PCBs.

To address limitations observed in past competitions, we refined the mirror geometry for improved goal visibility and replaced key structural components with aluminium to lower the centre of mass and increase impact resistance. We approached this season not just to compete, but to grow our technical capabilities and contribute to the wider RoboCup community. By documenting our designs and solutions for future teams to build upon, we aim to support RoboCup's mission of fostering AI and robotics research.

Communication Tools

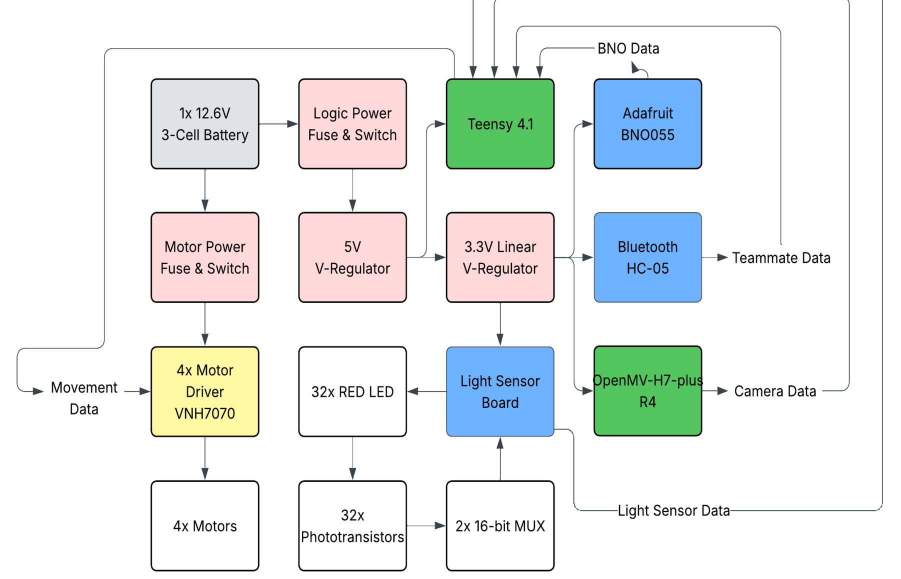
Discord

Microsoft Teams

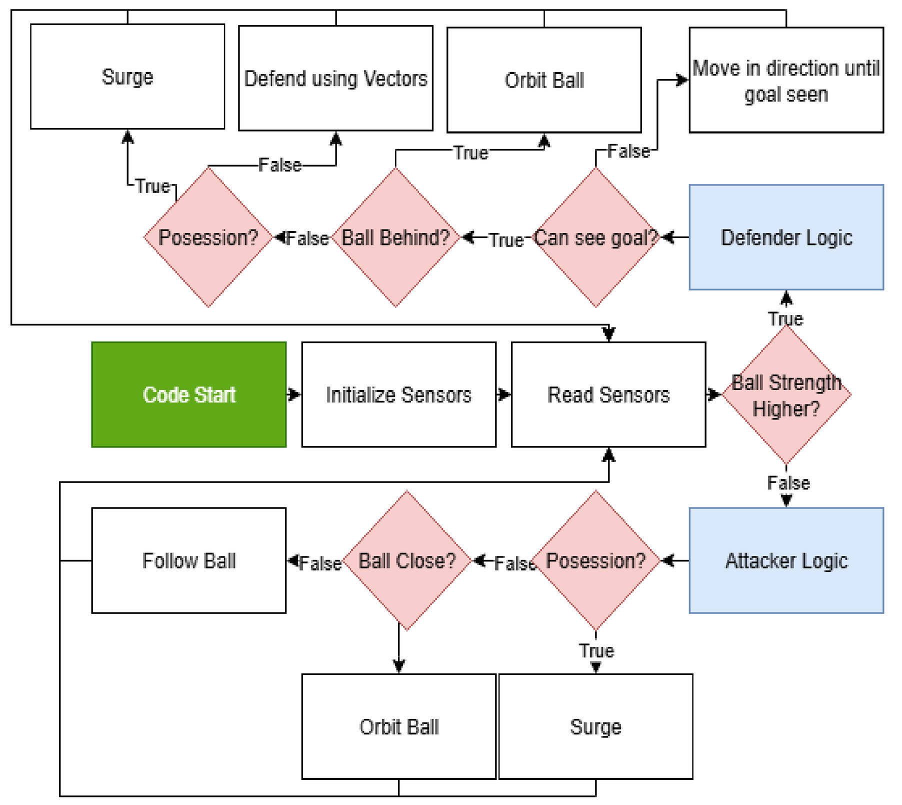
Team Communication and Collaboration, for scheduling sessions, progress updates, asking questions, and sharing files

Flowcharts and Tools for Hardware and Software

Electrical Logic Flowchart



Software Logic Flowchart



Hardware Tools

Autodesk Fusion 360 — 3D CAD modelling and PCB circuit design

Makera — CNC machine provider

Bambu Lab — Printing 3D models for testing & refining

Software Tools

Visual Studio Code — Coding software

PlatformIO — Compiler for microcontroller

OpenMV — Vision programming interface

Github — Version control & collaboration

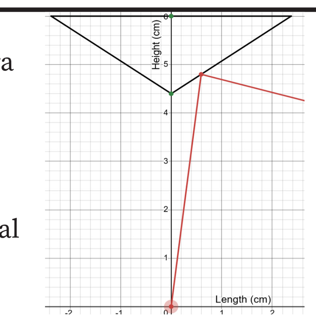
Data / Results / Discussion

Hardware

Mirror Selection and Simulation
To maximise horizontal field of view and ensure uninterrupted vision of the goal, we investigated the optimal mirror shape and dimensions for our omnidirectional vision system.

Mirror Shape Investigation
- Cone-shaped mirror selected over 2D shapes (e.g. rectangles)
  → Reflects light from all horizontal angles into the camera
  → Enables full 360° vision around the robot
- Simplifies vision processing compared to using multiple cameras

Mirror Dimension Simulation
Fixed parameters: Base diameter (23.75 mm), vertical distance from camera to mirror base
Variable: Mirror height — determines reflection angle and field of view
Goal: Find optimal mirror height for full field coverage
Method: Simulated ray reflections (mirror radius 2.375 cm) in Desmos
→ Adjusted mirror angle so reflected ray enters camera just below horizontal
→ Extended ray to floor (14 cm below camera) to find max range
Result: 16 mm mirror height → ~31 m viewing range (well beyond field, accounts for tolerances)



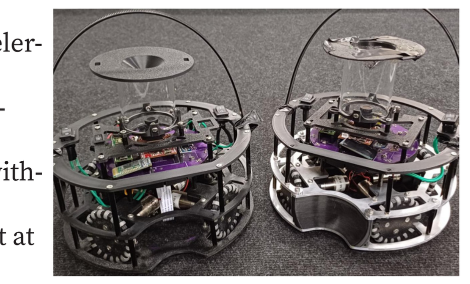
Mirror Dimension Simulation

Aluminium Plate Replacement
We replaced PLA with aluminium 6061 for the mid and base plates to improve structural strength and gameplay durability. This change was prompted by repeated failures during high-speed matches, where abrupt impacts with the goal and other robots caused the PLA plates to crack or break, particularly at stress-concentrated regions.

PLA (previous)	Aluminium 6061 (current)
- Lightweight and easy to 3D print - Low yield strength (~60 MPa) and brittle - Fractures under stress, unsuitable for high-impact applications	- High mechanical strength (~270 MPa) and ductility - Plates deform slightly under stress rather than cracking - More reliable in high-impact applications

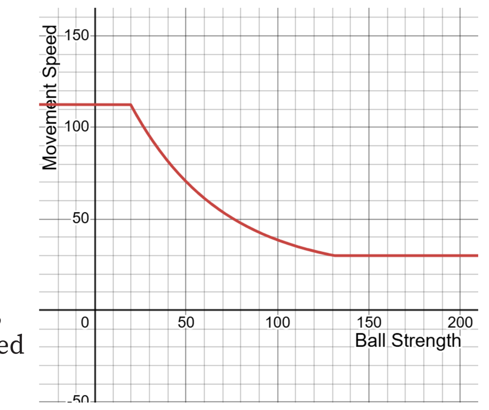
Mass Increase and Centre of Mass Impact
- Aluminium plates added 101 g → Still under weight limit
- Extra mass lowered centre of mass
  → Improved stability during turns and acceleration
- Traction is the limiting factor for acceleration, not motor torque
  → Increased normal force enhanced grip without reducing speed
- Top plate remains PLA to minimise weight at higher elevations

Robots with PLA vs Aluminium plates

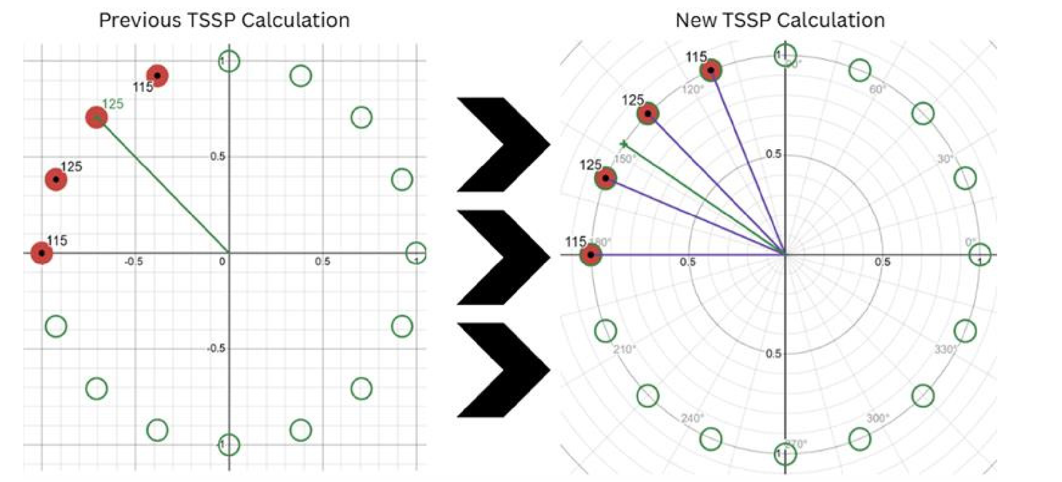


Software

An exponential function maps ball strength (distance to the ball) to the robot's speed, enabling rapid acceleration when far and smooth deceleration when close. This improves control and reduces overshooting, unlike the slower-decelerating linear model previously tested. When the ball is directly ahead, the robot accelerates to top speed to maximise scoring.

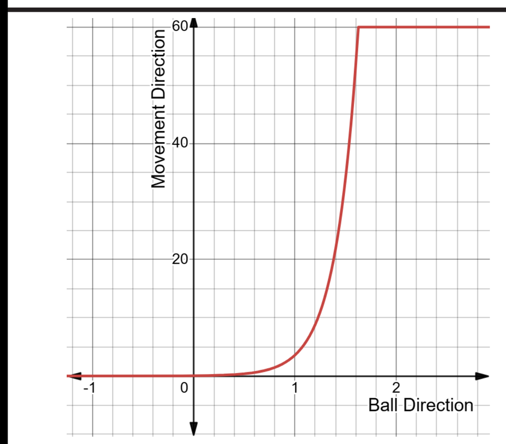


Ball Strength vs Movement Speed Graph



Previous Ball Direction Calculations vs Newly Innovated Ball Direction Calculations

The TSSP System library was custom-built by our team to meet specific design needs. It measures ball direction and strength by sampling each sensor 255 times, returning a value between 0 (no detection) and 255 (constant detection). A consistent 255 typically indicates a faulty sensor. If a single sensor fails, its value is replaced with the average of its two neighbouring sensors. However, software cannot correct clusters of failed sensors, which require hardware repair. Direction is now calculated using the top four TSSP readings, each weighted by its X and Y coordinates on the unit circle, with the angle derived using the arctangent function. Strength is determined through a weighted average: the strongest reading is multiplied by 3, the second by 2, and the third and fourth by 1, then divided by 7. This method improves accuracy and reliability in ball detection, with the example shown producing a strength of approximately 122.14 out of 255.

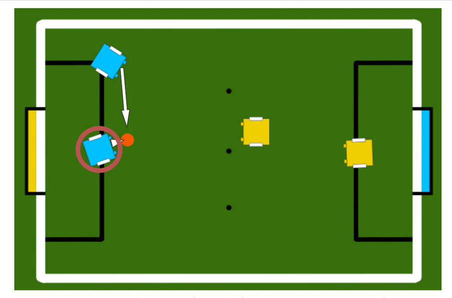


To refine its movement path, the robot adjusts its trajectory using an exponential function based on ball direction. This creates a smooth, circular orbit around the ball, enabling fluid and strategic positioning for attack. Unlike using cases for where the ball is, and moving accordingly, this exponential function provides a more accurate and smooth orbit.

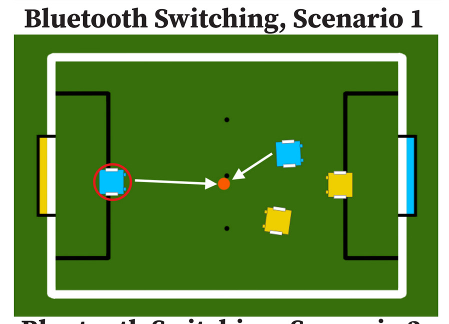
Ball Direction vs Movement Direction Graph

> Software Strategy [Methods/Production/Design]

We utilise an OpenMV H7 Plus module, configured to a QQVGA resolution (160x120 pixels), which is then cropped to a 120x120 pixel square for processing. To identify both goals on the field, the robot employs the OpenMV camera to perform colour-based vision processing. Using Python, it runs the find_blobs function from the OpenMV image library, which detects coloured blobs that match a specified LAB threshold. The center coordinates of each detected blob are extracted and transmitted to the Teensy microcontroller via UART using the writechar function from the pyb UART library.



Bluetooth Switching, Scenario 1

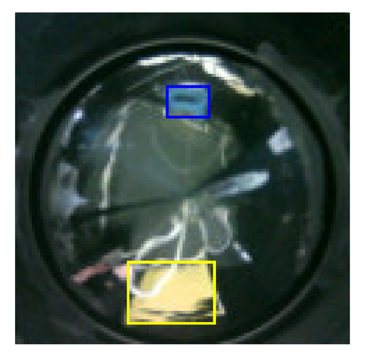


Bluetooth Switching, Scenario 2

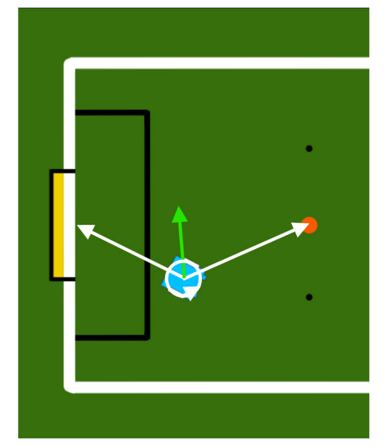
The first scenario prioritises ball strength, with the robot closer to the ball—indicated by higher strength—taking on the attacking role to maintain possession.

If the ball is behind the current attacker and closer to the defender, the robots switch roles minimising backtracking as the defender becomes the attacker and the original attacker takes a defensive role. This ensures the robot with the strongest visibility and least movement takes over the attack.

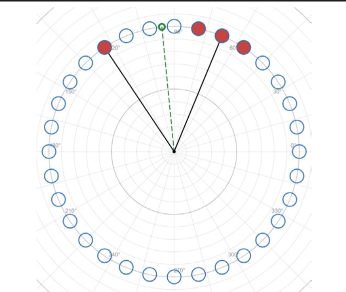
Low/High Speed Moving Areas
The attacker constantly advances toward the goal, guided by input from the onboard camera. It integrates this visual data into PID controllers, allowing it to quickly realign itself if displaced or rotated. Using the camera's X-coordinate data, the robot estimates its lateral position on the field. When positioned closer to the sides—where the goal appears narrower—it reduces speed to enhance scoring accuracy. Conversely, it increases speed when directly in front of the goal, where the scoring angle is optimal.



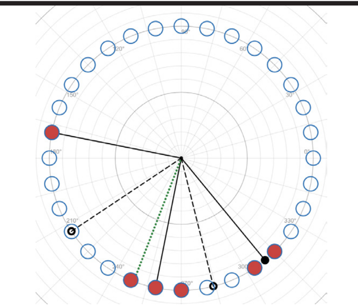
Camera Vision Example



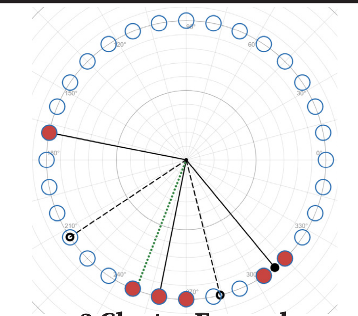
Defender Vectors Example



1 Cluster Example



2 Cluster Example



3 Cluster Example

When the robot detects a single cluster, it assumes the line is centered within that cluster and directly uses the cluster's central angle as the estimated line direction. In the case where two distinct clusters are identified, the robot computes the circular average of their respective angles. This average is interpreted as the line direction, representing the midpoint between the two detected segments of the line. When three clusters are detected, the robot applies a more refined estimation method. It identifies the two outermost clusters based on angular separation and considers the remaining one as the central or inner cluster. First, it computes the average angle between each outer cluster and the inner cluster individually. These two intermediate values represent directional influences from each side of the line. Finally, it takes the average of those two values to derive a balanced and smoothed line direction, accounting for all three detected segments in a hierarchical manner.

Our defender uses three PID controllers to move effectively. The first is a rotation PID, which keeps the robot facing away from our goal at all times. This orientation ensures it is always prepared to intercept the ball as it approaches. The second is a vertical PID, which manages the robot's position relative to the goal. When the robot is far from the goal, the control vector points at 180 degrees and increases in strength with distance. Conversely, when the robot is close to the goal, the vector strength decreases accordingly. The third PID responds to the ball's position on the field. If the ball is to the right of the robot, it moves faster as the angle difference increases, and slower when the angle difference is smaller—though still quickly enough to position itself between the ball and the goal. When the ball is not present, the robot recentres itself in front of the goal using the compass, compensating for any offset caused by the rotation PID. If the ball moves behind the defender, the robot temporarily switches to attacker orbit behavior until the ball is once again in front.