

# S.P.Q.R. Legged Team

## Report from RoboCup 2003

L. Iocchi and D. Nardi

Dipartimento di Informatica e Sistemistica  
Università di Roma “La Sapienza”  
Via Salaria 113 - 00198 Roma, Italy  
{iocchi,nardi}@dis.uniroma1.it,  
WWW home page: <http://www.dis.uniroma1.it/~spqr/>

**Abstract.** This report describes the development of techniques and the overall architecture organization of the software as well as the main achievements obtained in the competition RoboCup 2003. More specifically, we describe both the development efforts performed in 2003 (i.e. the realization of an efficient and robust color segmentation algorithm, the development of a localization technique based on Kalman Filter, and the realization of a coordination mechanism based on dynamic role assignment) and the previous work on locomotion, vision and action planning. Finally, we analyze the results of the game and comment on the results obtained during the tournament.

## 1 Introduction

S.P.Q.R. is the group of the Faculty of Engineering at University of Rome “La Sapienza” in Italy, that is involved in developing RoboCup teams within the Four-Legged League [?], the Middle Size League [?], and recently also within the Rescue Robots League.

In this paper we present the main achievements and research objectives of the **S.P.Q.R. Legged** team for RoboCup 2003 as well as a discussion on the results obtained during the competition. The work in the previous years was mainly focussed on three basic aspects: locomotion, perception, and deliberation [?], while the development for the 2003 competition has been focussed on: 1) color segmentation by means of an implementation of a dynamic thresholding method which is both efficient on the AIBO robots and robust to light conditions; 2) robot self-localization, by applying a probabilistic method based on feature extraction and Kalman Filtering [?]; 3) multi-robot coordination, based on dynamic role assignment [?].

In the following section we will briefly describe the main features of the system architecture and the modules developed; in Section 3 we show an analysis of our results during the competition RoboCup 2003; finally in Section 4 we draw some conclusions and describe future developments.

## 2 Software Architecture

The main focus of our research group is the realization of a team of cognitive soccer robots and to this end we have defined and implemented a hybrid architecture that allows for an effective integration of reasoning and reactive capabilities, a high level representation of the plans executed by the players, a formal account of the system for generating and verifying plans (see also [?,?] for a detailed description). This architecture has been designed to provide for a high level representation and deliberation on both the SPQR Legged Team and the SPQR Wheeled Team.

Specifically, there are four main functional modules: *Perception*, *Deliberation*, *Control*, *Coordination*. The communication among these modules is done by sharing a common memory that contains the internal model of the world evolving during the game.

This world model has two different representations: i) numerical data, representing object coordinates in the world (i.e. position of the robot in the field, position of the ball, etc.); ii) symbolic predicates, representing facts that holds during the game (e.g. the ball is close to the robot, the robot cannot see the ball, etc.). The first representation is used in the Control module to execute the actions (e.g. going to the ball); while the symbolic representation is used within the Deliberation module to take decisions about which primitive actions should be activated in the current situation of the world.

Thus, the Perception module analyzes the sensor data (which are mainly obtained by the camera) and stores and updates the internal world model. The Deliberation module is in charge of executing a plan and activating/deactivating the primitive actions according to the current situation of the world. The Control module is responsible for actual execution of primitive actions. Finally, the Coordination module is responsible for selecting the plan to be executed according to a distributed coordination protocol that has already successfully implemented in the Middle Size league [?] (see Section ??).

All the components of this architecture, that will be described below, have been developed from the students in our group.

### 2.1 Control

The Control module provides an interface to the physical functionalities of the robot by accepting abstract command from Deliberation and translating them into effective commands to the actuator. This is used both for locomotion and for moving the robot's head.

During past competitions, we experienced that a fixed set of walking styles caused a very complicated deliberation process, and we felt the need for a purely parametric locomotion generator. Our locomotion module accepts as inputs the  $x$  and  $y$  displacement, and the rotation  $\theta$  of the robot, to be executed in one step. Then it generates all the joint reference trajectories to obtain such displacements. So we do not need the off-line tool any more, and the other software modules can fully control the robot motion in the  $x, y, \theta$  space. One of the major observed

benefit is that the deliberation process and the low level motion control are much more easier and effective. We have added some more kicks to our previous set, to match specific game situations.

## 2.2 Perception: color segmentation by dynamic thresholding

One of the main problems with the AIBO robots is the implementation of the vision system, since the on-board camera has low quality and low resolution, the computational power of the robot is very limited with respect to standard PC, and the hardware color segmentation module only allows for static thresholding and thus it is not very robust to changes in lighting conditions. Moreover, color calibration is always a very demanding activity that is performed by the teams before the matches, in order to calibrate vision systems of the robots.

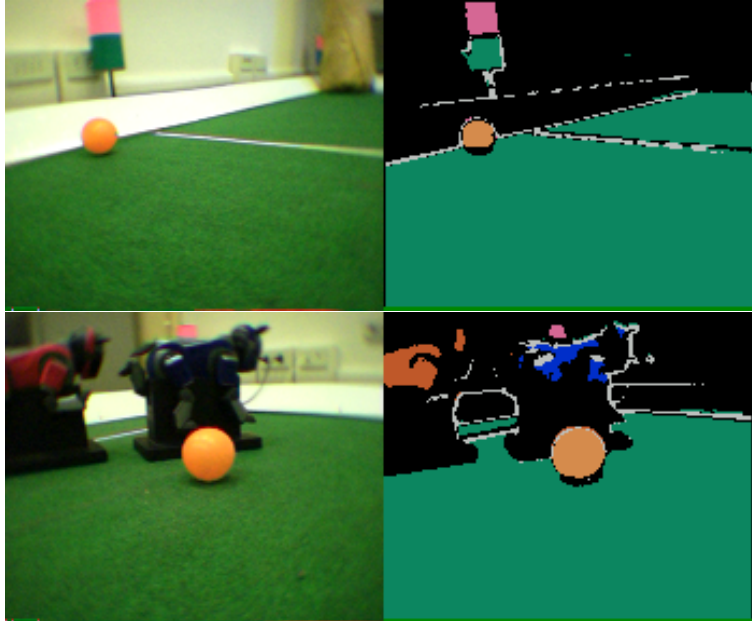
In order to improve color segmentation for the AIBO robots we have devised and implemented a method based on dynamic thresholding, that is both efficient and robust to different light conditions. This improvement has allowed for a better recognition of the objects in the field in terms of both precision and reliability and a localization technique (which is described in the next section) has been implemented by making use of features extracted by the color segmentation module.

The dynamic color segmentation technique we have implemented is based on the following steps: 1) image filtering for the U and V components based on a SUSAN operator; 2) creation of the color histograms for the U and V components of the image and detection of its modes; 3) color segmentation by using the modes of the histograms detected before. 4) edge detection for the Y components

The first step is important in order to reduce noise in the image and to have smoother histograms for the components U and V. We have designed a modification of the SUSAN filter [?] that can be efficiently implemented on the AIBO robots (we have called it *Real-Time SUSAN Filter*). With respect to the standard SUSAN filter, the Real-Time version has the two specific features: 1) the correlation function is rectangular instead of Gaussian; 2) it makes use of a 3x3 kernel in which only the 4-connected pixels are considered (i.e. only north, south, east and west positions). The Real-Time version of the SUSAN filter has a lower (but still significant in our application) noise reduction ability, and it has been implemented on the AIBO robots ERS-210 with 200 MHz CPU and for high resolution images (176x144) at a frame rate of 14 fps. The second step is the application of a dynamic thresholding method, in which the color histograms of the U and V components are created, the corresponding modes are detected and the color thresholds are computed. The third step is a color segmentation procedure that is based on the color thresholds computed in the previous step. Finally, in order to properly detect also the lines in the field, that are white and thus not well characterized by the U and V components, we have also applied an edge detection process for the Y component of the image. We have implemented a SUSAN Edge Operator, that has been shown to be more precise and effective than other methods [?]. An efficient implementation of such operator has been

obtained by defining a *Lite SUSAN Edge Operator*, that uses a pre-computed correlation function with a 3x3 kernel, that does not determine the direction of the edge and that does not perform the non-maxima suppression phase. This Lite version of the SUSAN Edge Operator can process more that 20 fps (for the AIBO robots ERS-210A with 400 MHz CPU and high resolution images 176x144).

The method proposed above has been implemented with several kinds of code optimizations and it currently allows for processing almost 10 fps (for the AIBO ERS-210A robots with 400 MHz CPU and high resolution images 176x144), during the normal operations of the robots (i.e. in parallel with other modules). The overall results of the technique in terms of quality of color segmentation is shown in Figure ??.



**Fig. 1.** Image color segmentation

Color segmentation is followed by three additional phases: blob extraction, object recognition and 3D reconstruction. Blob extraction has been implemented following the work in [?]. The segmented image is decomposed in runs, where every run is made up by horizontally contiguous pixels of the same color, and then vertically contiguous runs are merged by making use of a union-find algorithm, thus forming the blobs. Once the blobs are obtained, an object recognition routine classifies them on the basis of their color, while trying to recover from possibly wrong decisions through the use of some heuristics, depending on the Four-Legged League operational setting. During this process every blob is

marked with a label corresponding to the object it represents and additional information about the resulting objects is used to adjust classification according to known constraints. Finally, a method based on the robot kinematic model is adopted for calculating the position of the robot camera with respect to the ground and thus obtaining the position of the objects relative to the robot.

### 2.3 Perception: Vision-based Localization

Building on the capabilities of the vision system, a localization procedure was written, based on the estimation of the landmarks in the field and on the use of a probabilistic model defined by an Extended Kalman Filter (see [?] for a similar approach in the Middle Size League). This technique is based on the position of the landmarks with respect to the robot, and since the object classification reported in the previous section allows for discriminating among those landmarks (thus solving the data association problem), the localization technique has been implemented in such a way that a correct localization is achieved also in the case of “kidnapped” robots. We are currently performing extensive experiments for evaluating precision and accuracy of our localization method.

### 2.4 Deliberation and Action Planning

The Deliberation module is in charge of taking decisions on which action should be executed at every step. This is achieved by the execution of *plans*, that are transition graphs specifying the states in which the robot is during the game (nodes) and the primitive actions that should be performed in the current state (edges).

The Deliberation module thus includes the implementation of a set of primitive actions, that delegate the Control module to perform actual motion, and a plan execution monitor, to activate/deactivate these actions according to the conditions that are satisfied the symbolic representation of the world.

*Primitive Actions* . One of the main development efforts has been the reorganization of the primitive actions, in order to make them more parametric (with respect to speed, for instance). Since localization information has been made available through the localization module, the higher level plans were rewritten in order to take advantage of the knowledge about the robot location. In particular, the goalkeeper can reposition itself on the goal line without having to turn back and look at it (which has proved to be a very risky action). Moreover, localization allowed for a safer and more appropriate choice of the kick to be activated by the robot when it is near the ball, for instance by choosing a right head kick when it knows the opponent goal being on its right, without the need to look at it.

*Plan generation and execution* . The plans that are executed by the robot may be generated off-line by an automatic planner [?], or written by the user with a graphical tool. The Deliberation Module implements an algorithm that visits

the graph activating and deactivating actions according to the current state of the robot.

## 2.5 Coordination and dynamic role assignment

The Coordination module is responsible for implementing a distributed coordination protocol that allows the robot to assign themselves tasks according to the current situation of the environment [?]. The possibility of using wireless communication has allowed us to port the implementation realized for our Middle Size team. The approach relies on the definition of a set of roles, that are ordered with respect to importance in the domain, and a set of utility functions, one for each role, expressing the capability of a robot to play this role. The values of the utility functions are computed by all the robots in the team for all the roles and these data are exchanged broadcast through the wireless network. A coordination algorithm is then executed by each robot in order to determine robot-role assignment. This algorithm ensures that every role is assigned to one robot and that the robots assigned to the most important roles are those that are in the best situation (according to the values of the utility functions).

The roles played by the robots are implemented as *plans* to be executed by the Deliberation module. Such plans are represented as graphs, in which nodes denote states in the world while edges represent primitive actions that will be performed and cause state transitions. The plans that are executed by the robot may be generated off-line by an automatic planner [?], or written by the user with a graphical tool. The Deliberation Module implements an algorithm that visits the graph activating and deactivating actions according to the current state of the robot.

Moreover, in the Legged League the distributed task assignment was not the only requirement for coordination, since in several cases also the problem of resource conflict must be addressed (e.g. two defender rule). Therefore, we have added a mechanism of action synchronization based on explicit communication that has been used for example in the coordination challenge in RoboCup 2002. In particular, action synchronization is implemented by a sensing action and by a while loop in the plan that has the meaning of waiting until a specific condition is true. In this way, one robot can wait until the other robot sends a signal indicating that it is ready to perform a coordinated action (see [?] for details).

## 3 Analysis of the Results of RoboCup 2003

The results obtained during the official matches and test games performed by our team during the competition RoboCup 2003 have been very satisfactory for our group. We were able to win 4 of the 5 games in the preliminary round, qualifying for the quarter finals.

Apart from the scores of the matches, we have been able to analyze the behavior of the robots and to evaluate (although not in a systematic way) the effectiveness of the techniques developed.

The first general observation of the behavior of our team is that we performed well when playing against teams having robots generally moving slower or as fast as ours, while we had problems with teams having robots moving faster than ours. Although this is a general remark, we believe that different strategies should be performed according to the opponent team and that the speed of the opponent robots is a good parameter on which deciding an appropriate strategy. The analysis of our games showed us that we had defined a good strategy for playing against teams with about the same speed as ours, while the same strategy was not adequate for faster teams.

The second observation obtained from a visual inspection of the games, also after viewing recorded matches, is that the coordination mechanism we have implemented was very effective. In fact, there were very few situations in which two of our robots obstructed each other, and the dynamic change of roles was very effective in many situations in which the robots were able to score a goal. Moreover, we have defined our coordination schemas in such a way to have a robot always acting as a defender and, in fact, most of the play time one of our robots was in a defending position.

We also noticed a significant improvement of the performance of our team from game to game. We believe this was mainly due to the flexibility offered by the possibility of high-level programming of the robots, obtained by generating and writing plans in the form of transition graphs, as described in Section ???. This capability was particularly effective during the competition for a fast and easy-to-debug improvement of the behavior of the players.

Finally, also localization was very effective and sufficiently precise: for example, we never kick the ball towards our own goal, and the goal keeper and the defender were always able to return in the correct position, after being moved by other robots or when picked up by the referee.

## 4 Conclusion

The realization of the soccer software for the AIBO has been for our research group a testbed for our techniques developed also in other contexts of mobile robotics. In particular, we have successfully implemented on the AIBO robots action planning and coordination, that were used in other applications and on other robotic platforms.

However, there are still a number of problems that we have to address and to solve effectively and that are part of our future work: 1) implementing a path planning method, at this moment we do not have any path planning for reaching a point while avoiding obstacles, but just use a very simple reactive technique to avoid collisions; 2) having a larger set of kicks to use in different conditions (including ball interception for the goalkeeper); 3) improving the behavior of the goalkeeper; 4) defining a better team strategy when playing against fast robots; 5) developing techniques for estimating the position of the opponent players in order to improve the game strategies.

## References

1. J. Bruce, T. Balch, M. Veloso. Fast and inexpensive color image segmentation for interactive robots In Proc. of Int. Conf. on Intelligent Robots and Systems (IROS'2000), 2000.
2. C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo, A. Sgorbissa: Coordination among heterogenous robotic soccer players. In Proc. of Int. Conf. on Intelligent Robots and Systems (IROS'2000), 2000.
3. Castelpietra, C., Iocchi, L., Nardi, D., Rosati, R.: Design and Implementation of Cognitive Soccer Robots. In Procs. of The RoboCup 2001 International Symposium. (2001)
4. A. Farinelli, G. Grisetti, L. Iocchi, D. Nardi: Coordination in Dynamic Environments with Constraint on Resources. In Proc. of IROS Workshop on Cooperative Robotics, 2002.
5. Iocchi, L.: Design and Development of Cognitive Robots. PhD thesis. Dipartimento di Informatica e Sistemistica, Universitadi Roma "La Sapienza". (1999)
6. Iocchi, L., et al.: S.P.Q.R. Wheeled Team: Robot Soccer World Cup V. Springer Verlag. (2001)
7. Iocchi, L., Nardi, D., Rosati, R.: Planning with sensing, concurrency, and exogenous events: logical framework and implementation. In Procs. of KR2000. (2000)
8. Iocchi, L. and Nardi, D.: Hough Localization for mobile robots in polygonal environments. Robotics and Autonomous Systems, vol. 40, pp. 43-58, 2002.
9. Nardi, D., et al.: S.P.Q.R. Legged Team: Robot Soccer World Cup VI. Springer Verlag. (2002)
10. S. M. Smith, J. M. Brady. S.U.S.A.N. - A new approach to low level image processing. International Journal of Computer Vision, vol. 23, pp. 45-78, 1997