ISocRob 2009
Team Description Paper

Pedro Lima, João Messias, João Santos, João Estilita, Marco Barbosa,
Aamir Ahmad, João Carreira

Institute for Systems and Robotics, Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisbon, Portugal
http://socrob.isr.ist.utl.pt

Abstract. This paper describes the status of the ISocRob MSL robotic soccer team as required by the RoboCup 2009 qualification procedures. Since its previous participation in RoboCup, the ISocRob team has carried out significant developments in various topics, the most relevant of which are presented here. These include self-localization, 3D object tracking and cooperative object localization, motion control and relational behaviors. A brief description of the hardware of the ISocRob robots and of the software architecture adopted by the team is also included.

1 Introduction and Overview

The SocRob project was created in 1997 by the Intelligent Systems Laboratory of the Institute for Systems and Robotics at Instituto Superior Técnico (ISR/IST), Technical University of Lisbon, with its primary research focus on applications involving cooperative robotics and multi-agent systems. The ISocRob team is the project's case study on soccer robots, and has regularly participated in RoboCup Middle-Size League since 1998, in the RoboCup Soccer Simulation League in 2003 and 2004, and in the RoboCup Four-Legged League in 2007, in a joint effort with the Italian team SPQR.

This paper aims at describing the current status of the ISocRob team as of 2009. In Section 2 the hardware of the robotic soccer platforms currently in use by the team is described. Section 3 then describes the major accomplishments of the research carried out by the project since its last participation in RoboCup.

2 Hardware

The omnidirectional robotic soccer platform currently used by the ISocRob team, the OmniISocRob platform, was developed jointly between ISR/IST and the Portuguese SME IdMind. The following are the most relevant details regarding the capabilities of its actuators and sensors:

Actuators:

- Each of the robot's three Swedish wheels is actuated by a MAXON DC motor (model RE35/118776), through a MAXON gear (model 203118) with a reduction of 21:1, providing the robotic platform with a maximum translational speed of approximately 3.5 m/s and maximum rotational speed of 20 rad/s;
- In order to kick the ball, an electromagnetic strength controlled kicker is used;
- To aid in ball dribbling, a rolling drum is present near the kicker, with controllable rolling speed and elevation.

Sensors:

- The robot's vision system is based on an AVT Marlin F-033C firewire camera, which is equipped with a fish-eye lens providing a field-of-view of 185°, facing downwards. This dioptric system endows the robot with omnidirectional vision, capable of detecting relevant objects (such as the ball and other robots) at a distance of up to 5 m. This particular setup is also less sensitive to vibrations caused by the robot's motion than the previously used catadioptric system;
- Each of the robot's motors is coupled to a 500 CPR encoder for motor control and odometry;
- An AnalogDevices rate-gyro (XRS300EB) is present to improve self-localization.

Each of these components is powered by two packs of 9Ah NiMH batteries per robot.

In this robotic platform, the software architecture (which accounts for most of the required computational power) runs on a NEC FS900 laptop, equipped with a Centrino 1.6GHz CPU and 512Mb of RAM, which is connected to the robot's sensors and actuators through plug-and-play connections (USB and FireWire).

3 Addressed Research

This section points out the major research topics carried under the ISocRob team since its last participation in RoboCup Middle-Sized League in 2007.

3.1 The MeRMaID Software Architecture

The SocRob project is currently using MeRMaID (Multiple-Robot Middleware for Intelligent Decision-making) as its software architecture. Its ongoing development was marked by the release of version 1.0 in 2007 [1]. MeRMaID incorporates into a software architecture the most fundamental items that all robotic systems share, such as sensors, actuators and control software. By being sufficiently generic, it provides guidelines that developers should follow when implementing their solutions. The service-oriented design of the MeRMaID middleware is based

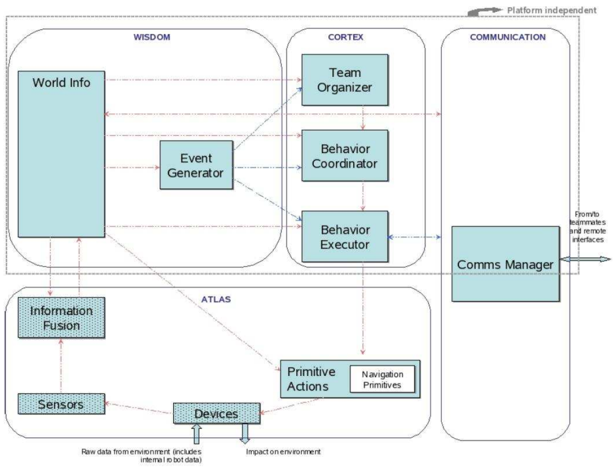

Fig. 1. Block diagram of the MeRMaID functional architecture. The block denotes main modules, denotes active objects, denotes multiple active objects, and denotes general objects (in object-oriented programming conventions). The indicates data flow, while indicates events flow.

on the Active Object pattern. Active Objects are objects that decouple method execution from method invocation in order to simplify synchronized access to an object that resides in its own thread of control. These objects retain their own execution context and execution flow. All processing is done within this context. MeRMaID possesses a modular structure. The main available modules are Atlas, Communication, Wisdom and Cortex. The Atlas module is where the interaction with the world occurs. All direct sensing and acting activity is performed within Atlas, perceiving and producing effects on world. Communication with other robots, external interfaces and the referee box takes place through the Communication module. Other modules connect to the Communication model to access the content of the exchanged messages. The Wisdom module is where the relevant information about the world, such as robot postures, ball position or current score, is kept and managed. This information can be obtained either by sensor information or messages received from teammates or the referee box (through the Communication module). The Cortex is where the decision process takes place, based on information retrieved from the Wisdom module. The Cortex connects to the Communication module to communicate with other robots, in order to carry out cooperative behaviors. MeRMaID has been used in the last 2 years in the European project URUS, where it successfully provided support for the integration of the software developed by the project partners.

3.2 Motion Control

Previous to the introduction of the OmniISocRob platform, the ISocRob team used differential-drive robots (Nomadic Super Scouts II). However, since the new omnidirectional platform came into use, most of the navigation primitives had not yet been updated to account for the holonomicity of these robots. To solve this problem, the ISocRob team developed new motion control solutions for the ball interception and dribbling tasks, specifically for omnidirectional robots. The approach taken by the ISocRob team, described in [3], was to extend existing motion control solutions in the field of robotic manipulators to the case of holonomic mobile robots, and particularize them to relevant tasks in the robotic soccer domain.

The proposed solution to the ball interception problem relies on a combination of trajectory tracking and *proportional navigation* techniques. In the context of moving object interception, the application of proportional navigation techniques usually results in shorter interception episodes, and react better to unexpected changes in the target's motion. However, by themselves, they are not capable of matching the velocity of the target at the instant of interception, which justifies the use trajectory tracking in the final instants of the interception process. Successful interception is then achieved by properly selecting the instants where the control of the robot should switch between trajectory tracking and proportional navigation.

In order to dribble the ball efficiently, the Interface-Control scheme introduced in the context of coordinative robotic manipulation, was adopted by the ISocRob team and extended to the field of mobile robotics. This approach relies on the calculation of the force that must be applied to the ball for it to reach the desired state. This is done through a dynamic model of the ball and a suitable 'object controller'. In order to translate this required force vector into a suitable displacement of the mobile robot, an appropriate controller for the robot must be used that is able to independently follow force and position references, which is accomplished through Hybrid Position/Force control.

An obstacle avoidance algorithm was also implemented, that is compatible with the proposed control solutions. This algorithm deflects the linear velocity vector of the mobile robot so that it passes tangent to the obstacles in configuration space (which are assumed to be circular). Besides being a computationally light algorithm, and not being subject to local minima, the paths followed by the robot while using this algorithm typically approach the globally shortest paths in the environment.

3.3 Self-Localization

Until recently, the self-localization of the ISocRob robots was accomplished through a Kalman Filter, which would make use of certain features in the field (such as the color of the goals) which are not available anymore according to the current RoboCup regulations. This has prompted the ISocRob team to implement a new self-localization algorithm, based on the popular Monte Carlo Localization (MCL) approach. The new approach, fully described in [2], makes use of information about the visible field lines to determine the posture of the robot, and takes advantage of the installed gyroscope to resolve ambiguity issues due to the symmetry of the field. As with the traditional MCL approach, a predefined number of particles are uniformly spread across the field at the algorithm's start. The prediction step of the algorithm then relies on a fusion of odometry and gyroscope data (in a technique called gyrodometry) to account for the robot's movement, and all particles are displaced accordingly. In the update step, a fixed number of points representing the field lines are obtained from each camera image through simple morphological operations that isolate the image's green-white transitions. A weight is then assigned to each particle based on the distance between the projected line points and the nearest field lines. In the resampling step, particles are drawn from the particle set according to their weight, and so the particles that best match the robot's real location have a greater chance of being redrawn. By repeating this sequence of steps the particles eventually become clustered around the robot's real posture. To reduce the computational effort, the algorithm is able to dynamically alter the number of particles that are used, since it is redundant to possess a large amount of particles in a small cluster around the robot's posture. In the event that the robot becomes lost (i.e. its estimated posture does not match its real posture), the required number of particles is increased, and new, uniformly distributed particles are created and inserted into the particle set.

3.4 3D Ball Tracking

An important aspect of any robotic soccer player is its ability to detect and track the soccer ball in a reliable manner. Since these robots are already able to elevate the ball from the field of play in some situations (by performing a high kick for example), it is important to be able to track the ball in a three-dimensional space. Furthermore, since it is expected in the near future that these robots must be able to identify an arbitrarily colored soccer ball, the dependence on ball-detection algorithms that rely on color information must be reduced. To overcome these limitations, the ISocRob team has implemented a 3D ball tracking algorithm, which makes use of a particle filter to detect and track a ball based on shape information [4]. The identification of the ball in each image captured by a robot's camera is based on Taiana's [7] ball projection model. A 3D model of the ball is used to calculate its 2D contour projected on the image. The particle filter is initialized by uniformly spreading a fixed number of ball hypothesis (particles) on the ground, in a 5 meter area surrounding the robot. This allows for a reduction in the search state space, as it is assumed that the ball is on the floor, and constrain the detection according to the camera resolution. These particles contain information about the ball's position and velocity in the 3D space. The prediction step then takes into account both the movement of the ball and of the robot to displace all particles accordingly and allow continuous tracking of the ball. In the update step, two YUV histograms are obtained, for both the inside and outside boundaries of the ball. The likelihood of a given particle is obtained by applying a similarity metric between these histograms, which allows tracking of arbitrarily colored balls since the reference color model for the inside boundary is not provided (i.e. the likelihood of a particle is simply a function of the mismatch between both boundaries). The particles are then resampled according to a low-variance technique that ensures diversity in the particle set.

3.5 Cooperative Object Localization

For any soccer team (robotic or not) to function properly it is fundamental that as many of the players as possible are aware of the location of the soccer ball. Since the range at which each robot is able to detect the ball is limited, it is important for the team to share the ball information between its players, becoming in this context a team of cooperative sensors. Non-parametric representations of the probability density function (pdf) of each of these sensors, as obtained through the particle filter based approach described in Section 3.4, are an efficient way of dealing with the sensor's nonlinearities, but, in their direct form, their to communication to other agents would require the transmission of large amounts of data (namely, the particle set). Under these circumstances, a more efficient representation of the sensor's information may be obtained through a Gaussian Mixture Model (GMM), which can be obtained from the original sample-based data through the application of parameter estimation algorithms such as the Expectation Maximization (EM) algorithm [4]. The lower dimension parameters of the GMM may then be transmitted efficiently to other team members. Instead of fusing all the data from the sensors into a single estimate of the ball's state, which assumes that the measurements taken by each sensor always contribute towards a more accurate estimate, the implemented cooperative perception model takes advantage of the GMM representation in two distinct forms. One is to improve the local ball particle filter in a distributed fashion way by injecting new particles drawn directly from the received GMMs. The other is to compute a ball team estimate directly from the received GMMs target distribution using Covariance Intersection (CI). This ball team estimate can also be used to improve each robot's self-localization, since a 'lost' robot may use this information as a landmark when updating its own belief.

3.6 Relational Behaviors

With the increasing capabilities of robotic soccer platforms, both as an individual agent as part of a team, cooperation between these robots is no longer just a significant research contribution. In some situations, this cooperation is of utmost relevance for the success of the team . This occurs, for example, in foul-taking situations, where the current RoboCup MSL rules state that for a goal to be valid after a free kick, at least two different robots must have touched the ball. One of the robots, referred to as the kicker, will have to move to the ball and kick in the direction of his partner - the receiver - who should intercept the ball. In these situations, a *relational behavior* is required. Relational behaviors are, in this sense, robotic behaviors that concern more than one player, and require the establishment of a *commitment* between the involved players. This commitment ensures that all of the players will pursue the execution of the behavior until its end. A relational behavior execution ends with its success or failure. In both cases, the intervening robots must be in agreement about the state of the commitment, meaning that if one of the robots wants to break the commitment it must inform the other(s). This way none of robots will become blocked in a deadlock situation. Another essential requirement for a relational behavior is synchronization. If there is no synchronism between the robots involved in the relational behavior, then this behavior is likely to fail. Due to this, an efficient commitment management is required and a good synchronization between the participants is essential. In [5, 6], a systematic methodology for the design of such relational behaviors is presented, based on the Joint Commitment Theory [8], and the necessary commitment and synchronization mechanisms are defined. Based on these methods, a set of relational behaviors that deal with common situations during a robotic soccer match were subsequently developed for the ISocRob team using Petri nets and implemented in the MeRMaID software architecture. In [9, 10], Joint Commitment Theory is again applied, using Petri nets, to implement soccer-oriented relational behaviors on a team of AIBO robots.

3.7 Visual tracking of teammates for the implementation of relational behaviors using implicit communication

Human-robot or robot-robot teamwork can be based on either explicit (e.g., wireless) communication or implicit (e.g., based on visual observation of the teammate) communication. The latter is more difficult to achieve, but avoids broken communication links and relies on the exchange of "natural" communication signals. A future goal of the ISocRob team is to model and implement teamwork commitment and synchronization signal exchange between the soccer robots, using implicit communication. This requires individual tracking of teammates and opponents visually by every team member in addition to the existing obstacle and ball tracking techniques. Tracking teammates involves identifying a double-ellipse shaped colored marker which meets certain requirements as mentioned in the RoboCup MSL 2009 rules and regulations. Each team member can be recognized by a specific orientation of the double-ellipse marker on them. A particle filter-based algorithm is then used to visually track these markers. This approach implements a robust technique developed in [11] with some innovative modifications to the update step of the particle filter, which reduce its computational complexity. The basic concept behind this technique is a Hough transform [12] using a lesser dimensional Hough space than it is theoretically required. An ellipse can be described using five parameters, namely the center coordinates (X_0, Y_0), half lengths of major and minor axis (a, b) and the orientation angle α with respect to to the X-axis in Cartesian coordinates. This implies that a traditional Hough transform would require a 5-dimensional Hough space to rigorously find the ellipses in a 2-D plane. In this approach, the Hough transform is accomplished using a single dimension accumulator. The major modifications to the algorithm in [11] is in the calculation of parameter b, the minor axis half length which eventually is the accumulator parameter. Here we calculate b using c, d, e and f and angle τ for any given point (x, y) which lies on the ellipse.

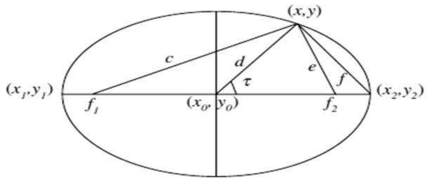

Fig. 2. Calculation of ellipse parameters

The foci of the ellipse are located at f_1 and f_2 (Figure 2). The property of the ellipse exploited here is that the sum of distances from any point on ellipse to the foci $(c + e)$ is always equal to the length of the major axis $(2a)$. Another modification is that the array of edge points, which form candidates to vote for an ellipse, gradually contracts as we find legal instances of ellipses in the image. This reduces the computational complexity further.

References

1. Barbosa, M., Ramos, N., Lima, P.: MeRMaID Multiple-Robot Middleware for Intelligent Decision-Making. Proc, of IAV2007 - 6th IFAC Symposium on Intelligent Autonomous Vehicles, Toulouse, France (2007)
2. Messias, J., Santos, J., Estilita, J., Lima, P.: Monte Carlo Localization Based on Gyrodometry and Line-Detection. Proc. of ROBÓTICA2008 - 8th Conference on Mobile Robots and Competitions, Aveiro, Portugal, pp. 39-50 (2008)
3. Messias, J.: Task Specific Motion Control of Omnidirectional Robots. MSc thesis, Instituto Superior Técnico, Lisbon (2008)
4. Santos, J.: Multi-Robot Cooperative Object Localization. MSc thesis, Instituto Superior Técnico, Lisbon (2008)
5. Torres, J: Relational Behaviors For Soccer Robots - Quantitative Analysis. MSc thesis, Instituto Superior Técnico, Lisbon (2007)
6. Antunes, T: Relational Behaviors For Soccer Robots - Qualitative Analysis. MSc thesis, Instituto Superior Técnico, Lisbon (2007)
7. Taiana, M.: 3d model-based tracking with one omnidirectional camera and particle. MSc thesis, Laboratorio di Intelligenza Artificiale e Robotica, Politecnico di Milano (2007)
8. Cohen, P. R., Levesque, H. J.: Teamwork. Nous, vol. 35, pp. 487-512 (1991)
9. Ziparo, V., Iocchi, L., Palamara, P., Nardi, D., Costelha, H.: PNP: A Formal Model for Representation and Execution of Multi-Robot Plans. In: Proc. of AAMAS 2008 - 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems, Estoril, Portugal (2008)
10. Ziparo, V., Iocchi, L., Palamara, P., Nardi, D., Lima, P.: Teamwork Design Based on Petri Net Plans, In: Proc. of RoboCup International Symposium 2008, Suzhou, China (2008)
11. Chia, A.Y.S., Leung, M.K.H., How-Lung Eng, Rahardja, S.: Ellipse detection with Hough transform in one dimensional parametric space. IEEE International Conference on Image Processing (2007)
12. Duda, R. O.,Hart, P. E.: Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM, vol. 15, pp. 1115 (1972)