Abstract

Alphatron is an Australian robotics team from Brisbane Boys' College competing in the 2024 International Robocup Junior Open Soccer competition. Our team, consisting of four secondary students, made two robots to compete. The robots have been under development for over 500 hours among each of our members, considering the physical and electrical design and manufacturing of each robot as well as the software that enables a fully autonomous soccer robot.

The robots were built utilising industry standard technologies – 3D printing, CNC routing – as well as through the incorporation of printed circuit boards (PCBs) to increase efficiency and reliability. Elements of considerable importance included a light sensor board, to allow the robots to stay within the field; a camera, to allow for precise tracking of each goal; bluetooth modules, to optimise gameplay performance; a kicker and more. Additionally, a combination of C++ (compiled through Platformio of VSCode) for the Teensy 4.1 and microPython for the OpenMV Camera H7 PLUS (developed via OpenMVIDE) was used for software development. This poster outlines both our production process as well as our findings on how to best build a soccer robot in efforts to share our gained knowledge with other teams around the world. Ultimately, we aspired to progress RoboCup's goal of making robots that have the capabilities to beat the best soccer players in the world.

ALPHATRON

Brisbane Boys' College, Ausrtralia

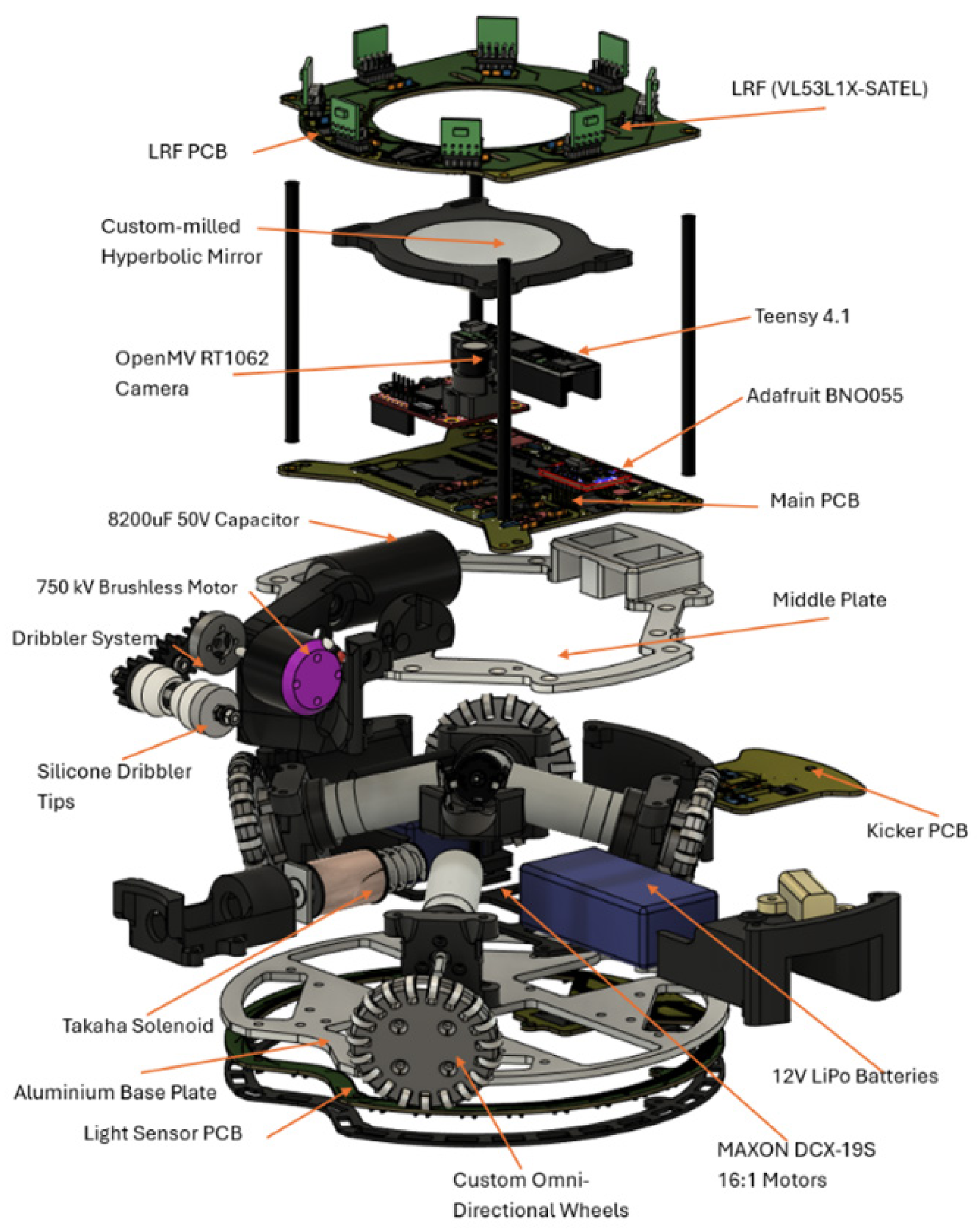
Our Website

GitHub Link

Team members (Left to Right):
Junpeng Huang (Software/Structural), William Robertson (Software/Mechanical), Maxwell Cassimatis (Electrical/Documentation)

Method/Production/Design

Hardware



Hardware Utilities

Software Utilities

Layout & Design Goals

The general layout, including major components, can be see in the exploded representation (left). Our main design goals were:
- Low centre of gravity (previous robots had tendency to flip)
- Earliest detection of line for effective out avoidance
- A balance between range and accuracy for vision/localisation
- Achieve high speeds & rotations with advanced ball control
- Reliable and robust hardware that required minimal repairs

Light Sensor Board Placement

Our current robot iteration featured the LSB surrounding the wheels, enabling the robot to detect boundary lines earlier and avoid getting stuck on field ramps (iterations on the right).

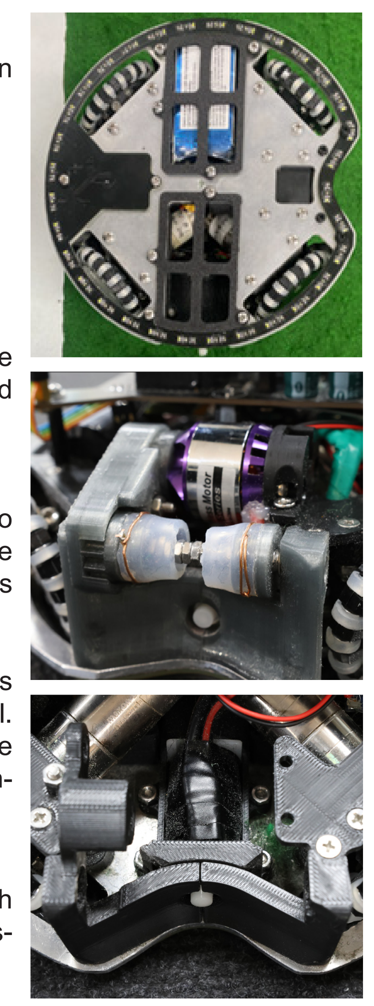
Kicker/Dribbler

The kicker operated through the discharge of a 50V capacitor into a 220-turn hand-wound solenoid using 0.63mm wire. Through the influx of current into the solenoid, a temporary magnetic field is induced, pushing out a metal pin – kicking the ball up to 3.3m.

The dribbler operated through the spinning of a 750kV brushless motor transferred to two conical dribbler tips to handle the ball. This is achieved using three gears in a 1:1:1 ratio, sustaining the motor's maximum of 9000RPM. Silicone tips were used to enhance grip - optimal for high movement speeds and rotations.

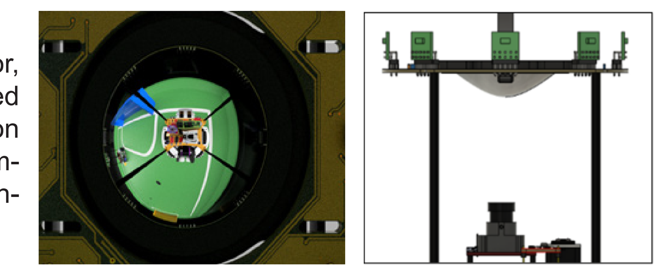
Materials

Our robot extensively used PLA due to its reasonable strength and availability. CF Nylon and ABS were used in the dribbler assembly and omniwheels, respectively.



Mirror Design

Our robots utilise a hyperbolic CNC mirror, with the mathematical function formulated through Desmos and CAD tested in Fusion 360 render. These mirrors' possess high image quality near the robot as well as substantial image range that covers the full field.



Main Board

The main board takes in all sensor readings and acts as a hub to connect the Teensy 4.1, OpenMV, BNO055 IMU, Bluetooth HC-05 modules and LSB (via a FFC). The OpenMV RT1062 model was used due to its superior blob-detection speed, resolution and frame rate. It also contains the power supply circuitry, two 3S 1.3Ah batteries to accomodate for high power draw, and connections to the Turnigy 30A ESC for dribbler motor control. It also included 4 VNH7070 drivers, and pads for the 4 conductive brass standoffs that carried power & signals to/from the LRF array.

Light Sensor Board (LSB)

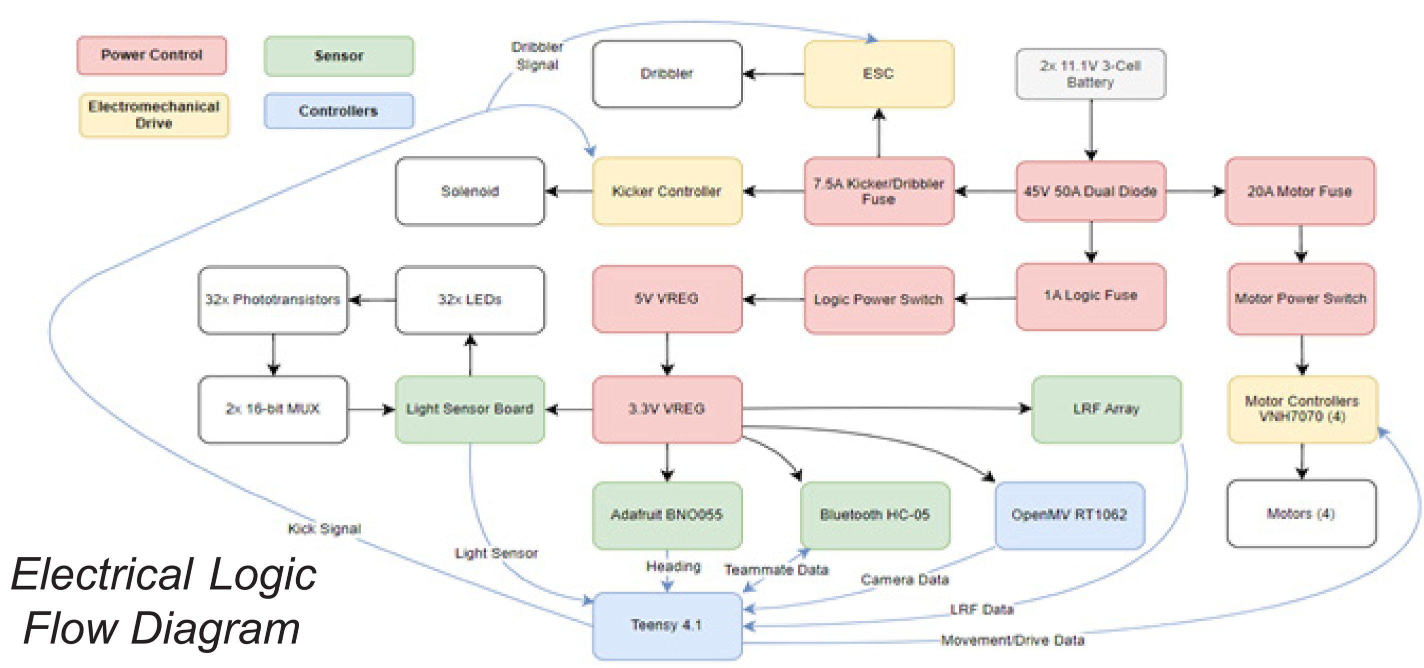
Our LSBs are comprised of 32 clusters of components, including phototransistors and LEDs arranged concentrically, and two 4-bit MUX506IPW to reduce Teensy pin usage. The LSB allows our robots to distinguish boundary lines of the field (out of bounds, goal box) and specific edge cases where cross-shaped LSBs cannot handle.

Kicker Board

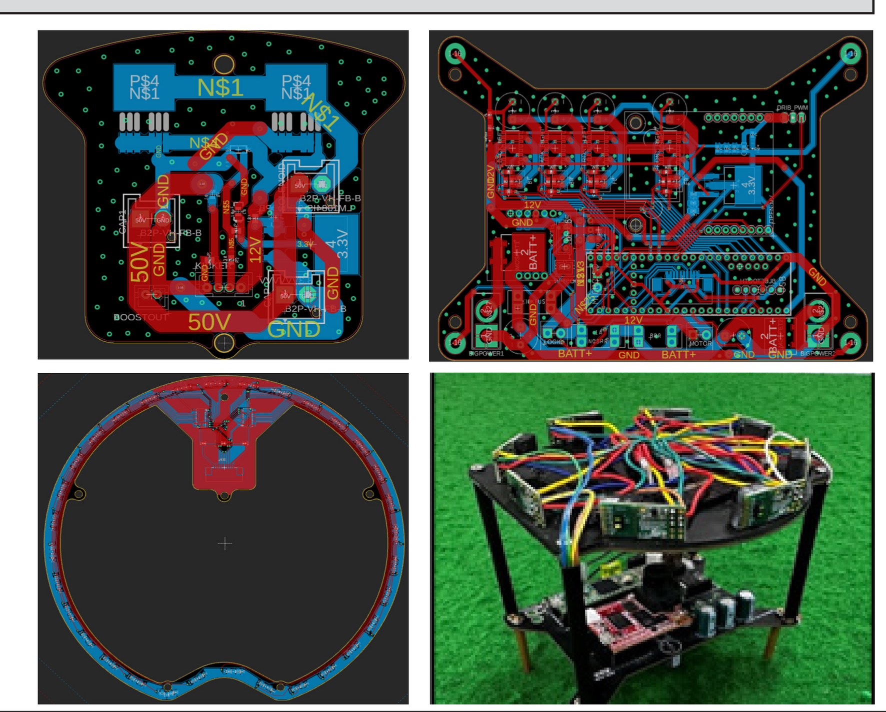
The electrical framework of the Kicker PCB included a voltage regulator, two 80V 270A MOSFETs, a 50Ω power resistor (HSA2533RJ), and an externally connected 8200μF 50V capacitor.

LRF Board

Our team designed an LRF system using 10 VL53L1X Laser Range Finders (LRFs), managed by a Teensy 4.0, which processes raw data into localisation info. To communicate LRF data to the main, we used brass standoffs as bus bars as conductors for serial communication and power.



Electrical Logic Flow Diagram



Data/Results/Discussion

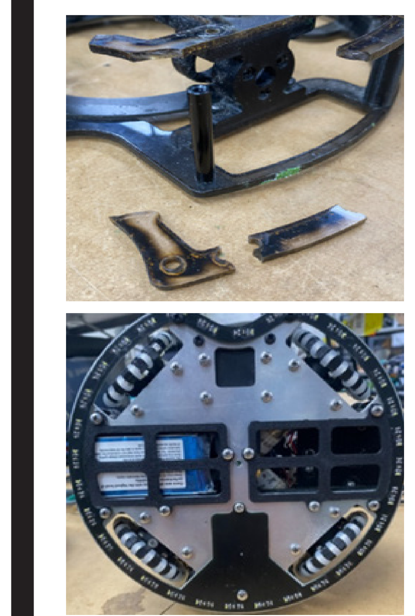


Plate Material Discussion

V1	V2	V3	V4
1.2 mm Aluminium	**3 mm Aluminium**	**3 mm Polycarbonate**	**3 mm Aluminium**
- Too thin (failed to protect other parts) - Prone to bending/ deformation	- Strong and rigid (no deforming) - Too heavy for lightweight soccer	- Lowered weight - Snapped and cracked under stress	- Strong and rigid (no deforming/snapping) - Lower centre of gravity - Suited Open's weight

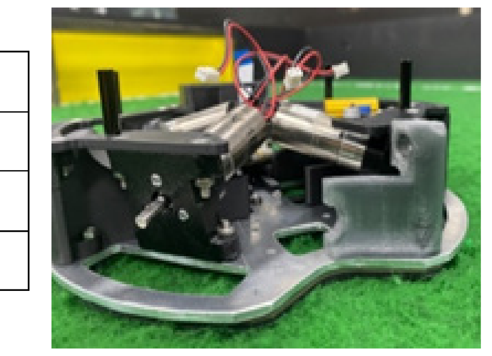
As of present, in the Open division, we returned to using 3mm aluminum plates, balancing rigidity and weight effectively. Combined with an evaluation from a theoretical perspective using Young's modulus (GPa) and yield strength (MPa), which the 3mm alumnium plates had the best combination of, we also methodologicallly tested each material candidate through the following procedure:

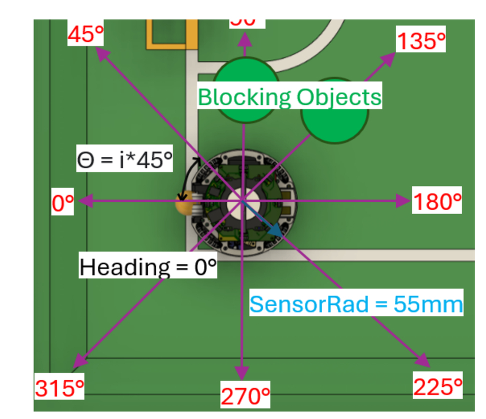
1. Manufactured the exact same design of plates in each material and placed on a testbed robot.
2. Modelled components requiring protection from plates (e.g. LSB, battery guards) with 3D prints.
3. Ran the robot at speeds of 0.5m/s, 1m/s, 1.5m/s and 2m/s into standard field walls.
4. Repeated step 3 by rotating the robot 90 degrees until all relevant angles were tested.
5. Recording no. of repeats of step 3 & 4 until modelled components started deforming/shattering.

Drive System Discussion

We chose the Maxon DCX-19S motors for our robot, leveraging both their reliability, performance and our experience with them in previous lightweight robots. Join-Max motors were dismissed due to high current draw and issues with stalling/stability, while Maxon EC flat motors, despite good performance, were excluded as it would push up the centre of gravity significantly. Maxon DCX-19S motors were too long to mount horizontally due to the LSB innovation and kicker position, we resolved this by tilting them 15 degrees, which did not impact motor speed on the field (measured via tachometer) nor the wheel grip.

Motor Name (9V)	No-load RPM	No-load Current (mA)	Stall Torque (Nm)	Dimensions (mm)
Join-Max	1200	600	1.77	34x32.5x68
Maxon DCX-19S	1075	136	1.56	15Øx68.1
Maxon EC flat	1465	163	0.693	63.2Øx16.5





Localisation Techniques

Camera localisation alone was inaccurate (10cm error), so we combined 10 LRF (VL53L1X) sensors to determine the robot's exact X and Y positions on the field. An uncertainty value measures the reading's reliability continuously, used to find a "confidence" value of localisation. Each sensor measures its distance from the wall. To convert these distances into X and Y coordinates, the sensor direction was adjusted using the angle between each LRF relative to the robot's heading. Using trigonometric calculations, the sensor distance and direction determine the robot's absolute X and Y positions.

Kicker Range & Voltage Test

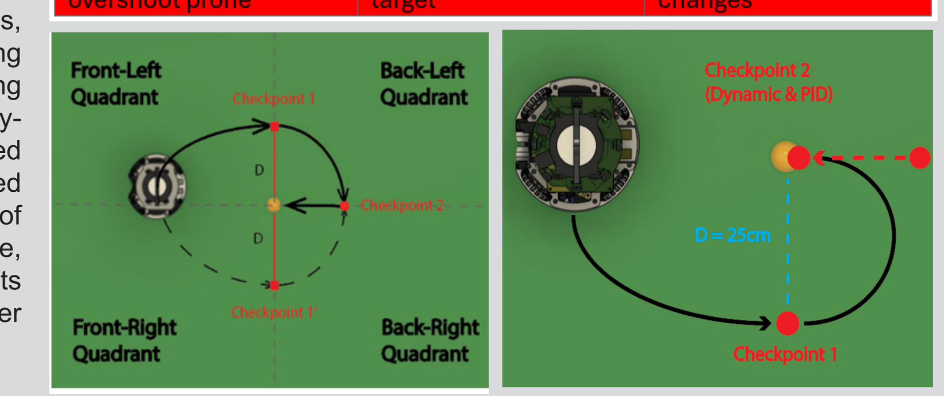
To test the kicker performance, the kick distance was compared to the boost voltage. This was done by adjusting the potentiometer to various voltages and conducting 5 trials for each voltage. The averaged results were: 12V (no boost) - 0.7m, 24V - 1.8m, 40V - 3.3m.

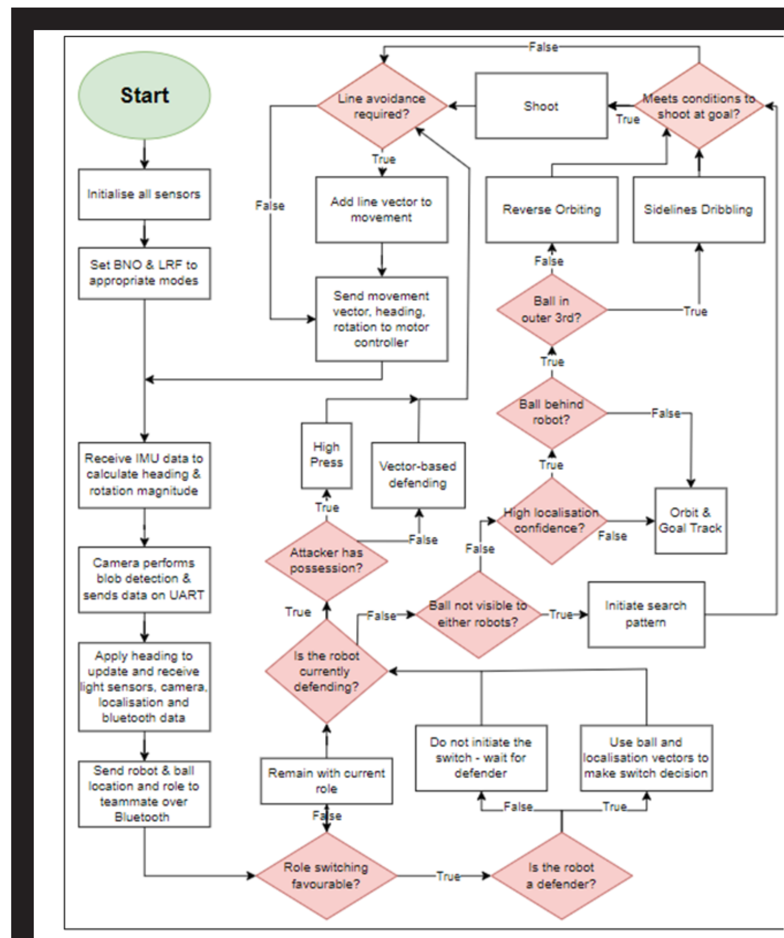
Ball Control Investigation

At the 2023 national competition, our robot used double exponential functions to determine the orbit angle but we found that ball capturing was inconsistent at best (229/387 attempts). Now with linearised ball distance and localisation, three methods were considered to improve ball control (see table on the left).

Initially, we implemented a quadrant-case-based vector orbit (right, first image) but had issues with ragged movement and constant overshooting. To improve this, we added a variable offset in the back quadrants, making the checkpoint 2 vector dynamic. This adjustment, along with an operation-speed-dependent PID and a proximity-based speed function, smoothed the orbit and reduced overshooting. The new method effectively compensated for the vector method's limitations using the strengths of PID (right, second image). It proved much more reliable, achieving over 270 successful captures out of 300 attempts and enhancing efficiency, especially with the dribbler requiring accurate orbiting to make effective captures.

Vector	PID	Exponential
- Intuitive: corresponds to real-life units - Easily control tightness & efficiency	- Easily adjusted to fix overshoot - P & D can auto-scale to robot speed – less tuning	- Nature easily varied through parameters - Effects visualised in Desmos
- Case-based system: not smooth orbit - Linear system: overshoot prone	- Requires initial tuning & field surface change - Requires variable target	- Difficult to tune - Needs tuning if operating speed changes





Software

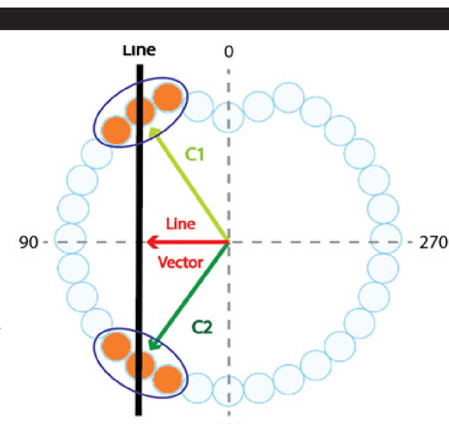
Program Structure

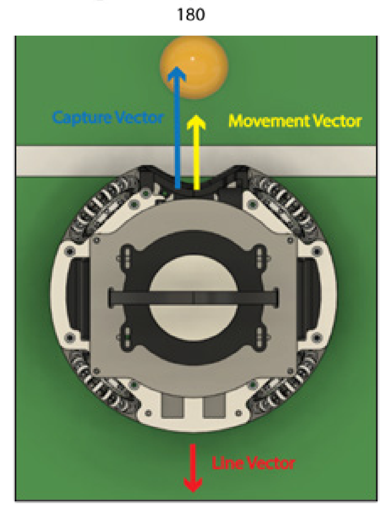
A visual representation of the code's structure can be seen to the left in a logic flow-diagram.

Line Avoidance

While line avoidance was a weak point in Bordeaux, our cluster identification and line angle calculations were reliable. More information on how far the robot was over the line was needed along with more specific geometry of the line to account for the curved goal boxes and T-intersections. Therefore, by treating our LSB as a modified unit circle, we converted the positions of each light sensors to vector form, where each phototransistor reading contributed equally to effectively present the magnitude, angle and shape properties of each line, instead of averaging vectors to each cluster. We also scaled the magnitude of each sensor vector to how much readings were above the line colour threshold.

Using a PID controller for the line vector magnitude, we also combined the attack vector with the line avoidance vector, allowing the robot to chase the ball while staying within the penalty area. This also allowed the robot to surge partially over the line the capture the ball, speeding up the efficiency of sidelines dribbling should it be knocked out of the dribbler.

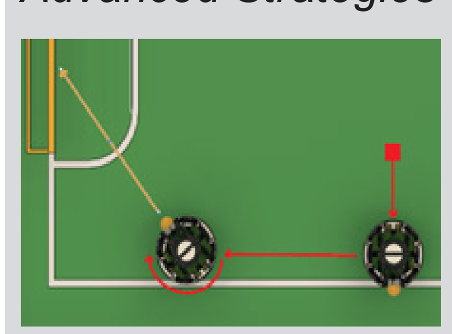




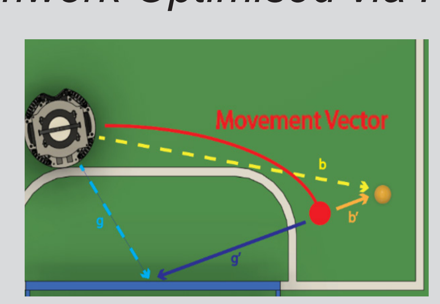
An Absolute Vector-Based System

To achieve a more intuitive absolute reference frame, we revamped all libraries (image processing, movement system, line avoidance, etc.) to account for the robot's heading. Each motor pair was defined by its individual unit vectors, allowing most movements to be converted to vectors (with magnitude in real-life centimetres and arguments as real-life angles) and making movement directions absolute to the field. This significantly reduced rotation-induced errors, improving consistency in all aspects and enhancing our robot's performance in advanced strategies and localisation systems.

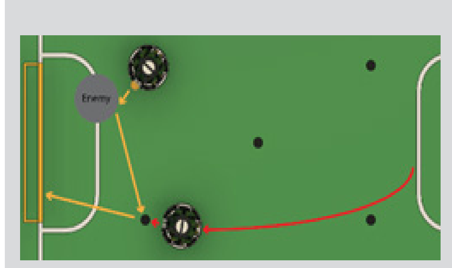
Advanced Strategies - Dynamic Teamwork Optimised via FSM



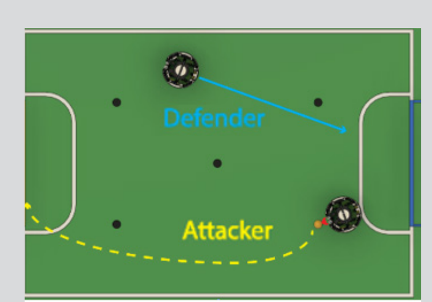
Sidelines Dribbling
Hides ball from opponents, increasing difficulty for opposition defenders to track the ball. Spins & shoots quickly when near the goal to lower chances of interception.



Goal-Keeping
Using vector-based defending and pre-emptive movement through the projection of ball velocity- the most effective method of preventing goalscoring.



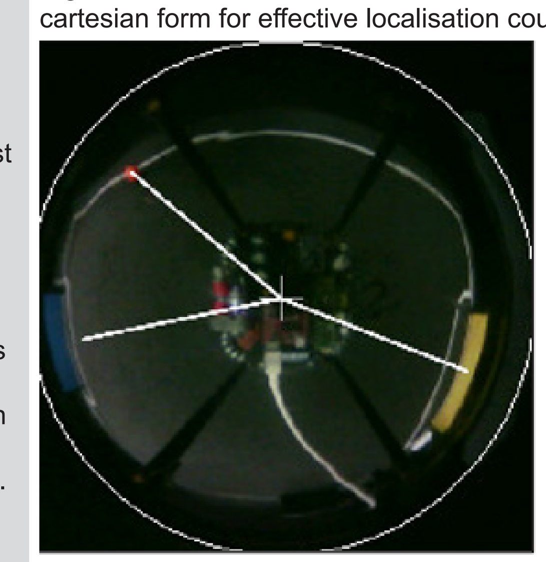
High Press
Defender doubles up the attack while in possession by camping neutral points to initiate another attack immediately if the initial one misses.

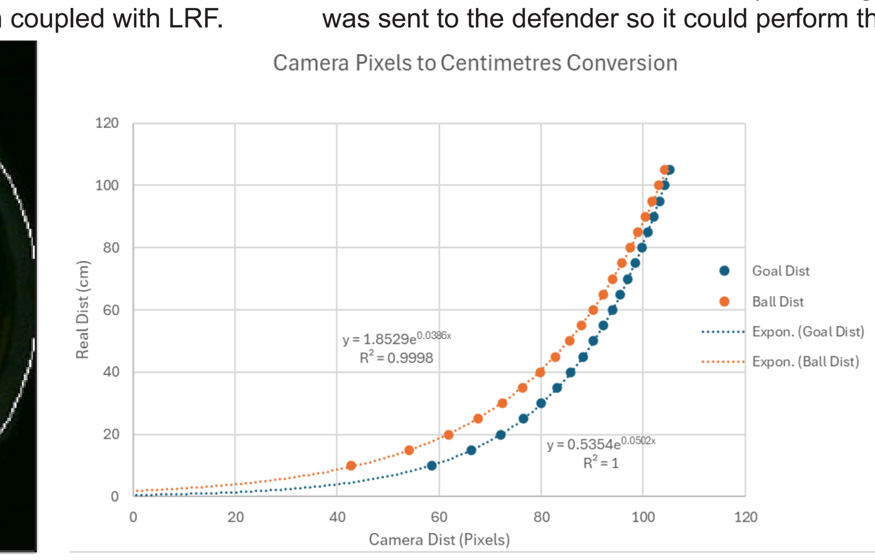


Role Switching
Increases effectiveness of attacking/defending, "clears" the ball while in staying in possession for fast counter-attacks.

Camera Vision

Using the OpenMV RT1062 and OpenCV, the blob detection function returned x & y coordinates of the largest blobs of 3 different RGB thresholds, which were defined for the ball and the yellow and blue goals (example seen below). Communicating data via UART, the Teensy 4.1 performed trignometric calculations to isolate the magnitude and angle of each blob, with the magnitude (originally in camera pixels) linearised to centimetres by developing conversion functions through regression in Excel. These vectors were converted back to cartesian form for effective localisation coupled with LRF.





Inter-Robot Communication

Using the Bluetooth HC-05 modules, the roles (attacker/defender), strategies (see left), and positions of each robot (absolute) and the ball (relative to each robot) (x/y coords, cms) were communicated between robots, where the absolute ball location can be found by adding the two coordinates. With the situation of each robot fully transparent, the robot can then evaluate whether the current strategy its employing is suitable, which provides crucial data for the FSM. This system was also essential due to the golf ball's size and passive nature, and the possibility of opponents hiding the ball. A search pattern would be initiated by the attacker if neither robots has vision (below, right), and the absolute ball location was sent to the defender so it could perform the goal-keeping strategy.