# ISocRob 2006

## Team Description Paper

Pedro Lima, Hugo Costelha, João Estilita, Nelson Martins, Gonçalo Neto, João Santos
Institute for Systems and Robotics, Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisboa, PORTUGAL

## 1 Introduction and Overview

The ISocRob team and its regular participation in the Middle-Size League (MSL) of RoboCup since 1998 are the competition side of the SocRob project[1], a research endeavour of the Intelligent Systems Laboratory of the Institute for Systems and Robotics at *Instituto Superior Técnico* (ISR/IST), Technical University of Lisbon, which started in 1997. The project goal is to develop a novel approach to the design of a population of cooperative robots based on concepts borrowed from Systems Theory [3] and Distributed Artificial Intelligence [6]. Besides the MSL team, ISocRob has participated in RoboCup Soccer Simulation League in 2003 and 2004, and started a new Four-Legged League team, which has pre-registered for RoboCup 2006.

**Fig. 1** – IsocRob new omnidirectional robots.

This paper describes the current status of the SocRob project research progresses and endeavours, as well as the status of the ISocRob team. We focus on the past addressed research, but also on current related research not yet (fully) applied to the actual robots.

The remaining of the paper is divided in two sections: Section 2 describes the details of the new robots hardware and software architecture. Section 3 addresses the on-going and some past research developed under the project, as well as new research challenges.

---

[1] The acronym of the project stands both for **Society of Robots** and Soccer Robots, the case study where we are testing our population of robots.

# 2 Hardware and Software

## 2.1 Hardware

ISocRob's team has been composed, since 2000, of four Nomadic Super Scout II robots. In the last two years, ISR/IST developed 5 new omnidirectional robots, in a joint venture with the Portuguese SMEs IdMind (responsible for the electronic hardware) and ServiLog (responsible for the mechanical hardware). At the time of writing, the 5 robots are ready to play, but some hardware details are not yet working at 100% (the robots can not run at full speed and the kicker does not always work properly), and the transfer of code from the old Scouts has not been fully made yet. Four of the robots are shown in Fig. 1.

The new robots have most of their processing power concentrated on a NEC FS900 laptop, with an INTEL Centrino 1.6 GHz processor, with 512MB RAM and a 30GB disk. The laptop includes a CD-ROM, wireless 802.11b, 3 USB 2.0 ports, and 1 mini-firewire port, as well as a spare Li-Ion battery for extra autonomy.

Three MAXON DC motors model RE35/118776, nominal voltage 15 V (operated at 12 V) and 91:6 (or 15.1(6):1) gear ratio (ref. MAXON 203116) support each robot tripod kinematic structure. The maximum velocity achievable with such motors is estimated to be 4m/s, leading to a maximum translational speed of the robot around 3.5m/s and a maximum rotational speed of 20rad/s (1146 degree/s). The motors are controlled by Faulhaber MCDC2805 controllers, which include a programmable PID controller, power amplifier and RS232 connection. The omnidirectional wheels were specially designed to reduce the robot vibration, by a suitable design of the small rolling drums placed along the border of the main wheel.

A new electromechanical kicker was designed by ServiLog, based on a piston pulled by a DC motor through a rack and pinion system, which can be detangled by acting on a pole by a servo motor. The kicker has a range of possible displacements, allowing kick force control (i.e., ball speed). The command electronics is designed to pull the piston to its maximum displacement value in less than 1s, using closed loop control (an infrared sensor measures the displacement). Another new device is the front drum that acts both passively when a ball is received by the robot, by damping its motion, and actively, by rolling the drum at constant speed over the ball to keep it “locked” with the robot.

Each robot is endowed with the following sensors:

- 1 AVT Marlin F-033C firewire camera. The camera is part of an omnidirectional catadioptric vision sensor, similar to the one used in the old robots [7]. The mirror was re-designed to obtain a bird's eye-view ranging from 250 to 7000mm away from the robot center. The mirror radius is 50mm, and the lens image plan is located 600mm above the ground. The bottom point of the mirror is located approximately 120mm above the image plan.
- 16 sonars (SRF04 RangeFinder) disposed in a ring around the robot.
- 1 500 CPR encoder per motor for motor control and odometry.

- 1 AnalogDevices rate-gyro XRS300EB to improve orientation determination.
- 1 Creative Notebook Optical Mouse (800 dpi resolution and maximum speed of 1m/s) to improve position determination.
- 2 Sharp infrared sensors, to measure the kicker piston displacement and to detect the ball when it is between the robot fingers.

To power the electronics and motors, 2 packs of 9Ah NiMH batteries per robot are now used, with a significant gain in autonomy/weight ratio. IdMind has developed special chargers for these batteries, which allow charging the robots in maximum 3 hours, with the batteries in place, as well as running the batteries from DC current with a cable.

``Plug-and-play" connections of most peripherals to the laptop where decision-making, guidance and navigation algorithms are running were used. The lower level electronics is mostly based in PIC microcontrollers, which manage the interface between the laptop and the several available devices (sensors and actuators).

### 2.2 Software

The new software architecture of the SocRob project, depicted in Fig. 2, keeps the fundamental principles of the functional architecture the group has been following since 1999 [8]. Nevertheless, it was completely re-designed to be re-written on C++, as well as to improve the matching between software modules and the main functional concepts.

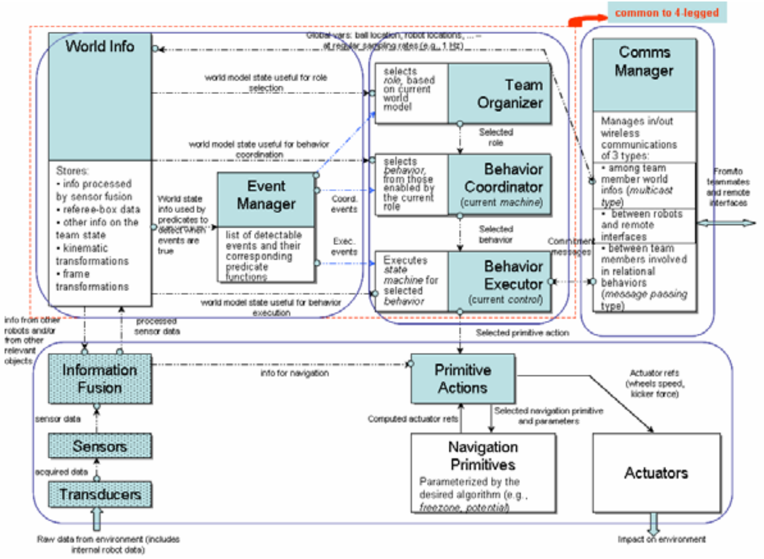

**Fig. 2** – SocRob project new software architecture.

The *World Info* is an object which stores the relevant information about the world, such as robot postures, ball position or current score. This information can be obtained either by sensor information or messages received from teammates or the referee box.

The *Behavior Executor* decides which primitive action to execute at each step, given a selected behavior. World Info data is used to take the decisions. Events are determined from World Info data as well, and are used to trigger internal state changes in the finite state automaton implementing a behavior. The *Behavior Coordinator* selects which behavior to run next, while the *Team Organizer* selects which *role* the player will perform. The selected *role* will only affect the set of behaviors a player can run, e.g., if a player has the role of defender, it will only be authorized to run defensive behaviors.

Behaviors can be classified into three distinct groups according to the relations among robots they imply [5]:

- Organizational: Those which are related to the team's organization, that is, the assignment of roles to players.
- Relational: Those involving two or more teammates, such as performing a pass or coordinately defending the goal.
- Individual: Those which are executed by one single robot.

## 3 Addressed Research and Future Challenges

From the very beginning of the project, one main concern has been the development of behavior coordination and modelling methods which support our integrated view to the design of a multi-robot population, not necessarily for playing robot soccer. The three types of behaviours considered by the behaviour architecture were described in the previous section. Behaviors are externally displayed and emerge from the application of certain operators. From the operators standpoint, the architecture has three levels, clearly mapped in the software architecture of Fig. 2:

- **Team Organisation**: where, based on the current world model, a *strategy* is established, including a goal for the team. This level considers issues such as modelling the opponents behavior to plan a new strategy. Strategies simply consist of enabling a given subset of the behaviors at each robot.
- **Behavior Coordination**: where switching among behaviors, both individual and relational, occurs so as to coordinate behavior/task execution at each robot towards achieving the team goal, effectively establishing the team *tactics*. Either a finite state automaton or a rule-based system can currently implement this level, but other alternatives are possible, such as Petri nets [2].
- **Behavior Execution**: where primitive actions run and where they interface the sensors, through the world info block, and the actuators. Primitive actions are linked to each other so as to implement a behavior. Currently, each behavior is implemented as a finite state automaton whose states are the primitive actions and transitions are associated to events. Events are defined as defined in this context as occurring when a change of (logical conditions over) predicate values (from TRUE  to FALSE or FALSE to TRUE) takes

place, e.g., event `lost_ball` occurs when predicate `has(ball)` value changes from TRUE to FALSE.

A Discrete Event Systems [3] based approach has been followed to the modelling of behaviours and their coordination. Individual behaviors are modelled by finite state automata (FSA), and some work has been done on optimal task planning by composing primitive actions into FSA, given the uncertainty associated to the action effects and the goal of minimizing the time to a goal [10]. Due to their capability to model concurrency, Petri nets are specially suited for modelling relational behaviours, as they can capture the concurrent nature of exchanging messages between teammates while each of them is executing their own primitive actions. Petri net models have been developed based on concepts borrowed from Joint Commitment Theory [1][4].

Sensor fusion methods for world modelling were also introduced by our team in [9]. The goal is to maintain and update over time information on the relevant objects, such as ball position and velocity, teammates pose and velocity, opponents pose and velocity, or position of the goals with respect to the robot. Such information is obtained by each robot from the observations of its internal sensors and then fused among all the team robots, using a Bayesian approach to sensor fusion.

Metric navigation has been one of our major research topics. Using an omnidirectional catadioptric vision system which preserves the Euclidean norm for ground-plane views, we developed a method to reset the robots odometry at regular time instants, from single shot images from the omnidirectional vision system. This has now been extended to include either Monte-Carlo or Kalman Filter based self-localization. We are currently extending the sensor fusion approach described above to add the new rate-gyro and optical mouse sensors to the set of observation sensors used in these methods, so as to improve self-localization even further.

An alternative distributed decision-making architecture supported on a logical approach to modelling dynamical systems, which is based on situation calculus [13], was also introduced by the team in [12]. This architecture includes two main modules: i) a *basic logic decision unit*, and ii) an *advanced logic decision unit*. Both run in parallel; the former intends to quickly suggest, using simple logical decision rules, the next behavior to be executed, whereas the latter uses more sophisticated reasoning tools (e.g., situation calculus) capable of planning, learning and decision-making, both for individual and cooperative (teamwork) situations. This configures an hybrid architecture where the basic (reactive) unit only controls the robot if the advanced (deliberative) unit takes too long to make a decision, assuming a situation urgency evaluation. A partial implementation of this architecture, the basic logic decision unit, was already performed using Prolog. In 2003, this approach was used to select behaviors at the Behavior Coordination level. Its modelling convenience allowed the quick development of different roles for field players (attacker *vs* defender), as well as *dynamic role change* between field players (defenders switch with attackers, depending on who is in a better position to get the ball).

One recurrent discussion among the team concerns the usage of self-localization. While an accurately self-localized robot seems definitely to be an advantage, most of our game actions (e.g., dribbling to the goal, kicking, passing) are currently relying on an accurate knowledge of the intervenient robots posture. A more robust solution would be to use vision to determine relative measurements, e.g., distance to ball or goals, and use them for those actions (e.g., dribbling to where the goal is seen, kicking to the goal when we see it open, or moving towards the goal and avoid the goalkeeper before kicking, passing to a teammate where it is seen by the passing robot). On the other hand, self-localization would be better suited for organizational purposes, such as distributing the robots per field zones, or deciding to pass to zones where we know that there is a well positioned teammate and less opponent robots.

Other research challenges we are currently investigating concern using reinforcement learning to adjust parameters of our discrete event task models, stochastic satisfiability methods to improve the efficiency of searching for optimal plans using discrete event models, modelling the behaviour of opponents using FSA, so as to use this in game-theory based decision optimization [11], and formation control methods to dynamically distribute the team robots across the field.

## Acknowledgments

This work was supported by the project `"SocRob'" (POSI/ROBO/43900/2002) of the Fundação para a Ciência e a Tecnologia and POSI, in the frame of the QCA, and by the Infaimon and BASE2 companies.

## References

[1] P. Cohen, H. Levesque. "Teamwork", *Nous*, Vol. 35, pg. 487-512. (1991)

[2] C. Girault, R. Valk, *Petri Nets for Systems Engineering*, Springer, (2003)

[3] S. Lafortune, C. Cassandras, Introduction to Discrete Event Systems, Kluwer Academic Publ., (1999)

[4] B. Van der Vecht, P. Lima, "Formulation and Implementation of Relational Behaviors for Multi-Robot Cooperative Systems", *RoboCup-2004: Robot Soccer World Cup VIII*, Springer-Verlag, Berlin (2005)

[5] A. Drogoul, A. Collinot, "Applying an Agent-Oriented Methodology to the Design of Artificial Organizations: A Case Study in Robotic Soccer", *Autonomous Agents and Multi-Agent Systems*, Vol 1, pp. 113-129, Kluwer Academic Publ., (1998)

[6] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, 1999

[7] P. Lima, A. Bonarini, C. Machado, F. Marchese, C. Marques, F. Ribeiro, D. Sorrenti: "Omni-Directional Catadioptric Vision for Soccer Robots", *Journal of Robotics and Autonomous Systems* 36 (2001)

[8] P. Lima, L. Custódio, R. Ventura, P. Aparício, "A Functional Architecture for a Team of Fully Autonomous Cooperative Robots", *RobCup-99: Robot Soccer World Cup III*, Springer-Verlag, Berlin (1999)

[9] P. Marcelino, P. Lima, "Bayesian Sensor Fusion for Cooperative Object Localization and World Modelling". *Proceedings of the 8th Conference on Intelligent Autonomous Systems*, Amsterdam, The Netherlands (2004)

[10] B. Damas, P. Lima, "Stochastic Discrete Event Model of a Multi-Robot Team Playing an Adversarial Game", *Proc. of 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles - IAV200*4, Lisboa, Portugal, (2004)

[11] G. Neto, P. Lima, "Minimax Value Iteration Applied to Robotic Soccer", *Proc. of IEEE ICRA 2005 Workshop on Cooperative Robotics*, Barcelona, April, (2005)

[12] V. Pires, M. Arroz, L. Custódio, "Logic Based Hybrid Decision System for a Multi-robot Team", *Proc. of 8th Conference on Intelligent Autonomous Systems*, Amsterdam, The Netherlands, (2004)

[13] R. Reiter, "Knowledge in Action", MIT Press (2001)