# CAMBADA: Team Description Paper

L. Almeida, J.L. Azevedo, P. Bartolomeu, E. Brito, M.B. Cunha,
J.P. Figueiredo, P. Fonseca, C. Lima, R. Marau, N. Lau, P. Pedreiras, A. Pereira,
A. Pinho, F. Santos, L. Seabra Lopes, J. Vieira

Transverse Activity on Intelligent Robotics
IEETA/DET – Universidade de Aveiro
3810-193 Aveiro, Portugal

**Abstract.** The CAMBADA middle-size robotic soccer team is described in this paper. This team was designed and is being developed by the authors, from scratch, during the last six months. The players, completely built in-house, incorporate several innovations at the hardware level, particularly the sensing and computational subsystems. At the software level, cooperative sensing uses a real-time database implemented over a real-time Linux kernel. Previous experience of the team in the simulation league has been highly relevant. Coordination approaches previously tested in that league are being integrated in the CAMBADA team.

## 1 Introduction

CAMBADA[1] is a new RoboCup middle-size league soccer team being developed by the Transverse Activity on Intelligent Robotics (ATRI), based at IEETA/DET, University of Aveiro. This project, started officially in October 2003, is funded by the Portuguese research foundation (FCT) [2]. Currently, it involves 10 university staff members, 3 research contracts, 1 PhD student, 1 service contract and 1 voluntary students.

CAMBADA is expected to compete for the first time in ROBOTICA'2004 (the Portuguese robotics festival) in the coming April. This paper is part of the qualification materials for RoboCup'2004.

ATRI is a multi-disciplinary group focused on robotics. Besides research in related topics, the activity within ATRI has also covered the organization of and participation in several robot competitions at national and international levels, some of which with remarkable outcomes, e.g. the Micro-Rato contest (local), the ROBOTICA 200x - Portuguese Robotics Festival (national), the Festival International des Sciences et

---

[1] CAMBADA is acronym of *Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture*; 'cambada' is also a Portuguese word for 'band' or 'mob'.

[2] Project POSI/ROBO/43908/2002, partially fundaded by FEDER.

Technologies, the RoboCup Soccer Simulation League, the AAAI contest. The authors therefore have high expectations with respect to the quality of their most recent project, the CAMBADA robotic soccer team, which is currently being finalized and is described in this paper.

Since the project will design and build the robotic players (instead of using commercial bases), the project is expected to contribute to the state of the art in a variety of fields, including actuators, distributed computational architectures, cooperative sensing, localization, basic robotic skills and team coordination. In order to develop successful robotic systems, many different technologies must be smoothly and coherently integrated. Integration is a crucial issue in robotics [10,19].

This paper is organized as follows: section 2 describes the physical design of the team, including mechanical structure, sensors, actuators, computational modules and communications; section 3 describes the software system of the team, including cooperative sensing, basic behaviors and coordination; finally, section 4 concludes de paper.

## 2 Team Physical Design

### 2.1 Mechanical Structure, Sensors and Actuators

Some of the main teams in the middle-size league (including the three times champion CS-Freiburg [6,21]) build their players over commercial robot platforms. There were some early notable exceptions, such as Golem and Sharif-Arvand, in which the players were almost completely built by the team developers. CAMBADA, as several other recent teams, will follow the second alternative.

The CAMBADA players were designed and are being completely built in-house. The baseline for robot construction is a cylindrical envelope, with 485 mm in diameter, which allows for a team of 5 robots, according to the recently proposed rules. The mechanical structure of the players is organized into four layers (Fig. 1). The components in the lower layer, namely the motors and wheels, are attached to an aluminium plate (below) placed 8 cm above the floor. The second layer, above the aluminium plate, includes the kicker and part of the computational infrastructure. The third layer contains a laptop computer, at 22.5 cm from the floor. The top layer consists of sensors, particularly vision sensors (fig. 1).

The players are capable of holonomic motion, based on three omni-directional roller wheels [3]. These are Transwheel 4202K “Cat-Track” wheels from Kornylak. Each wheel is actuated by a 24 V / 150 W Maxon motor, with a 7580 rpm no-load speed, controlled by a dedicated microcontroller. A Planetary gearhead for 15:1 reduction is coupled to each motor. Optical encoders of 500 p/r are attached to motor shafts.

The actuation mechanism for the kicker will be electromechanical or pneumatic. The currently operational kicker is pneumatic and allows for momentum control.

An electromechanical kicker is being developed, based on a 12W solenoid and on accumulation of energy in a capacitor. This kicker will also allow for momentum control. Both solutions will be tested and compared, and the best one will be chosen.

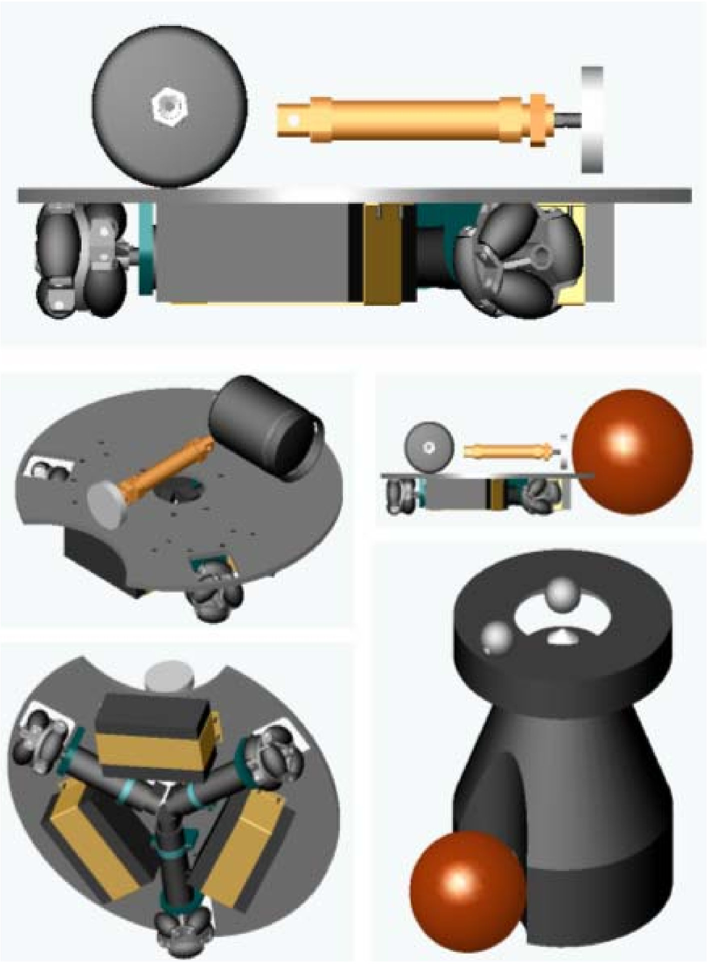

**Fig. 1.** Several CAD views of the CAMBADA player design

The sensing infrastructure is organized in three levels, which is consistent with the distributed approach of the robots and the team. The first level includes an intra-sensing support system, providing feedback for the control modules that will govern actuator actions. High resolution incremental encoders provide feedback for the motor control modules while contact, acceleration and/or force sensors will be exploited to support kicker force control.

The second level aims the sensing of the immediate robot surroundings. Instead of using infrared or ultrasound transducers for obstacle detection and avoidance, the robot navigation system relies on a single isotropic vision system that provides a 360 degrees field of view around the robot, approximately with a 1m radius (see Fig. 1, bottom right). This system uses a double mirror setup and a webcam. The mirror system is comprised of a central cone shaped mirror together with peripheral toroidal section mirror. The full setup can be accommodated in a cylindrical section with less than 10 cm height. This sub vision system is also used for proximity ball detection and handling. Omnidirectional vision has been used recently with some success in the RoboCup robots [2,14], but for obtaining global field views instead of for obstacle detection/avoidance.

Navigation on the game set and both self and team localization in space is the aim of the third level. In the current player prototype (Fig. 2), a low-cost webcam-type front camera, adapted with a wide-angle lens (approximately 106 degrees), is being used. This camera delivers 320x240 YUV images at a rate of 30 frames per second, and is connected via USB to the main computational unit of the robot (a laptop running the Linux/RTAI operating system).

Fig. 2 shows the player prototype already developed and used for the demonstrations presented in the qualification videos.

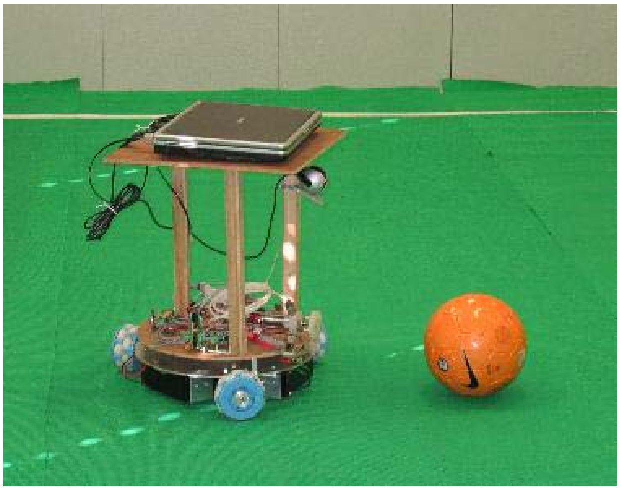

**Fig. 2.** Current CAMBADA player prototype

### 2.2 Computing, Communications and Monitoring Infrastructure

The robots computing system architecture follows the fine-grain distributed model [11] where most of the elementary functions, e.g. basic reactive behaviors and closed-loop control of complex actuators, are encapsulated in small microcontroller-based nodes, interconnected by means of a network. This architecture, which is typical in the automotive industry, favors important properties such as scalability, to allow the future addition of nodes with new functionalities, composability, to allow building a complex system by putting together well defined subsystems, and dependability, by using nodes to ease the definition of error-containment regions. Furthermore, a PC-based node, a laptop in this case, is used to execute higher-level control functions and to facilitate the interconnection of off-the-shelf devices, e.g. cameras, through standard interfaces, e.g. USB or Firewire (Fig. 3).

This architecture relies strongly on the network, which must support real-time communication. For this purpose, Controller Area Network (CAN) has been chosen, which is a real-time fieldbus typical in distributed embedded systems. This network is complemented with a higher-level transmission control protocol to enhance its real-time performance, composability and fault-tolerance, namely the FTT-CAN protocol (Flexible Time-Triggered communication over CAN) [1]. The use of FTT-CAN has the advantage of combining time-triggered communication with operational flexibility, supporting on-line reconfiguration and thus higher maintainability and capacity to cope with evolving requirements. At the present moment, the interconnection between CAN and the laptop is carried out by means of a CAN/RS232 gateway but a CAN/USB version is under development.

In what concerns the system software, a library of functions has been developed for the microcontroller-based nodes, to support communication and synchronization with the network, time triggering of actions, remote monitoring of internal state, and remote loading of programs. The PC-based node uses the Linux operating system complemented with RTAI – Real-Time Applications Interface, a real-time patch to enhance its capability to execute timed actions.

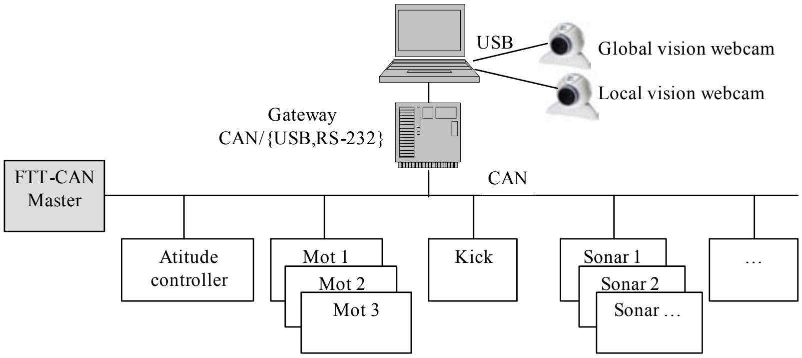

**Fig. 3.** The hardware architecture of each robotic player

The communication among robots uses the standard wireless LAN protocol IEEE 802.11x profiting from large availability of complying equipment. However, there are several issues concerning real-time communication over these protocols [5] most of which are related with the dynamic, open and high attenuation nature of RF-links together with the difficulty in detecting collisions. This project also includes research in that topic, namely the problems of clock synchronization and global time-stamping, network latency information and efficient traffic scheduling. .

This global architecture will support the required inter-robot interactions as well as global system monitoring. This latter aspect is carried out by an external station that has access to the robots internal state. During setup and robots tuning, particularly the sensorial subsystems, this station also has access to a global view of the playfield using an external camera.

## 3 Software System

The software system in each player is distributed among the various computational units. High level functions run on the laptop computer. Low level functions run partly on the laptop, partly on the microcontrollers.

### 3.1 Cooperative Sensing and Localization

The cooperative sensing and localization system manages the gathering of sensor data internally to each robot as well as from the robot team. The purpose of this system is to make the sensor data available to upper level behavioral tasks in an integrated fashion.

The cooperative sensing strategy will be developed based on the concept of Real-Time Data Base (RTDB) [11]. Although somewhat similar approaches have already been used in RoboCup (e.g. blackboard based approaches), in our case, real-time techniques will be used to enforce a correct timeliness in the refreshing of the items and to deliver information on the temporal accuracy of the items in the database. These techniques contribute to better control the robots and are not common in RoboCup middle-size league teams.

The RTDB contains a set of items that are images of local entities, with respect to each robot, as well as a set of items that are images of remote sensor data from other robots in the team. The RTDB management system automatically manages the refreshing of the sensor data in a transparent way, triggering the sensor systems and the necessary communications at an adequate rate in order to maintain a desired accuracy of its internal images. The RTDB is physically implemented on a block of shared memory that is accessible from tasks executing on the Linux side (non-real-time tasks) and on the RTAI side (real-time tasks). The timeliness aspects, including the related communication activity, are managed by the real-time subsystem (Fig. 4).

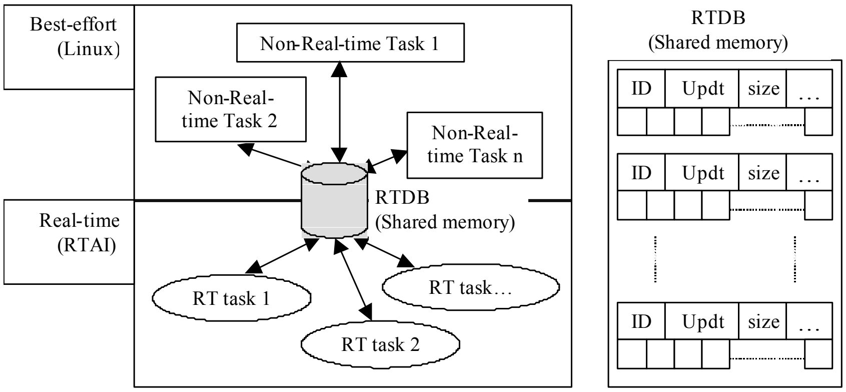

**Fig. 4.** A real-time database (RTDB) accessible locally to all processes

With respect to player sensing functions, simple color-based analysis, directly applied in the YUV domain, together with region-growing techniques for object formation, are used to detect the ball, goals, lines and obstacles. Distances are estimated based on the location of the objects in the image.

Eventually, data gathered by an USLPS (Ultra-Sonic Local Positioning System), a robot based system that will be developed, will also be integrated. DSP techniques will be used to solve problems related with signal generation, detection and reverberation in closed spaces.

Localization depends on two main landmarks, the goals. Every robot identifies its position with respect to both goals and communicates it to the other team mates. By integrating this information, the accuracy of localization information is improved [9]. The localization system relies strongly on vision, using feature extraction as reported in section 2.1. It is our intention to explore several configurations of the visual sensing system. Currently, we are using a double camera system, one for far vision and the other one for near vision (less than 1 meter radius). As already mentioned, the near vision system uses an omnidirectional system based on a special mirror. Another possibility that can be exploited later, is to use a binocular system for front (far) vision. The front binocular system, besides providing a wider viewing angle with respect to a single camera, will allow stereoscopic vision and, therefore, additional range information.

Cooperative sensing is essential for accurate localization (not only self-localization but also the localization of other objects in the field such as opponents, teammates, ball and goals). In which concerns cooperative sensing and information integration, one of the works that has been influential for our project is that of CS-Freiburg [6,21]. However, this team had a competitive advantage due to the use of laser range finders, and so it is hard to establish which part of their success was due to the software architecture.

### 3.2 Basic Behaviors

Basic behaviors are the action primitives the higher-level decision modules use to control the robot. It is essential to provide those modules with a good set of alternatives, each of which should be as efficient as possible.

Part of the computation involved in motion behaviors runs on the microcontrollers. The behavior control loop runs on the laptop. Given the holonomic capability of the players, the motion variables are the speed vector and the rotational speed. Based on the values of the variables, the motion micro-controller calculates the adequate speeds for the roller wheels. This information is then sent to dedicated motor micro-controllers. The current player prototype is able to move at a maximum speed of 1.4 m/s.

The basic behavious we are considering can be classified on the basis of their need, or not, for accurate self-localization and absolute localization of other objects in the field. Basic behaviours that can be implemented without localization include:

- SearchBall,
- GoToBall,
- InterceptBall,
- ApproachBall,
- Dribble,
- Shoot,
- Pass,
- FaceOpponentGoal,
- GotoAttack,
- GotoDefense,
- BlockOpponent, and
- UnStuck.

In the current state of development of the team, several behaviors have already been implemented: SearchBall, GoToBall, InterceptBall, ApproachBall, Shoot, FaceOpponentGoal. Basic behaviours that need self-localisation or absolute localisation of other objects in the field include:

- GoToPoint,
- DribbleToPoint,
- PassToPoint,
- FollowPath,
- ApproachBallFacingGoal,
- ApproachBallDefending.

The player acting as goalkeeper is enhanced with some specific behaviors like BlockGoal and PassiveInterception. These behaviors are hand coded [7]. This project also includes implementing and testinglearning techniques namely reinforcement learning [22] or neural networks [18].

Although most of the basic behaviors we use/propose are frequent among RoboCup teams [4,18,22], the omnidirectional movement capability of our robots (see section 2.1) offers new challenges for their implementation. The same basic behaviour, for example GotoBall, may be set to optimise speed, power consumption, or positional value of the chosen path, considering the position of the kicker during the movement. We intend to research the effects of several different path policies in the performance of a soccer player robot.

An interesting feature of our architecture is the control over the reactivity of basic behaviors (see [7,22]) by acting on the refresh rate of related sensors. For example, ApproachBall is very reactive, using a frame rate of 30 fps, while other behaviors rely on sensors that are updated less frequently. The same behavior, example Dribble, may be implemented with a hierarchical structure where lower layers are more reactive, and higher layers provide more accurate and higher-level information.

### 3.3 Play Strategies and Team Coordination

Play strategies and team coordination concern the decision-making processes carried out by each player in order to maximize, not only the performance of its actions, but also the global success of the team. An utility-based decision-making engine will be used for this purpose. Besides the state of the physical world, the decision-making process in each player will also take into account information on the beliefs and intentions of the teammates. This will likely be supported by a communication strategy based on a subset of KQML.

Given the characteristics of the domain (highly dynamic and uncertain environment, possibility of communication failures), centralized control at the team level will be limited to the infrequent interventions of a coach (see below). In the normal situation, team coordination will be achieved implicitly through common play strategies, that are adopted flexibly as game progresses. In this respect, we see approaches initially introduced and tested in the RoboCup simulation league being gradually adopted by other leagues. In particular, the soccer notions of “formation” and “role”, initially implemented in CMUnited [20] have been used since RoboCup2000 in some of the main F-2000 teams, including CS Freiburg [6,21]. A distinction between “strategic” and “active” situations has also appeared in various teams, both in the simulation league (e.g. CMUnited [20] and FC Portugal [13,16]) and the F-2000 league (e.g. CS Freiburg). Mechanisms for players to determine their target positions in strategic positioning situations will be either based on potential field methods [6,20,21] or SBSP [13,16]. Our previous involvement in the development of FC Portugal will be useful in devising an appropriate team coordination approach for our F-2000 team.

The possibility of achieving coordination in specific situations via the execution of pre-defined team-level plans (setplays) will also be considered.

Given that different alternatives at the hardware level are under evaluation, it is possible that, at some point, the team is heterogeneous. This constitutes an additional problem for team coordination. In such a case, a coordination strategy for exploring (taking advantage of) the differences will be developed. Heterogeneity will also lead to a revision of current methods for the exchange of roles between teammates.

Although control will be essentially distributed, an agent called “coach” will be included in the architecture. The coach is a program that will run in a computer located off-board the robots (can be the same that also runs the monitoring application). It will seldom interfere with the players’ actions, as it will have an essentially strategic function. The role of the coach is to analyze the game in the long run and provide strategic advice to the team. This analysis can be computationally expensive, so it makes sense to centralize it.

## 4 Conclusion & Ongoing Work

The CAMBADA robotic soccer team is being completely developed in-house. It is based on holonomic players. The players incorporate several innovations at the hardware level, particularly the sensing and computational subsystems. At the software level, cooperative sensing uses a real-time database implemented over a real-time enhanced Linux kernel.

Currently, a working player prototype is already available, which is able to:

- search, locate and follow the ball,
- search and locate de goal,
- avoid abostacles,
- kick,
- shoot to goal.

This is an experimental prototype that enabled work in the control and vision modules of the software system. This prototype does not completely follow the final design of the players. The players are currently being built in their final form. In parallel, work on cooperation is ongoing, particularly on the real-time database.

Previous experience of the team in the simulation league (FC Portugal team) will be relevant for the coordination of the team. Coordination approaches previously tested in that league are being integrated in the CAMBADA team.

The completely assembled players in their final architecture are expected to be available by the end of March (at this moment, all the components for the mechanical and electrical infrastructure are already available). During April, all efforts will be devoted to cooperation and tuning of basic behaviors. By the end of April the team is expected to compete for the first time in a national robotics soccer tournament in the framework of ROBOTICA 2004. Finally, the months of May and June will allow to tune all the details of the team so that everything is ready for RoboCup 2004.

## References

1. Almeida, L., P. Pedreiras and J.A. Fonseca (2002) "FTT-CAN: Why and How", *IEEE Tran. Industrial Electronics*.
2. Bonarini, A., et al. (2000) "An Omnidirectional Vision Sensor for Fast Tracking for Mobile Robots", *IEEE Trans. Instrum. Measur.*, 49 (3), p. 509-512.
3. Carter, B., et al. (2002) "Mechanical Design and Modeling of an Omni-directional RoboCup Player", RoboCup-2001, A. Birk, et al (edrs), Springer Verlag.
4. Castelpietra, et al. (2000) "Communication and coordination among heterogeneous mid-size players: ART99", *RoboCup-2000*, P. Stone, et al. (edtrs), Springer Verlag.
5. Decotignie, J.-D., et al. (2001) "Architecture for the Interconnection of Wireless and Wireline Fieldbuses", *FeT'01 - IFAC Conf. on Fieldbus Technologies*, Nancy.
6. Dietl, M., J.-S. Gutmann and B. Nebel (2001) "Cooperative Sensing in Dynamic Environments", *Proc. IROS'2001*.

7. Iocchi, L. and D. Nardi (1999) "Design and Implementation of Robotic Soccer Behaviors: a User Viewpoint", *Proc. Third Int'l ICSC Symp. Intelligent Industrial Automation (IIA'99) and Soft Computing (SOCO'99)*, 766-772, ICSC Acad. Press.
8. Jamzad, M., et al. (2001) "Basic Requirements for a Teamwork in Middle Size RoboCup", *Sharif-ARVAND team description*.
9. Kato, K., et al. (1999) "Identifying and Localizing Robots in a Multi-Robot System Environment", *Proc. IROS'99*.
10. Kitano, K., M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa (1995) "RoboCup: The Robot World Cup Initiative", *Proc. of IJCAI-95 Workshop on Entertainment and AI/Alife,* Montreal.
11. Kopetz, H. (1997) *Real-Time Systems Design Principles for Distributed Embedded Applications*, Kluwer.
12. Kuc, R. (1997) "Biomimetic Sonar Recognizes Objects […]", *Journ. Acoustical Soc. America*, 102 (2), p.689-696.
13. Lau, N. and L.P. Reis (2002) "FC Portugal 2001 Team Description: Flexible Teamwork and Configurable Strategy", *RoboCup-2001*, A. Birk, et al (edtrs), Springer Verlag.
14. Marques, C. and P.U.Lima (2001) "A Localization Method for a Soccer Robot Using a Vision-Based Omni-Directional Sensor", *RoboCup-2000*, P. Stone, et al. (edtrs), Springer Verlag.
15. Pinho, A. (2002) "An online preprocessing technique for improving the lossless compression of images with sparse histograms", *IEEE Signal Processing Letters*, vol. 9, no. 1, p. 5-7.
16. Reis, L.P. and N. Lau (2001) "FC Portugal Team Description: RoboCup 2000 Simulation League Champion", *RoboCup-2000*, P. Stone, et al. (edtrs), p. 29-40, Springer Verlag.
17. Sabatini, A.M. and A. Rocchi (1998) Sampled Baseband Correlators for In-Air Ultrasonic Rangefinders", *IEEE Transactions on Industrial Electronics*, 45 (2), p.341-350.
18. Schmitt, T., S. Buck and M. Beetz (2002) "AGILO RoboCuppers 2001: Utility- and Plan-based Action Selection based on Probabilistically Estimated Game Situations", *RoboCup 2001*, Springer.
19. Seabra Lopes, L. and J.H. Connell (2001) "Semisentient Robots: Routes to Integrated Intelligence", *IEEE Intelligent Systems*, vol. 16, n. 5, Computer Society, p. 10-14.
20. Stone, P. and M. Veloso (1999) "Task Decomposition, Dynamic Role Assignment and Low Bandwidth Communication for Real Time Strategic Teamwork", *Artificial Intelligence*, 110 (2), p. 241-273.
21. Weigel, T. et al. (2001) "CS Freiburg: Sophisticated Skills and Effective Cooperation", *Proc. European Control Conference (ECC-01)*, Porto.
22. Wiering, M., et al. (2002) "Clockwork Orange", *RoboCup 2001*, Springer.