CAMBADA'2018: Team Description Paper

R. Dias, F. Amaral, J. L. Azevedo, B. Cunha, P. Dias, M. Fiore N. Lau, A. J. R. Neves, E. Pedrosa, A. Pereira, F. Pinto, D. Silva, J. Silva

Intelligent Robotics and Intelligent Systems Lab
IEETA/DETI – University of Aveiro, Portugal

Abstract. This paper describes the CAMBADA Middle Size robotic soccer team for the purpose of qualification to RoboCup'2018. During the last year, improvements have been made in a significant number of components of the robots.
The most important changes include improvements on the hardware platform (wheels and grabber), the communication middleware, global obstacle tracking, leader election, set-plays and motor control.

1 Introduction

CAMBADA[1] is the RoboCup Middle Size League (MSL) soccer team of the University of Aveiro, Portugal. The project involves people working on several areas contributing for the development of all the components of the robot, from hardware to software.

The development of the team started in 2003 and a steady progress was observed since then. CAMBADA has participated in several national and international competitions, including RoboCup world championships (5^{th} place in 2007, 1^{st} in 2008, 3^{rd} in 2009, 2010, 2011, 2013, 2014, 2016 and 2017), the European RoboLudens, German Open (2^{nd} place in 2010), Dutch Open (3^{rd} place in 2012) and the annual Portuguese Robotics Open (3^{rd} place in 2006, 1^{st} in 2007, 2008, 2009, 2010, 2011, 2012, 2016 and 2^{nd} in 2013, 2014, 2015 and 2017). Moreover, the CAMBADA team achieved excellent results in the technical challenge of the RoboCup MSL: 2^{nd} place in 2008 and 2014, and 1^{st} place in 2009, 2012 and 2013. A 3^{rd} place in 2013, 2^{nd} places in 2012, 2015 and 2017, and 1^{st} place in 2011, 2014 and 2016 in the RoboCup Scientific Challenge were also achieved.

The general architecture of the CAMBADA robots has been described in [1, 2]. Basically, the robots follow a biomorphic paradigm, each being centered on a main processing unit (a laptop), which is responsible for the high-level behaviour coordination, i.e. the coordination layer. This main processing unit handles external communication with the other robots and has high bandwidth sensors, typically vision, directly attached to it. Finally, this unit receives low bandwidth sensing information and sends actuating commands to control the robot attitude by means of a distributed low-level sensing/actuating system.

[1] CAMBADA is an acronym for Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture.

This paper describes the current development stage of the team and is organised as follows: Section 2 briefly describes the hardware platform changes. Section 2.1 explains the improvements on hardware omni-wheels. Section 2.2 addresses the project of a new ball handling mechanism. Section 3 summarises the development of a new version of our communication middleware (RtDB v2). Section 6 shows the set-play engine implementation on the CAMBADA robots.

describes on-going work on the communications framework. And finally, Section 8 concludes the paper.

2 Hardware Platform

The CAMBADA team uses the same platform version since 2013, which has been thoroughly described in previous Team Description Papers. During 2017, the hardware improvements were aimed at the omni-wheels and the ball handling mechanism ("grabber").

2.1 Omni-wheels

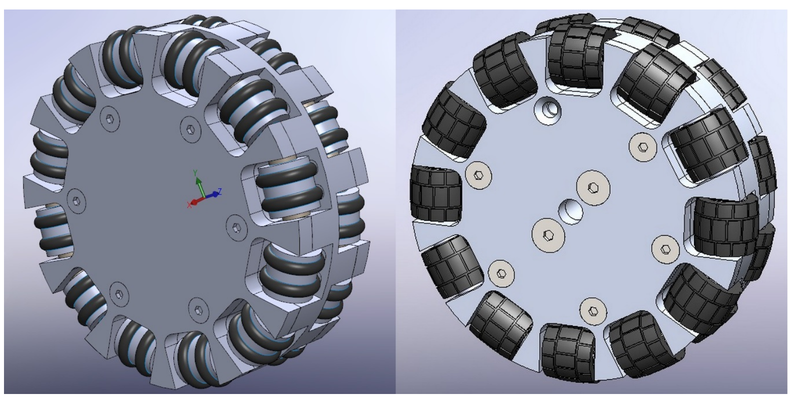


Fig. 1: New robot wheels.

New traction omni wheels have been designed. In the previous design (left side of figure 1), the free rolling rollers were based on a core aluminum cylinder upon which two rubber O-rings were responsible for the contact and traction of the wheels on the floor. With the new design (right side of Fig. 1), the roller has been replaced by a single structure, barrel shaped and made of hard rubber that follows the exact contour of the external diameter of the wheel. This solution results on a two-side improvement regarding the previous one: since the roller is now a continuous surface made of rubber, the friction coefficient between the wheels and the floor increases significantly, allowing a smooth control of higher acceleration patterns of the robot. On the other hand, since there is always a tangential curved shape of the wheel in contact with the floor, the overall structure of the robot performs a smoother movement, decreasing then level of vibration which, previously, had a significant impact on the vision system located on the higher level of the robot.

2.2 Ball Handling Mechanism

A redesign of the ball grabbers is also currently taking place. The main changes regarding the current version target the following most important aspects:

- Change the current hard link between the grabber and the platform to a new one that, by means a hard rubber sandwich structure will be able to sustain and absorb much more energy in situation of chock amongst robots.
- The maximum angular movement of the current grabbers when grabbing the ball is very short (around 8 degrees). Also, the translation the grabber according to the ball engaging direction were only of 10mm. By rearranging the location and initial angle of the grabber, these values will be increased to 16 degrees and 28mm respectively. This will allow to make the dumping system much more effective, reducing effectively the possibility of ball bouncing when approaching the grabbers at higher speeds.
- Finally, the omni wheel that has been used until know (of the shelf solution) has been redesigned to use a higher coefficient friction material in the free rollers, while changes to the chassis shape were made in order to reduce the possibility of contact between the ball surface and the chassis itself, a situation that sometimes happens with the off the shelf solution.

3 RtDB v2

A new version of our communication middleware has been developed, because some limitations have been found on the previous one, namely:

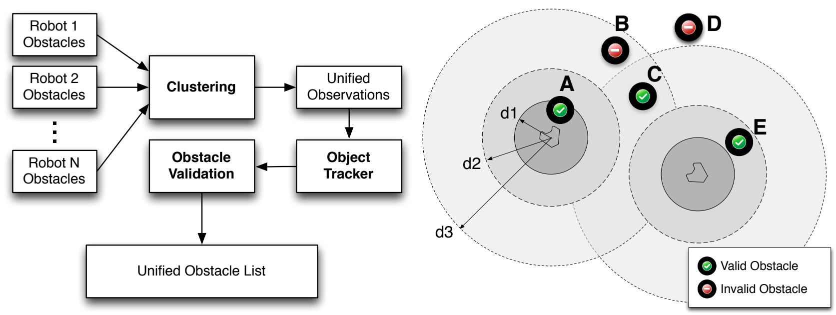
- Simple raw data structure binary storage in shared memory
- Inability to store dynamic items
- Static Configuration Scheme: a configuration file is used to define all local/shared structures
- Inability to dynamically (programmatically) create items in the RtDB
- Unreliable when agents are using different structures for the same key
- No compression on broadcast

In the new version, we to kept the blackboard logic, but with ability to store dynamic data structures and containers (for example, C++ std::vector and alike), add resilience to structure changes, no full source re-compilation when a structure changes. All this with a smooth integration with the previous API. All this with negligible impact on the duration of the operations.

We plan to release this new version soon on our public GitHub[2] repository.

[2] `https://github.com/CAMBADA`

4 Global Obstacle Tracker



(a) Merge obstacles from multiple agents (b) Example for obstacle validation

Fig. 2: Multi-object tracking with distributed sensing.

The Middle-Size League provides an excellent test-bed for multi-robot cooperative sensing - it consists in a semi-structured and partially observable environment, similar to one found in the real world. We have been developing a multi-object tracking architecture for our team, but general enough to be adapted to other multi-agent systems [3]. In this architecture, each agent collects information about the environment and, by merging information from all agents, the team as a whole gets a better global understanding of the world state.

5 Leader Election

One of the main focus of this league is on distributed planning - the robots communicate through Wi-Fi with strict bandwidth limitations, there are no external sensors, so most computations must be performed locally on the robots. However, distributed planning sometimes leads to conflicts that do not exist when there is a centralised decision - for example, situations where two agents decide to assume the same role or situations where the planned motion of one robot crosses another robot path. With unreliable network conditions, these kind of situations tend to occur more often. On the other hand, centralised approaches have the risk of collapsing the whole system if the master fails. The adopted solution was to get the best of both worlds with an hybrid solution: centralise the decision to avoid the conflicts, but automatically negotiate the agent that will perform the decision (the "leader") as a fail-safe, to overcome the deadlock problem if the master fails.

In the particular case of CAMBADA in the MSL, we are centralising the role selection on the coach to avoid conflicts. With our leader election system integrated with our RtDB, a (random) leader is automatically selected among the playing robots who takes over the task that was being carried by the coach when it fails.

More information about this leader election system will be released on future publications.

6 Set-plays

In the context of an internship and in collaboration with Tech United, we have implemented a set-play engine in the current CAMBADA architecture.

Fig. 3: The setplay configuration tool

With this set-play engine, we are able to create and configure a set of set-plays for different situations using the configuration tool depicted in Fig. 3. The architecture was adapted from a previous work on the FC Portugal simulation league team [4]. The tool allows the user to create step-based set-plays with associated conditions to control the transitions between these steps.

7 Motor Control

Some effort has been put in identifying the model of the robot and to test various control solutions on the CAMBADA robots in order to improve the general motion behaviour of the robot.

Finding the mathematical model is important because it can be used for future work and taken into consideration for higher level control. Two different controllers were designed: PI and RST. Unfortunately, the RST controller has not been able to guarantee the intended performance for which it was designed. An RST controller relies a lot on the model of the system and in the tests we

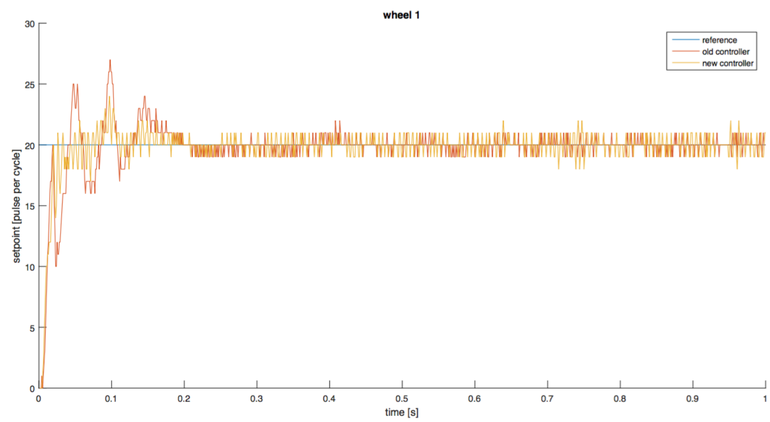


Fig. 4: Comparison between new controller and previous controller.

were not able to stabilise the system - even though much effort was put in understanding the root of this problem, the system kept oscillating. Following, the PI controller with anti wind-up has been able to guarantee similar rise time with lower overshoot when compared with the previous version of the CAMBADA low-level controller (Fig. 4).

8 Conclusions

This paper describes the current development stage of the CAMBADA robots, both in the hardware platform and at the software level.

Several improvements have been carried out on the hardware platform (wheels and grabber), the communication middleware, global obstacle tracking, leader election, set-plays and motor control.

References

1. A. Neves, J. Azevedo, N. Lau B. Cunha, J. Silva, F. Santos, G. Corrente, D. A. Martins, N. Figueiredo, A. Pereira, L. Almeida, L. S. Lopes, and P. Pedreiras. *CAMBADA soccer team: from robot architecture to multiagent coordination*, chapter 2. I-Tech Education and Publishing, Vienna, Austria, In Vladan Papic (Ed.), Robot Soccer, 2010.
2. José Luís Azevedo, Bernardo Cunha, and Luís Almeida. Hierarchical Distributed Architectures for Autonomous Mobile Robots: a Case Study. In *ETFA2007- 12th IEEE Conference on Emerging Technologies and Factory Automation*, volume 1-3, pages 973–980, Patras, Greece, September 2007.
3. R. Dias, B. Cunha, E. Sousa, J. L. Azevedo, J. Silva, F. Amaral, and N. Lau. Real-time multi-object tracking on highly dynamic environments. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 178–183, April 2017.

4. Nuno Lau, Luis Seabra Lopes, Gustavo Corrente, Nelson Filipe, and Ricardo Sequeira. Robot team coordination using dynamic role and positioning assignment and role based setplays. *Mechatronics*, 21(2):445 – 454, 2011. Special Issue on Advances in intelligent robot design for the Robocup Middle Size League.