Portuguese Team: Team Description Paper for RoboCup 2013

A. J. R. Neves[1], N. Lau[1], Luís Paulo Reis[2], Bruno Pimentel[1], João Silva[1], Alina Trifan[1], Nima Shafii[2], Vasco Santos[1], Sanaz Zadegan[1], Sanaz Bahmankhah[1]

[1]IEETA/DETI – University of Aveiro, 3810-193 Aveiro, Portugal
[2]LIACC - University of Porto / UM – University of Minho, Portugal

Abstract. The Portuguese Team intends to participate in the Standard Platform League at the RoboCup 2013. The team is composed of researchers mainly from University of Aveiro, some of them with experience in several RoboCup leagues (simulation 2D and 3D, Middle Size League, rescue simulation, mixed reality, @home), and with cooperation with researchers from Universities of Minho and Porto. This document briefly describes the team for the purpose of qualification for RoboCup 2013.

1 Introduction

Portuguese Team is a RoboCup Standard Platform League (SPL) team composed mainly by members of University of Aveiro, some of them part of the simulation mixed team FC Portugal and from the Middle Size League team CAMBADA. It works in cooperation with researchers from the Universities of Minho and Porto, particularly members of the FC Portugal simulation teams, from which some base methodologies were imported.

Based on previous experience, Portuguese Team developed an agent similar to the one developed for FCPortugal 3D and uses a similar distributed architecture as the one used in the CAMBADA robots. This distributed architecture is based on several processes, namely Communications, Vision, Agent and Localizer, centered in a Real-time database. The agent is based on several modules, each one with a specific purpose: WorldState, AgentModel, Geometry, Optimization, Skills, Utils, Strategy and DeviceManager.

This document describes the current development stage of the team and is organized as follows: Section 2 describes the general architecture of the software running on the robot. Section 3 describes the vision system developed for the NAO robot. Section 4 describes general implementation lines of the agent. Section 5 enumerates and describes several tools created for debug and configuration purposes, to help in the team development. Finally, Section 6 presents some conclusions, followed by a few of the main related publications.

Statement of Commitment: Portuguese Team hereby commits to participate in the 2013 RoboCup competition if it is qualified for participation with a team formed mainly by members of University of Aveiro, with possible support of members from University of Minho.

Team Constitution (human): The Portuguese Team is currently formed by the following members: João Silva (Team Leader, PhD student-CAMBADA), Luis Paulo Reis (PhD-FC Portugal), Nuno Lau (PhD-FC Portugal/CAMBADA), Bruno Pimentel (PhD), António J. R. Neves (PhD-CAMBADA), Nima Shafii (PhD Student-FC Portugal), Alina Trifan (PhD student-CAMBADA), Vasco Santos, Sanaz Zadegan, Sanaz Bahmankhah (Students).

Most of the team members are PhD or PhD students with strong background on cooperative robotics and a large number of RoboCup participations in the context of FC Portugal, CAMBADA and/or 5DPO teams.

Team Constitution (NAOs): We currently have 3 H21 NAO v3.3 robots, but we intend to acquire a pack (of 5 or 6) H25 NAO v4, after January.

2 Software architecture

The software developed for the Portuguese Team robots uses a similar distributed architecture as the one used in the CAMBADA robots [1]. This distributed architecture is based on several processes, namely Communications, Vision and Agent, centered in a Real-time database (RtDB).

Following the CAMBADA software approach, the software used in the robot is also distributed. Therefore, three different processes are executed concurrently. All the processes run on the robot's processing unit in Linux. Furthermore, a NAO qi module is also implemented, to handle the communication between the agent and the robot DCM (Fig. 1).

Inter-process communication is handled by means of a RtDB, implemented as a block of shared memory. The RtDB is divided into two regions, the local and the shared one. The local region allows communication between processes running

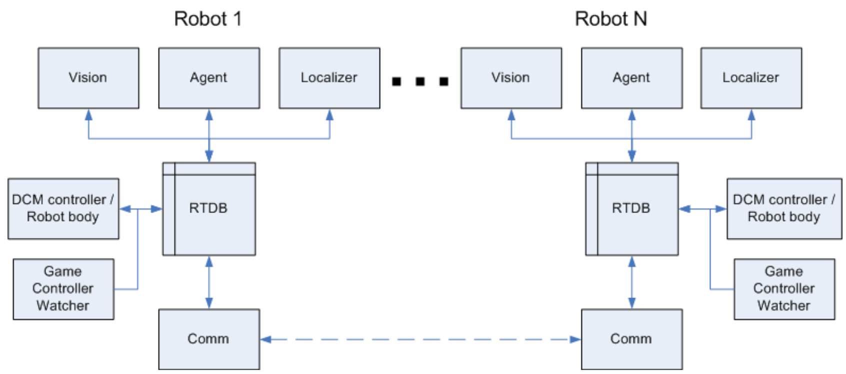


Fig. 1: Diagram of the software running on the robots.

on the robot. The shared region implements a Blackboard communication paradigm and allows communication between processes running on different robots. All shared sections in the RtDB are kept updated by an adaptive broadcasting mechanism that minimizes delay and packet collisions.

The processes composing the Portuguese Team robot software are:

- **Vision**: is responsible for acquiring the visual data from the robot cameras.
- **Agent**: is the process that integrates the sensor information and constructs the robot's worldstate, taking the decisions based on this information.
- **Communications**: handles the inter-robot communications, receiving the information shared by other robots and transmitting the data from the shared section of the RtDB.
- **gcWatcher**: listens to the game controller and writes all the referee information on the RTDB.
- **Localizer**: is responsible for running the localization algorithm whenever the agent requires it, through RTDB communication.

3 Vision

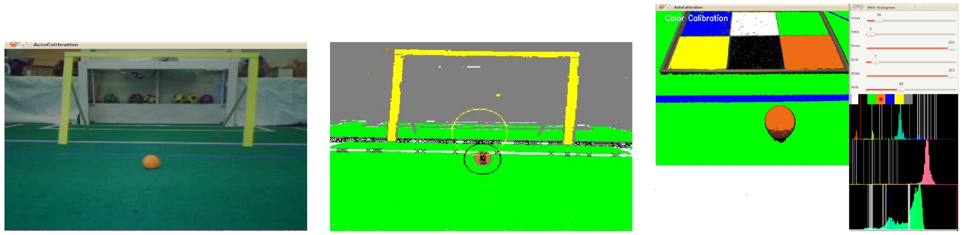
The architecture of the vision system can be divided into three main parts: access to the device and image acquisition, calibration of the camera parameters and object detection and classification [2]. Moreover, apart from these modules, two applications have also been developed either for calibrating the colors of interest (NaoCalib) or for debugging purposes (NaoViewer). These two applications run on an external computer and communicate with the robot through a TCP module of the client-server type that we have developed. The current version of the vision system represents the best trade-off that the team was able to accomplish between processing requirements and the hardware available in order to attain reliable results in real time.

Having the possibility of running the vision module as a server, the two applications that we have developed, NaoCalib and NaoViewer can act as clients that can receive, display and manipulate the data coming from the robot. Thus, NaoViewer is a graphical application that allows the display both of the original image as well as the corresponding index image containing the validation marks for each object of interest that was found. This application was essential in terms of understanding what the robot "sees" since NAO does not have any graphical interface that allows the display and manipulation of images. Also considering the limited resources of the robot the choice of building a graphical interface on the robot was out of the question. NaoCalib is a very helpful application that we developed for the calibration of the colors of interest (see Fig. 2).

4 Agent

The architecture of the agent is modular and with similarities with the agents used on both the simulation league and the MSL league. It starts by integrating the robot body information into an agent model and then uses it to build a world model. The creation of the world model is based on work developed for the CAMBADA team [3]. Afterwards, the agent will evaluate its conditions and the ones from the world (including information shared by its team mates) and decide how to act, filling a shared memory with all the data necessary for that action to occur.

The main modifications from the simulated agent were in the low level communications. In simulation, the agent communicates with the server to send the actuator values and receive the sensor values. In the real robot, instead of communicating with a server, the agent writes the actuator values and reads the sensor values from the shared memory.



(a) (b) (c)

Fig. 2: In (a), an image captured by the NAO camera. in (b), the same image with the colors of interest classified. The marks over the color blobs represent the detected objects. In (c), an example of the classification of the colors of interest, by means of the NaoCalib application.

The high level behaviors can be ported from the simulated robot to the real robot without changes, as long as the low level behaviors are developed for both the simulated and the real robot. When adapting the behaviors from simulation to a real robot we tried to keep the architecture of the agent similar on both [4–6].

Several walking behaviors developed and optimized for the FCPortugal 3D simulation team [7] are being adapted and tested. On the last competition, the existing behavioral structure was as depicted in Fig. 3, being the main ones the Slot and the CPG behavior models [8].

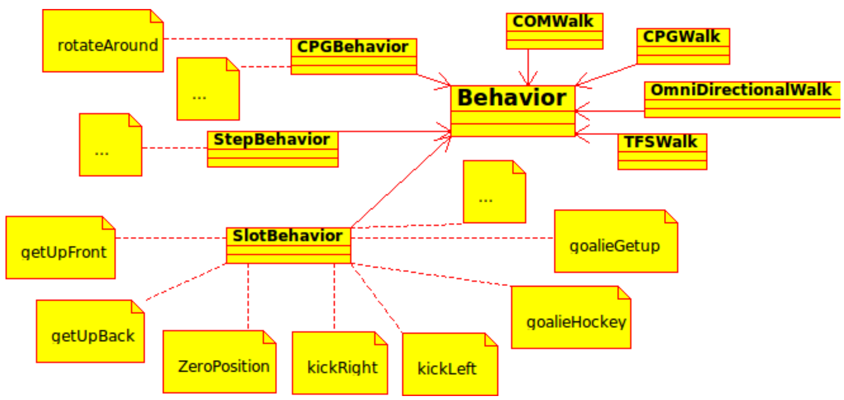


Fig. 3: Behavior structure diagram.

4.1 World modeling

One of the first steps for the integration is the transformation of image pixel points to metric coordinates centered on the robot. This is obtained through a geometric and trigonometric analysis of the camera relative to the ground [9]. The main points of interest are the ball, the goals and the white lines, for localization.

Although the localization calculations are performed by a separate process, the agent is still responsible for it. The need for the separation is related with the need to keep the agent cycle time perfectly periodic. The localization algorithm is based on guided update steps modeling the localization problem as an error minimization task and using an efficient numerical minimizer [10], adapted from the Tribots Middle Size League team. All the detected white points are basically matched with the known field layout through the referred error.

For achieving a good localization, the agent can decide to gather line points information from several points of view, in order to get a good grasp of all around it. This is obtained through an active sensing methodology, which moves the robot with the main objective gathering that information.

4.2 Decision model

As in the MSL CAMBADA team, the decision process is organized in several different stages. The agent decision is organized in roles and behaviors. It can be a simple static role assignment that only depends on a very small set of conditions (as the penalized or the goalie roles, for instance, which only depend on game Controller state and/or robot number) to a more complex assignment which depends on the conditions of the robot and the world around, including strategic issues.

The role is then responsible for a more detailed analysis of the conditions and take actions to achieve its goals. These actions are the behaviors available to the robot.

The assignment of the roles is made dynamically, so each robot can assume any given role during a game, dependent on the game conditions.

Each role and behavior is an instance that can be used at any time. Currently, the main roles are:

– **Goalie**: this role implements a state machine for the behavior of the goalie. It tries to keep in the goal area, tracking the ball trajectory and go down to defend when it needs.
– **Striker**: this is the main role during free play, is the robot that more actively goes for the ball. It is always trying to get it and score.
– **Midfielder**: this role is the third role that exists during free play. Apart from the Goalie and the Striker, all the other robots assume this role. Each of them keeps a strategic positioning, estimated by this role, to keep the team coherent and cooperative. Any Midfielder that reaches a situation more advantageous than the Striker will be reassigned as Striker, and the former one will fall back to the strategic positioning as a Midfielder.
– **Penalized**: this is the role assigned to penalized robots, to keep them crouched and inactive in compliance with the rules.

A set of other roles are implemented, both for development and fall-back situations, like manual game controller mode.

4.3 Strategy

The formation model used is called DT, originally developed in the Simulation League, where the positioning of each player results from defining a set of key ball positions for which the player target position is defined and then interpolating these positionings using a Delaunay Triangulation of these key ball positions.

The assignment of the strategic positioning is completely dynamic, with a priority based approach. Each robot estimates the most effective position for itself, making use of the worldstate information shared among the team mates. The key ball positions and base triangles used for the DT are previously configured in a graphic tool.

Our strategy model is prepared to have several formation files that can be used at any given time, according to a set of game conditions, defined by several types of parameters (game time, game result, opponent team, number of active robots, ...).

This model is currently being used on several teams of the group, particularly Simulation 2D and 3D and CAMBADA.

5 Development tools

For testing and debugging during development, several tools were created.

– **basestation**: This is a graphical tool that allows us to visualize the main data shared between robots, particularly their position and velocity, their perception of the ball in terms also of position and velocity, their perception on goals position, their role and behavior in the game, their state (as active or inactive) and in full debug, the line points they see (Fig. 4 top left).
– **cameraCalib**: This tool was created to test the projection model. It allows us to test and configure the several parameters involved in the vision systems of the robot in order to get a correct ground projection of each pixel. The verification is made visually with a user defined grid overlapping the real image (Fig. 4 bottom left).
– **directJointControl**: With this tool we can directly control the angle of each joint or, more importantly, to move the robot in puppeteer mode, for defining poses used in sequential behaviors (Fig. 4 bottom right).
– **cursor**: This is a remote control tool for dynamically test the walking parameters of an omnidirectional walk. It is used to test different set of parameters when on a new ground surface without the need to reconfigure for each test (Fig. 4 top right).

6 Conclusions

This paper describes the current development stage of the Portuguese Team for the purpose of the qualification for RoboCup 2013. Most of the work was done in the development of an efficient vision system and on the adaptation of some modules from the software used on the CAMBADA robots. A good deal of effort was also applied in the development of several tools to support development, both debug and configuration tools. Moreover, the agent from FCPortugal 3D was adapted, mainly the low level behaviors and walking.

The team wants to thanks the support of the institutions involved on the project, IEETA, University of Aveiro, as well as FEUP, INESC and Universities of Minho and Porto.

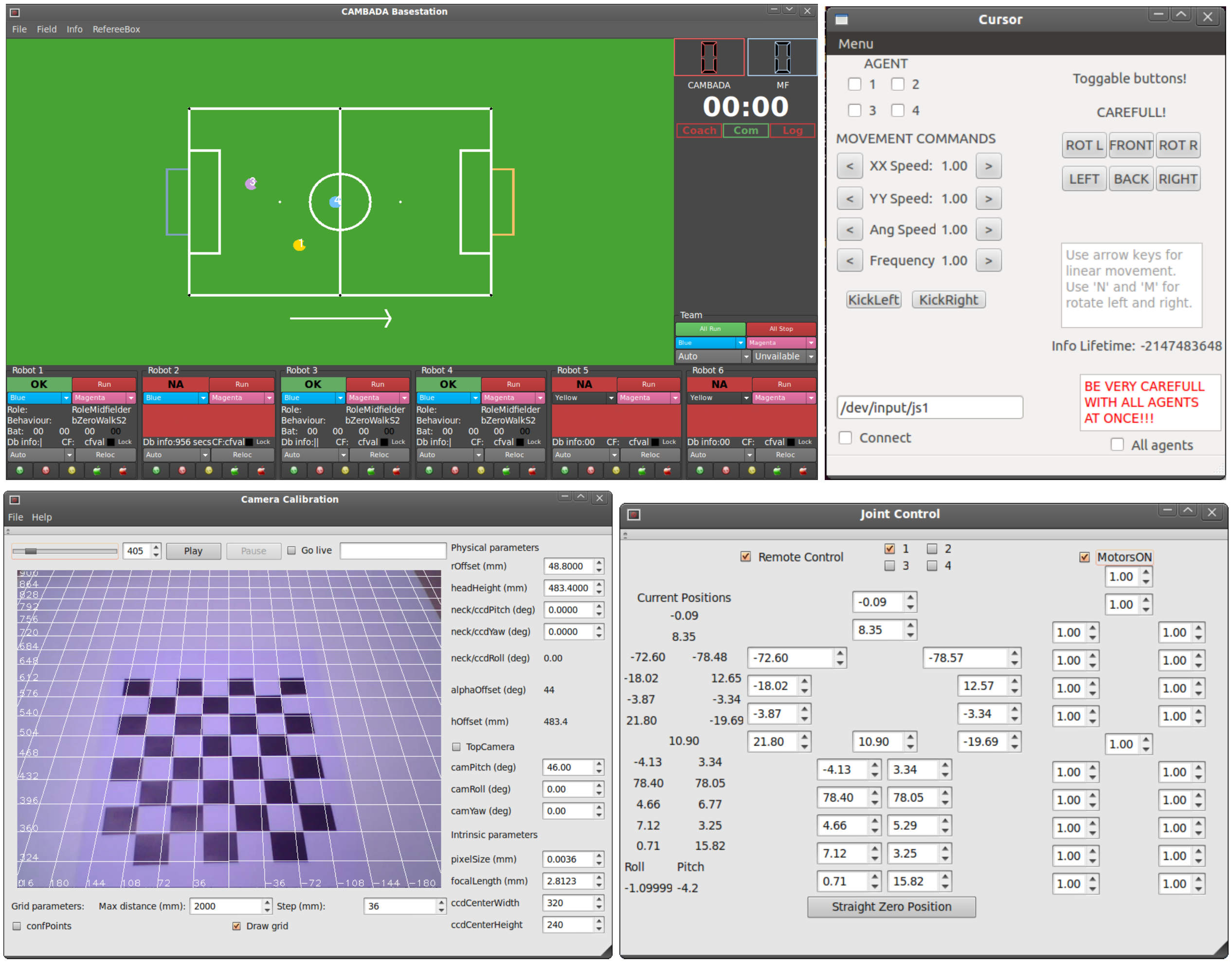


Fig. 4: Screenshots of the developed graphical tools.

Main project related publications

1. Neves, A.J.R., et al.: CAMBADA soccer team: from robot architecture to multiagent coordination. In Papić, V., ed.: Robot Soccer. InTech (2010) 19–45
2. Trifan, A.L., Neves, A.J., Cunha, B., Lau, N.: A modular real-time vision system for humanoid robots. In: Proc. of IS&T/SPIE Electronic Imaging 2012, San Francisco, USA (2012)
3. Silva, J., Lau, N., Neves, A.J.R., Rodrigues, J., Azevedo, J.L.: World modeling on an MSL robotic soccer team. Mechatronics **21** (2011) 411–422
4. Domingues, E., Lau, N., Pimentel, B., Shafii, N., Reis, L.P., Neves, A.J.R.: Humanoid behaviors: From simulation to a real robot. In Antunes, L., Pinto, H.S., eds.: Proceedings of the 15th Portuguese Conference on Artificial Intelligence. Volume 7026 of Lecture Notes in Artificial Intelligence (LNAI). Springer (2011) 352–364
5. Cruz, L., Reis, L.P., Rei, L.: Generic optimization of humanoid robots' behaviours. In Antunes, L., Pinto, H.S., Prada, R., Trigo, P., eds.: Proceedings of the 15th Portuguese Conference on Artificial Intelligence. Volume 7026 of Lecture Notes in Artificial Intelligence (LNAI). Springer, Lisbon (2011) 385–397
6. Rei, L., Reis, L.P., Lau, N.: Optimizing a humanoid robot skill. In Lima, P., Cardeira, C., eds.: Proceedings of the 11th International Conference on Mobile Robots and Competitions,. (2011) 78–83
7. Shafii, N., Passos, L.S., Reis, L.P.: Humanoid soccer robot motion planning using graphplan. In Lima, P., Cardeira, C., eds.: Proceedings of the 11th International Conference on Mobile Robots and Competitions,. (2011) 84–89
8. Picado, H., Gestal, M., Lau, N., Reis, L., Tomé, A.: Automatic generation of biped walk behavior using genetic algorithms. In Cabestany, J., Sandoval, F., Prieto, A., Corchado, J., eds.: Bio-Inspired Systems: Computational and Ambient Intelligence. Volume 5517 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2009) 805–812
9. Silva, J., Neves, A.J.R., Lau, N.: Estimating world coordinates in perspective vision systems for humanoid robots. In: Proc. of the 18th Portuguese conf. on Pattern Recognition RecPad 2012, Coimbra, Portugal (2012)
10. Lauer, M., Lange, S., Riedmiller, M.: Calculating the perfect match: an efficient and accurate approach for robot self-localization. In: RoboCup 2005: Robot Soccer World Cup IX. Volume 4020 of Lecture Notes in Computer Science., Springer (2006) 142–153