Transcendence 2021
Team Description Paper

Liu Jie Xu, Neo Hao Jun, and Tan Kai Cong

Hwa Chong Institution, Bukit Timah 269734 Singapore, Singapore
Website:https://bozo.infocommsociety.com
Contact Email: bozo@infocommsociety.com

Abstract. This paper documents the recent advancements made by Team Transcendence in preparation for RoboCupJunior 2021. Small hardware changes were made like improving the mirror quality and having a more reliable IMU. Majority of work done has been focused on software, specifically the robots' strategy in goal scoring, localisation and inter-robot communication. Novel, advanced strategies from the simulation are also showcased, as well as details of our future plans which could help turn such strategies into a reality.

Keywords: RoboCupJunior 2021, autonomous robot, vision system.

1 Introduction

1.1 Team Background

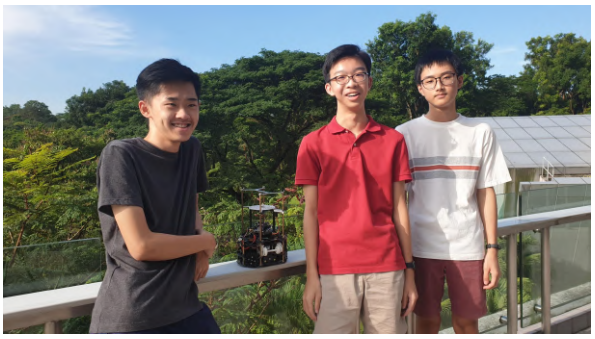


Fig 1. Team Transcendence members

Team Transcendence consists of three students from Hwa Chong Institution, Singapore. The team was first established in 2018, with members who had prior experience, and has participated in RoboCup Soccer Open together since 2019.

1.2 Current Year's Highlights

As planned from last year, our robot's hardware was sufficiently future-proof to be brought into 2021 with minimal changes, only making slight improvements to our

mirror's quality, as well as changing our IMU to get more reliable readings (teams interested in our hardware stages can visit our website or 2020 poster [1] for more information). This gave us the time to fully focus on software development, such as improving our camera system and the robot's strategy in areas of localisation and inter-robot coordination. We have also tested new strategies in our simulation program, which has helped point us in a more concrete direction for our future plans.

2 Hardware

2.1 Mechanical Design

Our robot is driven by 4 high torque 1700 RPM JMP-BE motors with our own 3D printed double layered omni-wheels. We have a double catchment area setup, with a dribbler on both sides and a solenoid kicker in front. Our dribblers use a 820kV brushless motor with a 1:2.5 gear reduction and has a spring suspension system hence allowing it to control the ball firmly at over 2000 RPM of backspin on the ball.

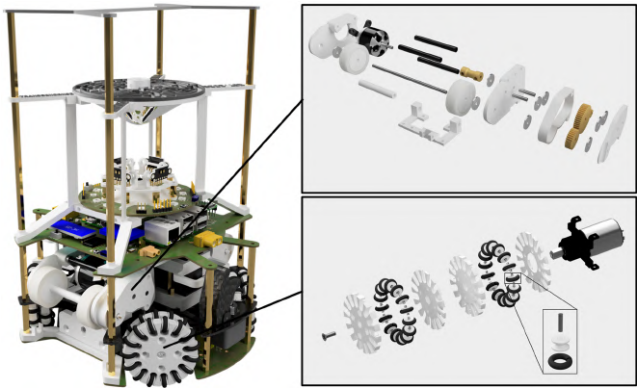
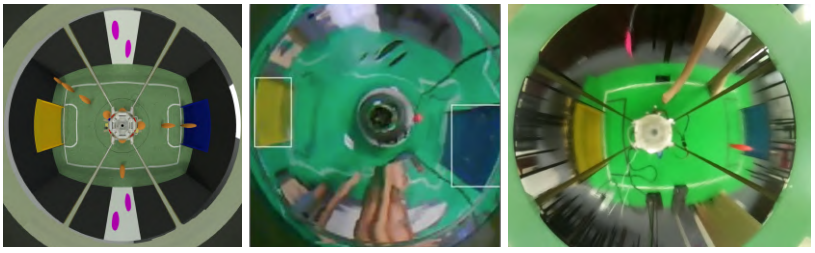


Fig 2. Transcendence 2021 Robot Design

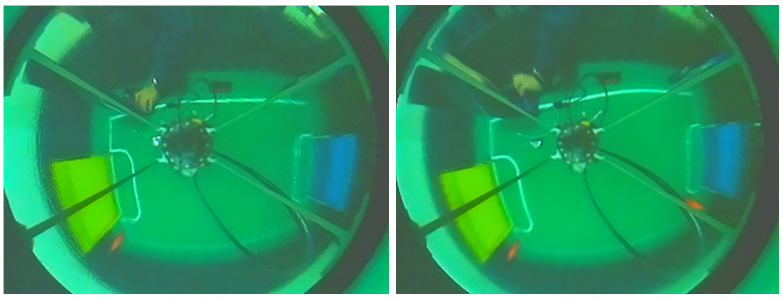
Mirror (Manufacturing) In 2020, we made our hyperbolic mirror by machining it from a CNC lathe, followed by multiple rounds of wet sanding and polishing, as inspired from MSL teams [2]. Unlike our previous attempts of moulding mirror sheets, this method produced a highly accurate mirror and was not warped. (Fig 3)



(a) **(b)** **(c)**

Fig 3. AutoCAD mirror render (3a), moulded mirror sheet (3b), machined mirror (3c)

Last year, we sanded our mirror up to 2000 grit. However, seeing that scratches near the upper part of the mirror led to distortions, we experimented with higher grit sandpapers of up to 4000, in 500 increments. We found out that sanding up to 3000 grit significantly reduced the scratches, but beyond that, we hit diminishing returns.



(a) **(b)**

Fig 4. Views of mirror sanded up to 2000 grit (4a) and 3000 grit (4b)

2.2 Electrical Design

Our robot houses all electronic components on its 3 structural PCBs, with almost all parts being integrated directly into the PCB as SMD components, which saves space and cuts costs. Each PCB has an STM32F103 chip which does low-level tasks, and streams data over to the Teensy via UART, achieving efficient task-parallelism.

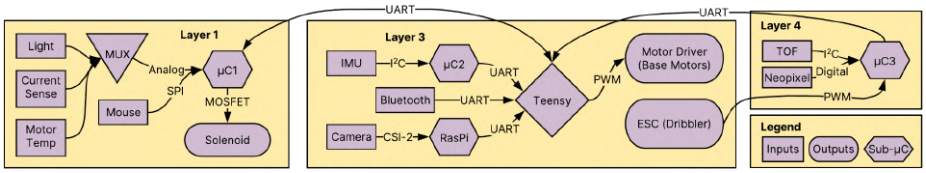


Fig. 5. Overview of main electrical components and interconnections

Experimenting with Different IMU In 2020, we opted for the Adafruit NXP Precision IMU with a Madgwick filter running on the STM32 chip. However, we found it to be highly susceptible to magnetic interference. This year, we switched to the BNO055 IMU, configured as a 6DOF IMU (no magnetometer), which supported a higher sampling rate and full resilience against magnetic interference. We came to this decision after conducting multiple benchmark tests (Table 1). Although the BNO055 (6DOF) had larger errors under stress testing, the lack of magnetic interference and faster response made it a suitable compromise. Besides, we have tested the robot, using this sensor, and it could remain pointed forward for far more than 10 minutes.

Table 1. Results of various tests of the 3 IMUs (averaged across 5 tests)

Description of test	NXP	BNO (9DOF)	BNO (6DOF)
Stationary test: Error in angle of stationary robot between start and end of 5 mins	0.0°	0.0°	0.0°
Spin test: Error in angle after robot spun	8.6°	5.6°	11.7°

around clockwise at 30 RPM for 5 mins			
Oscillation test: Error after robot oscillated 180° from its starting point at 0.5Hz for 5 mins	6.0°	5.4°	7.3°
Magnetic interference test: Max error in angle after placing bottom of 2nd robot next to IMU	19.4°	5.2°	0.0°
Time taken to reach steady state after a rapid angular displacement of 180°	300ms	50ms	30 ms

3 Software

3.1 Vision system

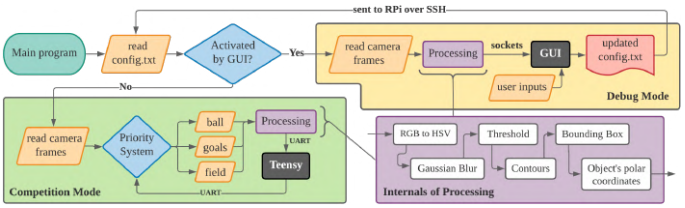


Fig 6. Transcendence camera system structure (RPi program)

Camera program Our camera system uses a Raspberry Pi (RPi) and RPi Camera V1. It is capable of tracking objects at 80 FPS with 13 ms of latency. To achieve this, we made many optimisations like coding in C++, which reduced latency as compared to interpreted languages like Python, while allowing us to enable true multithreading. We also run Arch Linux ARM on the RPi, drastically decreasing our boot time.

Graphical User Interface (GUI) We wanted to have an efficient plug-and-play debugging experience similar to the OpenMV IDE, hence we designed a client-side GUI using Qt5 and sockets over an ethernet connection. For 2021, we added the ability to send the camera programs over and make (compile) it with just one click. This further increases convenience since if there are any changes to the program, we do not have to manually transfer the files and SSH into the RPi to compile.

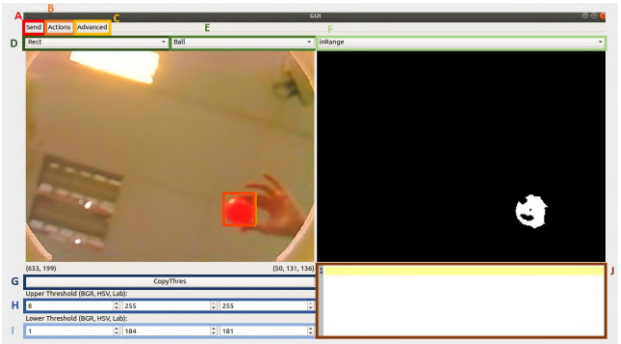
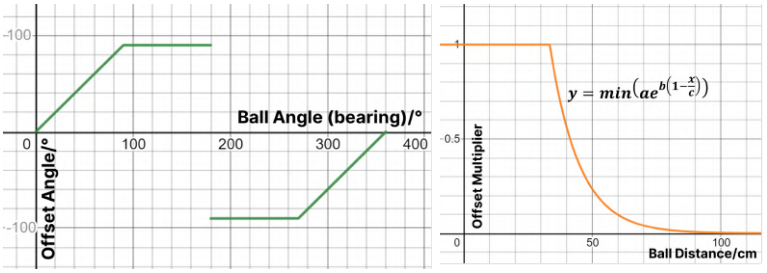


Fig. 7. Graphical user interface layout.

3.2 Robot Strategy

Striker As a striker, the robot actively chases the ball and shoots it into the goal. To catch the ball in its front catchment area, the robot orbits its way behind the ball by applying an offset angle (Fig. 8a) that decreases the further the ball is (Fig. 8b).

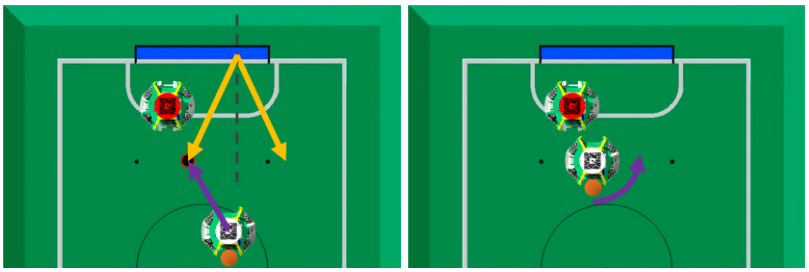


(a) **(b)**

Fig. 8. Graph of Offset Angle against Ball Angle (8a) and Offset Multiplier against Ball Distance (8b). Robot Angle = Ball Angle + Offset Angle * Offset Multiplier

Once the striker has possession of the ball in its front catchment area, it performs a similar orbiting motion towards the open area of the goal. When it is within a close range to the goal, it activates the solenoid, kicking the ball into the goal.

However, when the ball is within a ±25° range from the back of the robot, it will use the back dribbler instead to catch the ball with the same orbiting motion. To score from the back, the robot finds the 2 points that are 40cm away and 25° to the left and right of the open goal area (Fig. 9a). It moves to the closer point and flicks towards the open area, allowing the ball's backspin to carry itself into the goal. (Fig. 9b).



(a) **(b)**

Fig 9. Robot aligns itself 25° to the left of blue goal (9a), before rotating counter-clockwise to flick the ball in from its right side (9b)

Defender As a defender, the robot line tracks along the penalty area line to block the ball from its own goal. This is done by moving in the horizontal component of the robot-to-ball vector. A PID controller varies the robot's speed, with the process variable being the horizontal component's magnitude and the desired setpoint being 0. If the robot goes off the line or is too far away from its own goal, it will move back to the penalty area line, ensuring that it is always defending the goal.

Inter-robot cooperation The robots share information about the ball's position via Bluetooth. When the defender is in a more advantageous position to gain possession of the ball or to launch a counterattack, the defender will take on the role of a striker..

The robots also share their own localisation data (as described in Section 3.3), such that if any one of the robots is unable to see the ball, it can use the other robot's ball and location data to calculate the absolute position of the ball, and thus find the angle and distance of the ball relative to its own position.

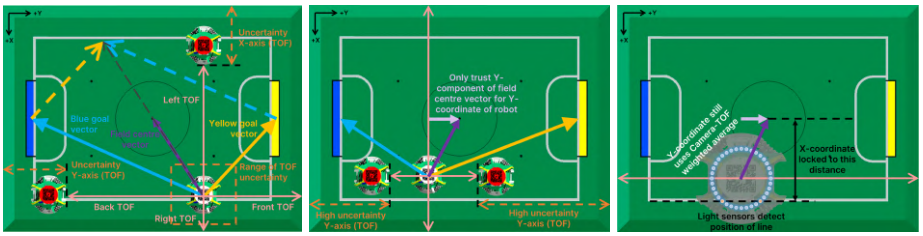
3.3 Localisation

TOF The 4 TOF sensors can measure the distance of the robot from the wall in the 4 cardinal directions, allowing it to calculate its position. Also, since we know the real dimensions of the field, we can add up the distances measured by opposite sensors and find the difference from the real dimension to estimate uncertainty in the reading.

Camera The robot calculates the midpoint of the resultant vector by adding the vectors to the yellow and blue goal, which corresponds to the field's centre. The vector from the centre to the robot is simply its position vector with the centre as the origin. Also, the smaller the goal area, the more likely it is to have been blocked by other robots, hence the confidence of the position decreases with smaller goal areas.

Fusing sensor data If the camera calculated position is within the range of uncertainty of the TOF based position, then each of their confidences is normalised and a weighted average position is calculated (Fig. 10a). Otherwise, if the two values are too different, the robot looks at the previously calculated position and further weights it in favour of the camera if it was nearer to the centre of the field (since it is more likely to see both goals clearly), or in favour of the TOF sensors if it was nearer to the corners (since it is more likely to see the wall without obstruction). Edge cases for when the robot cannot see both goals, or if the TOF uncertainty goes above a certain threshold, will make it solely trust the TOF or camera respectively (Fig. 10b).

Also, when the robot is on the line it "locks" the appropriate X/Y axis value into a range, purely based on the light sensors (Fig. 10c).



(a) **(b)** **(c)**

Fig 10. Various sensor fusion scenarios, where camera calculated position lies within range of uncertainty of TOF values (10a), a special case when TOF uncertainty for Y-axis is too high (10b), and using light sensors to increase accuracy of position (10c)

Uses Localisation is mainly used to allow the robot to travel to specific points on the field, e.g. when returning to a default "resting" location when no ball is detected, or positioning itself for an optimal back dribbler flick. It is also necessary to calculate absolute positions of field objects, which is a crucial stepping stone towards more advanced strategies as described in Sections 4 and 5.

4 Simulation

Inspired by RoboCup SSL, we created a simple 2D simulation with Python last year, and have continued using it to test out novel strategies that require complex information which we theoretically can obtain, but have not programmed for yet (e.g. opponent robots' positions). This allowed us to investigate the benefits of new features/strategies without actually having to implement them physically. The strategies we have been trying out aim to create more "offensive (active) defenders".

4.1 Tackling

A tackle play involves the "defender" robot chasing down the opponent robot that is closer to the ball (which would likely be the opponent's striker). The defender forces the opponent's striker into the closest corner of the field, thus reducing the game into a 1v1 striker vs opponent goalie situation, which would likely be to our advantage.

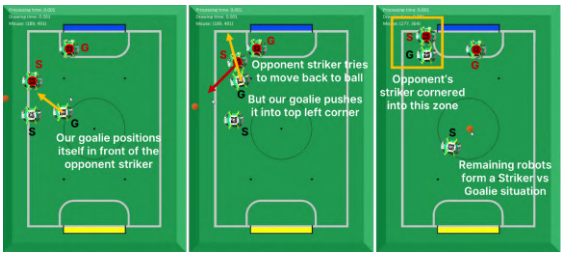


Fig. 11. Screenshots of a Tackling play in our simulation program

4.2 Blocking

Inspired from *American* Football, Blocking is where the defender positions itself between the ball and the closest opponent robot, in an attempt to protect the striker's possession or clear a scoring path for the striker during an offensive run.

Fig. 12. Screenshots of a Blocking scenario in our simulation program

5 Future Work

5.1 Increasing accuracy of locomotion

Our current motors, although fast and light, are rather inaccurate and release substantial amounts of heat due to its high current draw. Hence, we will be switching to brushless DC motors for 2022, likely that of the Maxon EC Flat series [3]. They are more efficient and have a smaller footprint, enabling us to fit within the new size limits more easily. In addition, they can be opted with pre-installed hall effect sensors, allowing us to accurately control the real RPM output of the wheels.

5.2 Advanced localisation methods

Inspired by RoboCup MSL, we plan to detect the lines, combined with the goal positions, to implement a particle filter for localisation through vision. [4] This is possible due to the high image quality from using a machined mirror, allowing us to see the white lines clearly. Furthermore, the object detection on the RPi is already threaded so adding the resource-demanding line detection function would have minimal impact on the FPS of object tracking. Besides, we will be switching to a Teensy 4.1 for 2022, which has 5x the clock speed of the Teensy 3.5 at 600 MHz [5], so running the particle filter on it should be computationally manageable.

We will also be working on vision-based robot detection. As depicted in Fig. 13, when detecting the green field colour, negative areas are formed around objects on the field. Through a process of elimination of known objects e.g. balls, goals, own robot, the remaining areas can be considered as other robots. This information can be used to implement more sophisticated strategies as mentioned in Section 4.

Fig. 13. Negative areas are formed around non-green objects

6 Conclusion

In this TDP, we have highlighted the various hardware improvements we have made since last year, showcased the main aspects of our robot's current program, as well as presented our other various innovative strategies that will fuel our future plans. In addition, in accordance with our team's philosophy, all our hardware and software designs are open-sourced on our GitHub repository [6] while more comprehensive information on the robots is available on our website.

References

1. Transcendence: 2020 RoboCup Poster, https://bozo.infocommsociety.com/assets/documents/poster.pdf
2. Lopes, G., Ribeiro, F., Pereira, N.: Catadioptric System Optimisation for Omnidirectional RoboCup MSL Robots. Lecture Notes in Computer Science. 318–328 (2012).
3. Maxon Group. Maxon EC Flat Motors, https://www.maxongroup.com/maxon/view/content/ec-flat-motors.
4. Lu, H., Xun, L., Hui, Z., Mei, H., Zheng, Z.: Robust and Real-time Self-Localization Based on Omnidirectional Vision for Soccer Robots. Advanced Robotics, 27, 799-811. (2013)
5. Stoffregen, P.: Teensy Technical Specifications, https://www.pjrc.com/teensy/techspecs.html.
6. Transcendence: 2020/21 RoboCup GitHub Repository, https://github.com/bozotics/robocup2020/