

Robot Club Toulon Team Description 2019

V. Gies, V. Barchasz, Q. Rousset, J.M. Herve, B. Talaron, C. Albert, G. Borowycz, N. Prouteau, Q. Baucher, S. Larue, Q. Anselme, J. Dussart, R. Lattier, S. Marzetti, J. Golliot, and T. Soriano

Université de Toulon,
Avenue de l'Université, 83130 La Garde, France rct@univ-tln.fr
Home page : <http://rct.univ-tln.fr>

Abstract. Robot Club Toulon Middle-size league (MSL) team is a new team aiming at participating in the RoboCup 2019. For a first season, our team has developed a whole robot team from scratch. This paper explains the most important developments, even though each part is something new for our team. As we have made an extensive use of other teams documentation for the conception of our robots, we also tried to share some new bricks with the MSL community such as a simulator for optimizing the kicking systems, a linear Kalman filter for positioning, or using a smart camera for image processing.

Keywords: RoboCup Soccer, Middle-Size League, Multi-robot, Electromagnetic kicker, Image Processing

1 Introduction

Robot Club Toulon is representing University of Toulon, France, in the RoboCup Middle Size League (MSL). The team is participating in the Middle-Size League for the first time this year.

Although we have no experience in the RoboCup, our team has been participating to several robot competitions for the last 5 years, with 4 titles in the French Institute of Technology National Cup (link to RTC results).

At the moment of writing this paper, RCT team consists of 2 PhDs, 4 MSc, 7 BSc, 3 staff members including 2 researchers in electronics and robotics and an engineer. Considering we are a new team, this paper describes shortly most of the parts of our soccer robots. More details about the robots can be found in the *Mechanical and Electronic Presentations*. In this paper, scientific improvements done during the last year are also presented.

2 Robot Platform

Our robots have been entirely designed by our team. They are strongly inspired by existing robots designs [6,3], using a 3-wheels omnidirectional platform having a pyramid shape, with a coil gun kicking system and a ball control.

2.1 Electronics

Architecture of the robot relies on a cortex composed by an **embedded computer** interfacing advanced sensors such as two LIDARs and an omnidirectional camera for positioning, scene analysis and collision avoidance, and communicating with a **peripheral board** for interfacing actuators and simple sensors as shown in Fig. 2. The kicking system is a **third board**, independent for development and safety reasons due to high voltage.

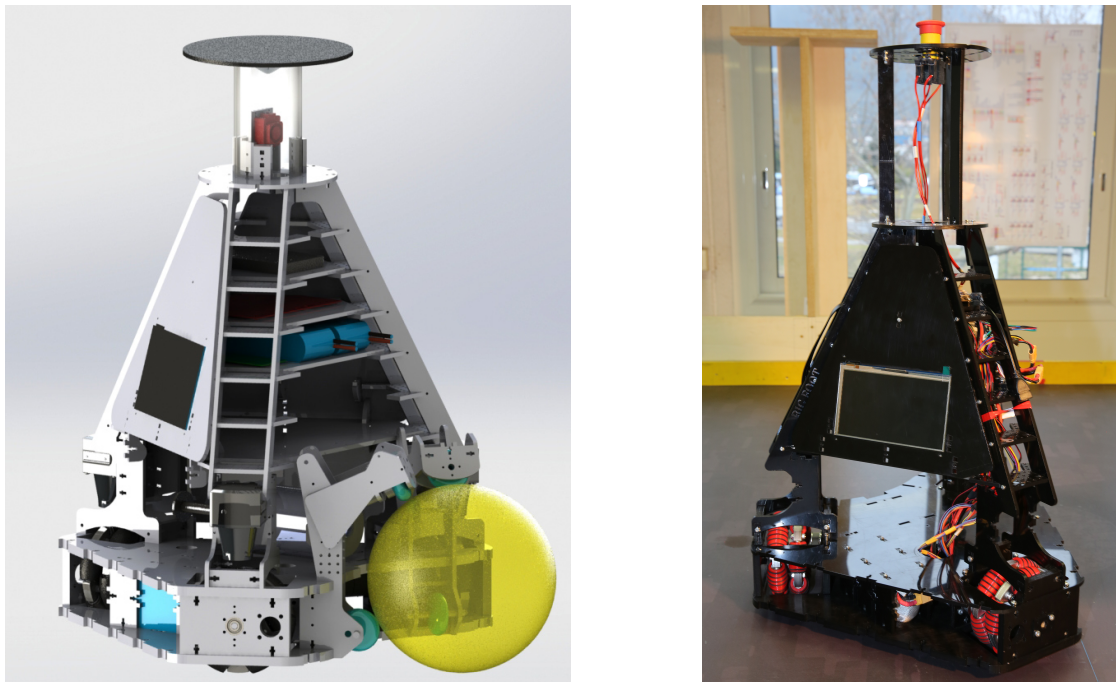


Fig. 1. Computer image of the 2019 robot of Robot Club Toulon Team in its latest version including a PMMA transparent tube at the top for the camera, and picture of the robot in an older version and without the ball control system.

This architecture is bio-inspired : the cortex of the system is an embedded computer able to perform complex tasks, whereas most repetitive tasks (autonomic nervous system) such as sensor management are using dedicated hardware from the peripheral board. This one embeds a Microchip DSP (*dsPIC33EP512GM310*) having hardware peripherals for multi-threading tasks at a low level, and dedicated circuits such as 32 bits counters with SPI interface (*LS7366R*) for decoding quadrature encoders signals from more than 2 quadrature encoders (maximum available on DSPs). Datas are exchanged between autonomic nervous system, the cortex and sensors through dedicated interfaces such as USB, SPI or UARTs.

An **embedded computer** is used for high level behavior coordination and processing such as artificial intelligence (AI). A *LattePanda Alpha* has been chosen, it is programmed in C#. This embedded computer interfaces the smart camera with an omnidirectional home-made mirror, and two SICK TIM561 LIDARs for collision and obstacles avoidance.

The **peripheral board** is a home-designed one having a PCB in 4-layers with a DSP from Microchip as main processor. This board is able to drive six 150W motors such as Maxon RE40 ones, 8 quadrature encoders and up to 20 digital or analog I/O. Basic sensors such as IR proximity sensors, ultra sonic telemeters, IMU and precision gyroscope (*ADXRS453*) are connected to these I/O.

Power for the propulsion and recharging kicking system is supplied by two 5600mAh 4S LiPo batteries, whereas power for electronics and embedded computer is supplied by two 2650mAh 4S LiPo batteries followed by four TRACO switching regulators (3.3V, 5V, 12V, 15V).

2.2 Hardware and mechanical features

Mechanical design of RCT robots is a 3-wheel omnidirectional robot driven by independent 150W Maxon RE40 motors having a gearbox ratio of 1:19. This platform is described in details in the *Team Mechanical*

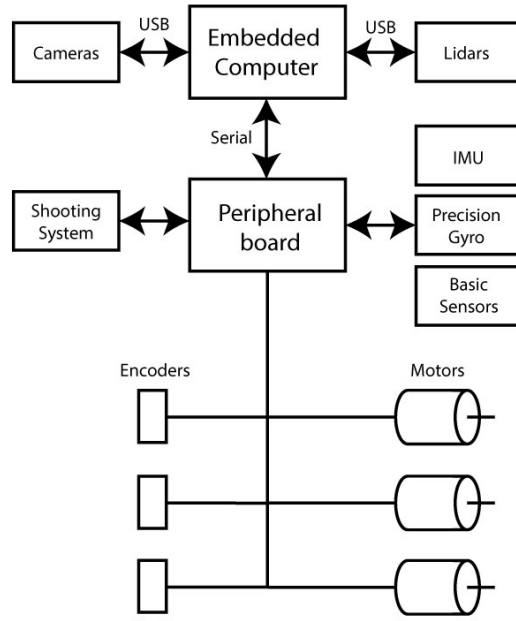


Fig. 2. RCT robots electronic bio-inspired architecture

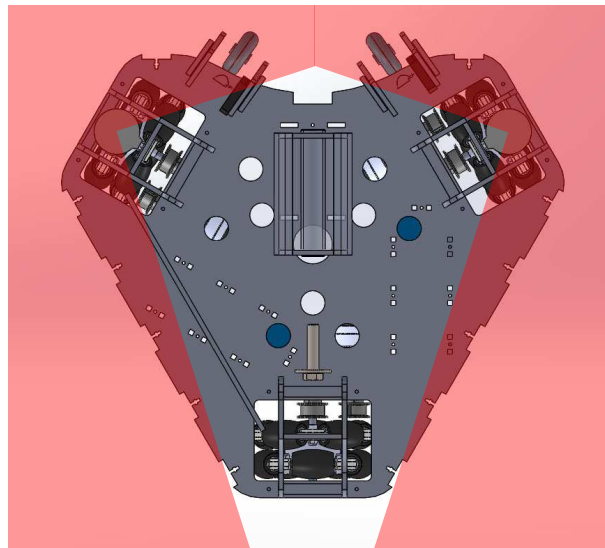


Fig. 3. Sectional view from top at LIDARs level. LIDARs active zones are in red.

Presentation paper.

Compared with other teams such as CAMBADA[3,2], there is nothing special in the mechanical aspects of our robots expect a strong design constraint for placing 2 LIDARs having the ability to see all around the robot. Because of that, space have to be empty around each LIDAR on 270° as shown in Fig. 3.

2.3 Kicking system

Inspired by CAMBADA and Tech United ones [6,3], our kicking system relies on a coil gun. Moving part is a 20mm of diameter iron bar sliding in a stainless steel tube. On the stainless steel tube a magnetic circuit has been added in soft iron on order to channelling magnetic field lines. Inside this magnetic circuit, a 1200 loops coil has been wound around the stainless steel tube using 1.25mm diameter isolated copper wire.

This kicking system has been designed and validated using a finite elements simulator as shown in Fig. 4, in order to calculate the inductor value for each position of the iron bar in the stainless steel tube. Then the differential equation characterizing the position and speed of the iron bar over the time as been solved numerically in order to find the best initial position of the iron bar for maximizing the kicking strength.

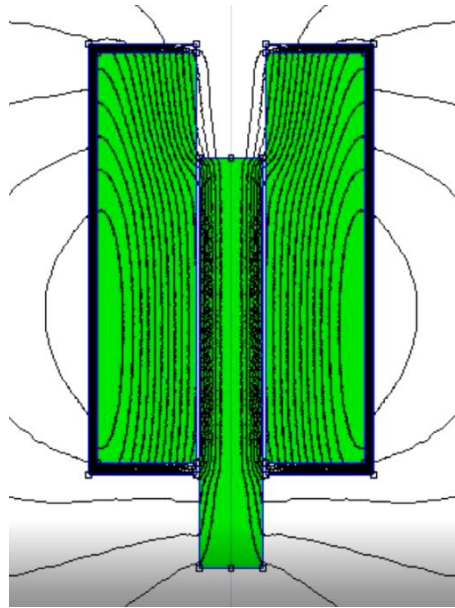


Fig. 4. Magnetic field simulation using a finite elements software.

3 Software

3.1 Perception and vision

Perception of the environment is done by redundant high semantic level sensors such as omnidirectional camera and LIDARs. This redundancy is a singular choice among other teams : it allows to make the most of the sensors on different situations. For example, omnidirectional camera image analysis is very useful when working on objects in a same geometrical plane (for example determining position using the soccer field lines), but less efficient when working with 3D objects far from the robot as shown in Fig.5.

At the opposite, objects in 3D are very well detected by LIDAR with a statistical error of ± 20 mm whereas objects in an horizontal plane will not be detected by a LIDAR working on an horizontal plane too.

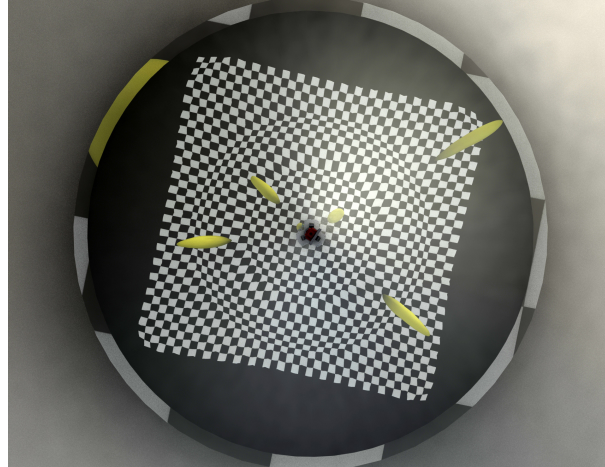


Fig. 5. Simulation image seen by the *Jevois* camera of the robot over a black and white checker board with yellow balls. The more camera is looking to a far distance, the more the balls are distorted. At the opposite, objects in a plane orthogonal to the camera and mirror revolution axis (such as the checker board) are not distorted.

For these reasons, combining an omnidirectional camera with a LIDAR is interesting. There is one drawback concerning the LIDAR : occlusions can occur when the ball or an opponent is close to the robot. In order to avoid that, a second LIDAR has been added in order to reduce these occlusion situations.

These high throughput rate sensors are connected to the LattePanda Alpha embedded computer in USB.

Omnidirectional vision system It relies on a custom calculated shaped mirror design by our team and achieved at the mechanical department of the Toulon Institute of Technology as shown in Fig. 6. It is designed to allow seeing the surrounding scene from 75cm atop the ground with a radius of 10m around the robot, with the smallest possible distortion, meaning that 2 equal distances in a same horizontal plane of a scene have to remain equal in the final image. Mirror has been calculated in software using a home-made finite elements model and simulated using Solidworks rendering tool has shown in Fig. 5.

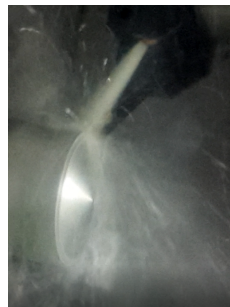


Fig. 6. Mirror making on a CNC production system at Toulon Institute of Technology.

System used for vision is a *Jevois* (<http://jevois.org/>) smart machine vision camera which is a combination of an OmniVision OV9653 1.3MP camera sensor, a quad-core CPU and a dual-core GPU. It embeds image processing and features extraction in the vision system itself. Using a microSD card, it can

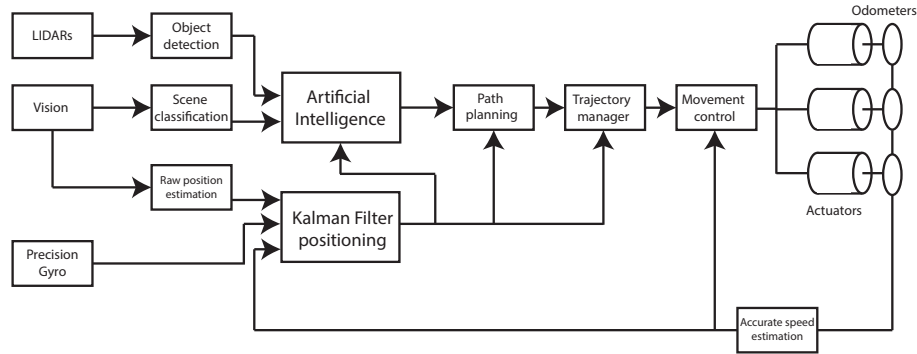


Fig. 7. Software architecture.

be loaded with open-source computer vision algorithms (including OpenCV, TensorFlow or Darknet). Output returns a processed image and data about extracted features on a serial link.

LIDARs As explained before, in order to add redundancy to the vision system, two SICK TIM561 LIDARs are used in each robot in order to have a 360° view of what is happening around the robots as shown in Fig. 3. Each LIDAR has a perception angle of 270°. Combining both of them allows to maintain an omnidirectional range image of the scene even in the case of a partial occlusion on one of the LIDARs.

LIDARs are used to predetermine a list of objects around the robot. This list is then cross-checked with the vision system analysis of the scene in order to determine obstacles such as opponent robots or the ball.

3.2 Positioning, path planning and trajectory generation

As explained in Fig.7, movement actions decided by the AI and relying on the accurate positioning calculated by Kalman filter are executed in three steps: path planning, trajectory Management and movement Control. Each of these steps is described in this section.

Positioning Probably the most important part of the robot software, accurate positioning is done using a Kalman filter. An extremely interesting example has been described in [4]. This implementation is a two-stage one using extended Kalman filters. In order to limit computation time and to let more CPU resources available for other tasks, we have designed a simple non-extended Kalman filter for our robot, having in input x_{camera} , y_{camera} , θ_{camera} , $\dot{x}_{odometry}$, $\dot{y}_{odometry}$, $\dot{\theta}_{odometry}$, $\dot{\theta}_{gyro}$.

In order to have a linear model, all the variables have to be calculated in the soccer field referential, resulting in a linear implementation of the Kalman filter.

This filter has been implemented and tested, leading to an error of less than 2 cm on a complex move iterated many times as shown in Fig. 8. A more detailed paper about it has been submitted for the RoboCup 2019.

Path Planning is necessary for determining the path to the current destination to be reached by the robot in order to follow the AI strategy. This path is calculated by AI using an A* path finding algorithm in order to check whether they can be reached or not in a defined time limit.

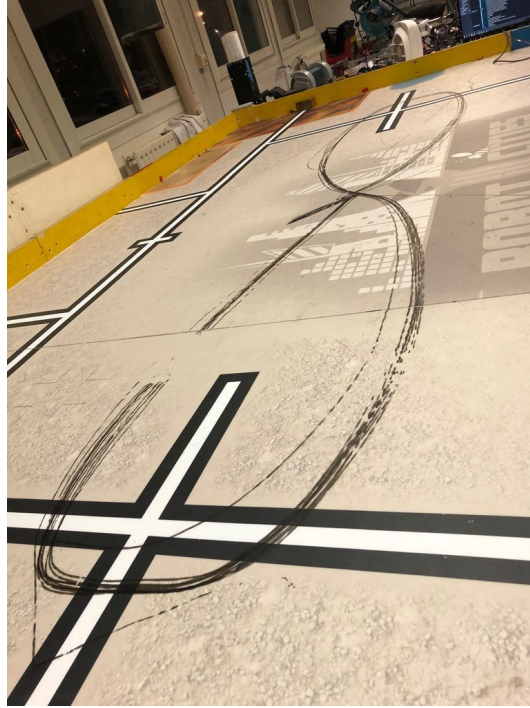


Fig. 8. Kalman filter validation : trajectories of a reduce size MSL robot travelling many times along a given path. Robot has been following many times a path with a maximum error of 2 cm around this theoretical path.

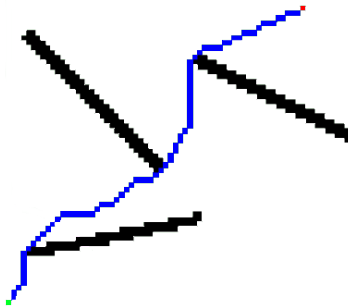


Fig. 9. A* algorithm example used for avoiding obstacles.

A* algorithm is fed in real time data from both LIDARs, image processing and positioning. It takes about $1ms$ to compute in *C#* a path on a $220*140$ grid having a cell size of 10 cm by 10 cm and used for mapping the MSL field ($22m*14m$).

Trajectory planning is used for obtaining a smooth movement using minimum jerk algorithms [5,1] and following the path returned by the path planner.

To minimize jerk, trajectories in X , Y and θ in the field referential are following a 6^{th} order polynomial determined by solving a set of 6 equations obtained by specifying initial and final conditions about position, speed and acceleration. Polynomials are calculated (it is only a matrix inversion) in real time for each update of the move to be done by the robot ensuring a smooth movement in every directions.

4 Conclusion

Participating in the RoboCup for a first year is a real challenge. Every thing is to be addressed in the same time : mechanics, electronics, embedded programming, AI, and a lot more including more unusual activities such as designing mirrors and learning to manufacture them or trying to find the best components for overcoming difficulties.

Thanks to the help of other teams, and especially Tech United (many thanks to Wouters who spent much time to answer our questions), it has been a great adventure for these 6 first months, and we are proud to have know a robot almost functional. It will be replicated as soon as we will receive all what is needed for 6 robots to make a team.

For the near future, the goal keeper will be designed very similar to the other robots, with the addition of a goal keeping system using very fast servomotors already used in other competitions, and a second *Jevois* camera for incoming ball detection.

For this year, we just expect to meet the RoboCup MSL standards to be able to participate. As it will be the first time, our goal this year is to fullfill the technical challenges in order to qualify for the tournament.

References

1. Trajectory generation with a minimum jerk trajectory, <https://mika-s.github.io/python/control-theory/trajectory-generation/2017/12/06/trajectory-generation-with-a-minimum-jerk-trajectory.html>
2. Azevedo, J., Cunha, B., Neves, A., Lau, N., Pereira, A., Corrente, G., Santos, F., Martins, D., Figueiredo, N., Silva, J., Cunha, J., Ribeiro, B., Sequeira, R., Almeida, L., Seabra Lopes, L., Rodrigues, J., Pinho, A.: Cambada, hardware description (01 2019)
3. Dias, R., Amaral, F., Azevedo, J.L., Azevedo, M., Cunha, B., Dias, P., Lau, N., Neves, A.J.R., Pedrosa, E., Pereira, A., Pinto, F., Silva, D., Silva, J., Sousa, E.: Cambada team description 2017 (2017)
4. Kon, J., de Molengraft, M.V., Houtman, W.: Planar pose and velocity estimation of a soccer robot. a two-stage kalman filter approach using odometry, vision and imu data. Bachelor End Project (2018)
5. Kyriakopoulos, K.J., Saridis, G.N.: Minimum jerk path generation. In: Proceedings. 1988 IEEE International Conference on Robotics and Automation. pp. 364-369 vol.1 (April 1988). <https://doi.org/10.1109/ROBOT.1988.12075>
6. Schoenmakers, F., Meessen, K., Douven, Y., van de Loo, H., Bruijnen, D., Aangenent, W., Olthuis, J., Houtman, W., de Groot, C., Farahani, M.D., van Lith, P., Scheers, P., Sommer, R., van Ninhuijs, B., van Brakel, P., Senden, J., van t Klooster, M., Kuijpers, W., , van de Molengraft, R.: Tech united eindhoven team description 2018 (2018)