CAMBADA'2020: Team Description Paper

R. Dias, F. Amaral, I. Angelico, J. L. Azevedo, B. Cunha,
P. Dias, C. Gomes, N. Lau, P. Martins, A. J. R. Neves, E. Pedrosa,
A. Pereira, F. Pinto, P. Rodrigues, D. Silva and J. Silva

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

Abstract. As part of the qualification materials to apply for RoboCup 2020, this paper describes the most recent improvements made by the CAMBADA Middle Size robotic soccer team. During the last year, improvements have been made in a significant number of components of the robots. The most important changes include changes at the hardware - vision and gateway -, a Set-Play effective generator and a new logging tool.

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.

Overall, CAMBADA has achieved **9 1st places**, **9 2nd places** and **9 3rd places** in all competition tournaments it has participated in. Also, it has won **3 Technical Challenges** (also, 1 second place) and **3 Scientific Challenges** (also, 3 second places and 1 third place).

The team has participated in:

- **14 RoboCup championship competitions**:
  RoboCup 2019, RoboCup 2018, RoboCup 2017, RoboCup 2016, RoboCup 2015, RoboCup 2014, RoboCup 2013, RoboCup 2012, RoboCup 2011, RoboCup 2010, RoboCup 2009, RoboCup 2008, RoboCup 2007, RoboCup 2006
- **16 editions of the Portuguese Robotics Open**:
  RoboCup Portuguese Open 2019, RoboCup Portuguese Open 2018, RoboCup Portuguese Open 2017, RoboCup Portuguese Open 2016, RoboCup Portuguese Open 2015, Robotica 2014, Robotica 2013, Robotica 2012, Robotica 2011, Robotica 2010, Robotica 2009, Robotica 2008, Robotica 2007, Robotica 2006, Robotica 2005, Robotica 2004

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

– **Other 5 local events**:
RoboCup European Open 2016, RoboCup IranOpen 2014, DutchOpen 2012, GermanOpen 2010, DutchOpen 2006

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.

This paper describes the current development stage of the team and is organised as follows: Section 2 briefly describes the hardware platform changes. Section 3 introduces CR7 - The CAMBADA Set-Play Engine for dynamic set-plays generation. Section 4 briefly describes a new logging tool, CameLog, devoted to in detail analysis of data from all robots through out the full games. Section 5 introduces the recent updates performed by CAMBADA in the RefBox for 2020. Finnaly, section 6 wraps out this paper.

2 Hardware Platform

2.1 Vision System Redesign

CAMBADA's current vision system presents a series of challenges specially if we look into the future expected increase in field size. Furthermore, the current vision system is already five years old, uses vertical bars that hold the mirror made of titanium but time has shown that, despite the good behaviour of this structural solution, flexibility characteristics of these bars seams to decrease over time. On the other hand, the design of the mirror assumed the assumption that the field size would remain for a long period at the previous 18m x 12m. It uses an hyperbolic equation from which, after a calibration procedure a distance map on the floor plane similar is obtained.

This distance map can, however, be misleading. In fact, as we move away from the robot, and due to the hyperbolic design of the mirror, degradation of space resolution occurs at a very significant rate. As a result, current mirror provides a very high resolution close to the robot, but a very poor one when we move away. Fig. 1 a) shows the similar eagle eye view, but now presenting only the actual existing pixels. Furthermore, it has been repeatedly demonstrated that during a stressful competition, and after being hit by the ball several times, the mirror supporting system will slightly get misaligned over time. Extrinsic parameters may, therefore, be no longer valid after 3 or 4 games compelling the team to recalibrate the vision system repeatedly throughout the competition Fig. 1 b).

An effort was therefore taken in order to:

– design a more stable mechanical solution for the catadioptric system;

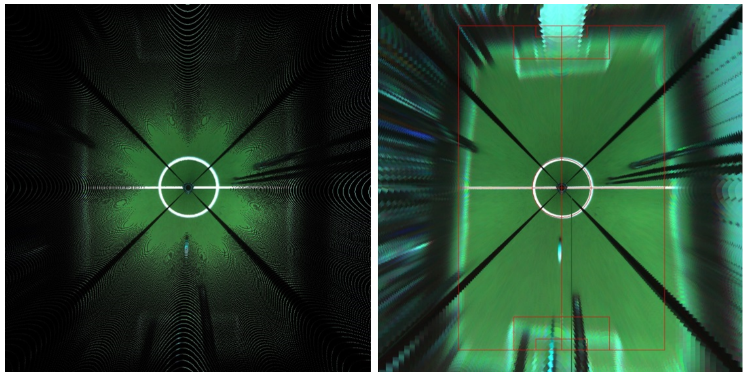

Fig. 1: a) pixel resolution as f(distance). b) 0.5^o error effect in the distance map.

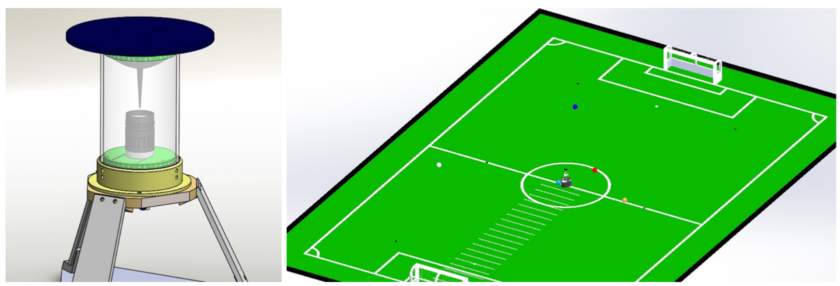

Fig. 2: New hardware structure and simulated full size MSL field with a robot at the center.

– redesign the mirror so that distribution of spatial resolution may enhanced.

To achieve this last goal a set of objectives were set regarding the equation of the mirror surface:

– to reduce significantly the resolution close to the robot;
– to reduce the size of the robot body on the image enhanced;
– to increase resolution at higher distances by trying to keep resolution degradation more linear while still being able to see above the floor at least 1m.

Regarding the mechanical solution the approach was basically to follow well proven solutions already implemented by other teams (Fig. 2). To design the new mirror, a simulated full size MSL field was used (Fig. 2).

A set of new equations, including 3rd order and 4rd order polynomial terms were developed and iteratively adjusted trying to fulfil the above specified requirements. The simulated results can be seen in Fig. 3. The results obtained with the more complex surface, based on a square root of a 4rd order polynomial Equation 1 are very promising.

$$f(y) = \sqrt{1000 + (k_1 * (x + a))^2 + (k_2 * (x + b))^3 + (k_3 * (x + c))^4} \qquad (1)$$

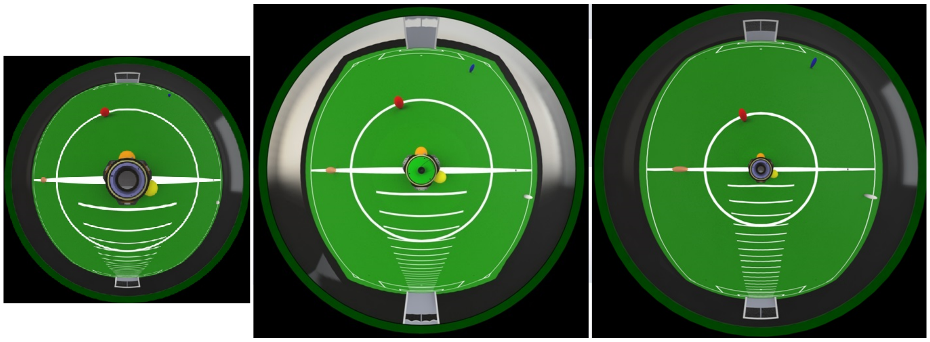

Fig. 3: Simulated images obtained from a) original mirror b) 3rd order term equation c) 4rd order term equation.

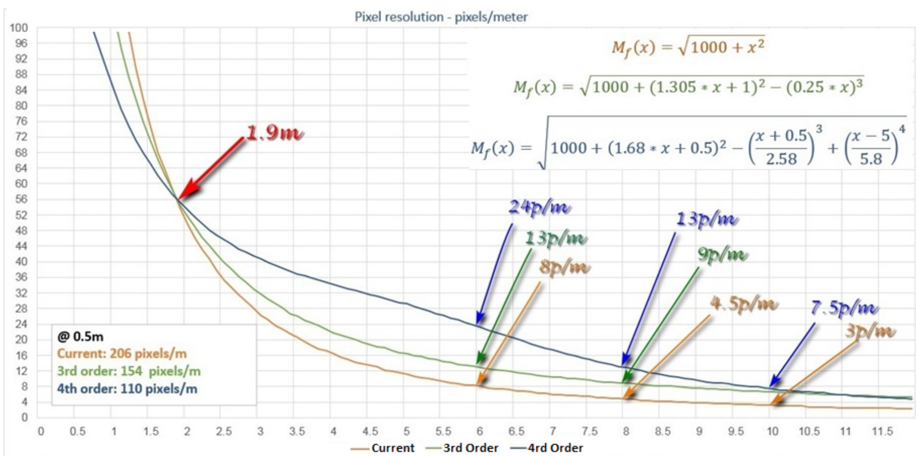

Fig. 4: Pixel resolution, in pixels/meter as a function of the distance to the robot centre for the 3 tested equations.

Among the clear advantages, shown in Fig. 4, the following can be highlighted:

- decrease of resolution beyond 3m from the robot is almost linear;
- resolution at a distance between 4.5 and 8.5m from the robot is 3 times higher than with the current solution (e.g. 24pixels/m @6m versus 8pixel/m);
- at the limits of the field (11m), resolution is still more than twice the current resolution;
- the mirror still allows to see around 1.5 m above the ground level at its horizon.

2.2 Ethernet-CAN Gateway

The low-level sensing/actuation system, which is composed of a set of micro-controllers interconnected by means of a CAN bus, connects to the high-level decision layer through a gateway module (see Fig. 5). A new gateway module was developed which uses an Ethernet connection and UDP protocol to implement the communication with the PC. At the communications level, the new solution allows for a reduction of the end-to-end delay time by a factor of approximately 20 when compared to the USB-based solution. Another benefit, to be fully understood in the coming MSL tournaments, is the expected improvement of the whole system reliability due to a more robust physical connection to the PC.

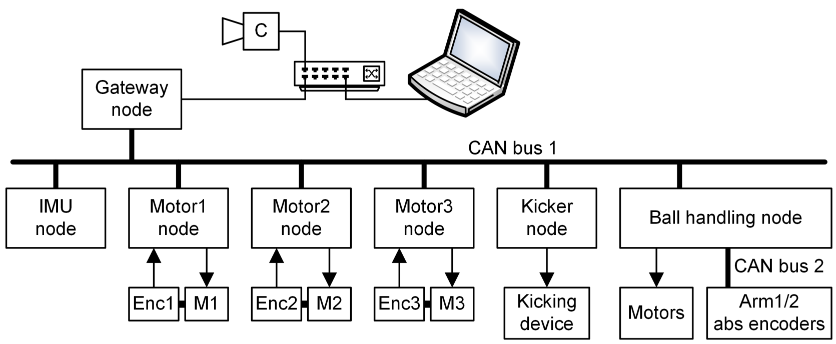

Fig. 5: The CAMBADA hardware architecture.

3 CR7 - The CAMBADA Set-Play Engine

In 2014, the CAMBADA agent software architecture has been redesigned to address this issue and turn the CAMBADA architecture into one that could be easily changed without compromising the final result - something that the previous architecture failed in providing. In order to achieve the best results in the competitions, it is utterly important that the software architecture is modular and flexible to allow the deployment of different strategies for different teams.

With a matured modular software architecture in place and making use of the enhanced world model with features implemented over the last years, it was possible to develop a new setplay engine - the CAMBADA Robot 7 (CR7) engine.

Our CR7 set-play engine allows us to create and configure a set of set-plays for different situations using a visual configuration tool. Built from scratch and using some concepts of the aforementioned architecture, this new framework has significant improvements, especially focused on the usability. It infers the transition conditions from the actions defined by the end-user (instead of requiring the user to define all the conditions manually). This way, the user has the ability to quickly create set-plays without worrying about consistency.

We integrated this engine into the CAMBADA agent (which is a process running in every robot) and also the coach (process running in the basestation computer). Because of how the previously presented Leader Election algorithm works in CAMBADA, the coach will be the selected as a leader by default and the robots will synchronise with the coach Setplay Manager execution state. If the coach process fails or is not running, one of the robots will act assume the leadership and manage the Setplays for the whole team.

The CAMBADA Basestation tool has also been updated to display a preview of the running Setplay while the robots are executing it - it shows target positions, margins, pass-line clear status, etc.

We have successfully used the CR7 engine in the Portuguese Robotics Open 2019 with very satisfactory results - some statistics are being evaluated at the time of this writing.

4 CameLog

With last year's introduction of RtDB v2, and the ability to store and disseminate dynamic data structures, the way was paved to allow the creation of a dynamic logging tool for the team.

CameLog is being developed as a daemon, installed and running on each agent computer. It supports configuration of which RtDB fields we want to log and it can be configured with two different modes:

- Asynchronous - it has an internal timer that periodically gets a snapshot of the RtDB items to store.
- Synchronous - the CameLog process snapshot is triggered synchronously from within the team process pipeline.

The structure of the logs makes use of the MsgPack binary serialization features, similar to the RtDB format. On each snapshot, a header is added, containing the timestamp and size of that snapshot, to allow effective decode of the log file.

The saving of information is performed in smaller chunk files of N minutes (currently with approximately 1 minute long) so that the log can be recovered with a maximum N minutes loss. Furthermore, a circular buffer of files is defined to hold around the last 1 hour of log.

The recovery of the log is performed through our basestation, where we can select the initial and final moment for gathering data. When performing this task, we do not want to assume clock synchronization among the several agents computers. Thus, the gamestate over the game time is used to create a "profile" that allows us to identify a pattern to be able to pinpoint common points. These patterns should be aligned by maximum similarity, thus allowing to estimate the time offsets for each agent, relative to the basestation clock, to guarantee that we crop the log within the same time frame in every agent (as illustrated in Fig. 6).

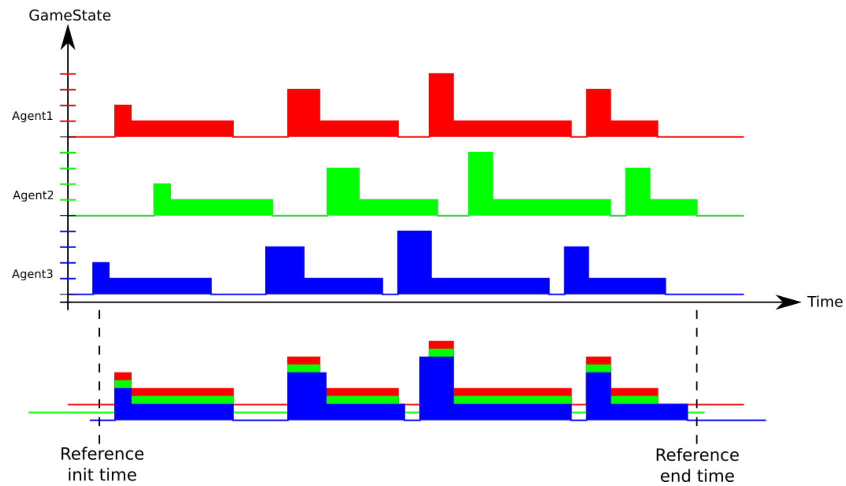

Fig. 6: Game state profiles over time.

5 MSL Referee Box Update

According to what have been decided during the last MSL workshop, CAMBADA has updated the RefBox to a new version. This version sends to the BaseStations using a JSON format. The format send by the RefBox no longer uses the single char format. Instead, a JSON packet is sent to both BaseStations for each command delivered by the RefBox. This JSON packet has 2 fields when sending the state game and 3 fields when needs to indicate the number of the destination robot. The fields to send information are:

- **command:** describes the command sent by the RefBox, for exemple: KICK-OFF, STOP, START, etc.
- **targetTeam:** Multicast Address of the team for destination command. If the command is destinated to both teams, the field is an empty string.
- **robotID:** this field only is sent in case of substitution. Sends the number of identifications to a single robot to run the command.

Furthermore , the colors are no longer relevant. The teams are now identified as “Left” or “Right” as seen from the RefBox position. Their names still appear on the RefBox and Referee/Audience clients

6 Conclusions

This paper described the latest scientific and technical developments of the CAMBADA team, both in the hardware platform and at the software level. The most important changes include redesign of the vision system, design and adoption of an Ethernet gateway among all hardware modules, improvements on the CR7 set-play design tool focused on its usability and its integration into the CAMBADA agent. The new dynamic logging tool for the team, named

Fig. 7: Version 2.0 of refbox.

CameLog, was also introduced and described as well as its particular multi file approach and practical time synchronization among logged data. Finnaly, as a consequence of the decisions of the last MSL workshop, upgrades to the RefBox MSL tool was also presented. With the all these improvements, CAMBADA expects to stay one of the most competitive teams in the fast-growing RoboCup Middle-Size League.

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.