SPQR + ISocRob
RoboCup 2007 Qualification Report

L. Iocchi[1], L. Marchetti[1], D. Nardi[1], P. U. Lima[2], Marco Barbosa[2], Hugo Pereira[2], Nuno Lopes[2]
[1] Dipartimento di Informatica e Sistemistica
Sapienza Università di Roma, Italy

[2] Institute for Systems and Robotics
Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisboa, Portugal

1 Introduction

SPQR + ISocRob is a joint team between two research groups: SPQR (Italy) and IsocRob (Portugal). SPQR[1] (Soccer Player Quadruped Robots) is the group of the Faculty of Engineering at University of Rome "La Sapienza" in Italy, that is involved in RoboCup competitions since 1998 in different leagues (Middle-size 1998-2002, Four-legged since 2000, Real-rescue-robots since 2003, Virtual-rescue and @Home in 2006). The IsocRob[2] team represents the competition side of the SocRob (Soccer Robots/Society of Robots) project [5], a research endeavor of the Intelligent Systems Laboratory of the Institute for Systems and Robotics at Instituto Superior Tcnico (ISR/IST), Technical University of Lisbon, which started in 1997. ISocRob has participated in RoboCup Middle-Size League (MSL) almost every year from 1998 to 2006, as well as in the Soccer Simulation League in 2003 and 2004, and started a new Four-Legged League team in 2005.

This report presents the development of the joint team, which is intended to be formed as a coordinated heterogeneous multi-robot team. It also describes development efforts of the two groups for RoboCup 2007.

2 Heterogeneous multi-robot team

The main research topic underlying the development of this joint team is heterogeneous multi-robot coordination. Our approach is to have different software solutions for the basic modules of the robots: i.e., each team will develop its own code for the basic robot functionalities (such as perception, localization, control, etc.). The robots will instead share a common formalism for high level plan representation and a coordination protocol that will be executed for dynamic task assignment during the game.

While developments of the two groups in the basic functionalities are reported in the next sections, here we will briefly sketch the common high-level plan representation formalism and the coordination protocol.

2.1 High-level plan representation

A Petri net based approach has been followed by both teams for the modeling of behaviors and their coordination. Due to their capability to model concurrency, Petri nets are particularly suited for modeling relational behaviors (those involving two or more teammates, such as performing a pass or coordinately defending the goal), as they can capture the concurrent nature of exchanging messages between teammates while each of them is executing their own primitive actions.

[1] spqr.dis.uniroma1.it
[2] socrob.isr.ist.utl.pt

In our work, behaviors are modeled by PNs as follows:

- each place in the Petri net is labeled by an associated primitive action (e.g., moving to a given posture, kicking the ball, intercepting the ball) or resource (e.g., availability of an object, or of a robot, or of a communication signal);
- each transition in the Petri net is labeled by an event, 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 token in a place means that the primitive action associated to that place is currently active (i.e., it is running) or that the resource labeling that place is currently available. Transitions are enabled when all its input places have at least one token each, meaning that the pre-conditions for the next step are satisfied. A transition is fired if its enabled and the associated event occurs.

Models of relational behaviors have been developed [9] based on concepts borrowed from Joint Commitment Theory [2]. In a relational behavior, communication (either implicit, e.g., by mutual observation, or explicit, e.g., using communications) between the involved players is essential.

The exchanged messages can be of two different kinds:

- commitment related messages - to establish or break a commitment - meaning that every robot getting involved in a relational behavior commits either to achieve its goal or to let the others know when it needs to withdraw;
- behavior flow messages - to synchronize the execution of the behavior by both parties. In the pass example, the player who will receive the pass must send a message stating that it is waiting for the pass.

Petri nets are used to model both types of messages. In Figure 1, a simple example of a Petri net model of a pass, involving commitment and synchronization.

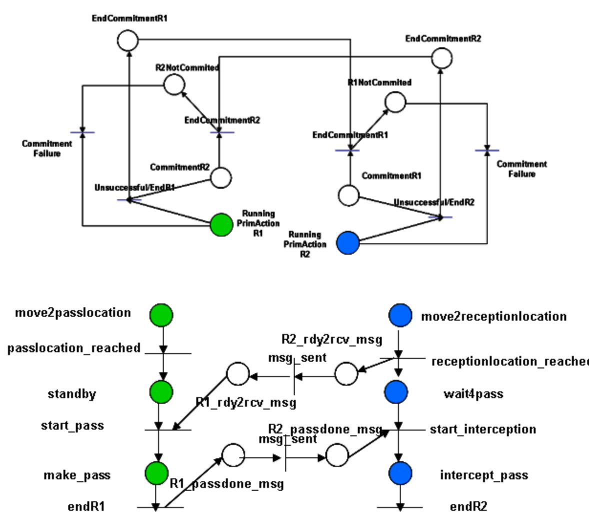

Figure 1: Simple PN model of a pass, showing commitment and synchronization. The commitment block repeats for every pair of places with the same color as the corresponding commitment module colors

Finally, an execution mechanism for Petri Net Plans (PNP) has been also presented in [11]. This actually allows for distributed execution of multi-agent plans represented as described above.

2.2 Distributed coordination protocol

Distributed task assignment for heterogeneous mobile robots has already been studied in [4]. In the SPQR+ISocRob joint team we intend to adopt and possibly extend such an approach. The advantages in using such a distributed task assignment process are in its effectiveness and robustness. In particular, roles are ordered by an important factor (e.g., the attacker going to the ball is more important than the defender) and the sub-optimal task assignment guarantees that more important roles are always correctly assigned. Moreover, the approach is robust to communication problems and performance of the team degrade gracefully with network latency and communication loss.

As an extension to the above method, we will consider action synchronization to address situations in which conflicts in executing tasks arise. This is useful both in some game situations (e.g., for coordinating goal-keeper and other robot actions when the ball is around the own penalty area) and in the Passing Challenge.

3 SPQR

Since 2005, our team has adopted the German Team Architecture [8]. This choice has been motivated mainly by its high modularity, allowing for developing software modules realizing specific functionalities rather independently from the others. So far we have replaced all the important modules of the original GT2004 code: color segmentation has been replaced with a module for dynamic color segmentation [3]; image processing now uses a different approach to detect the objects, which is based on a limited number of perceptual pixels that activate object recognition; localization is implemented with a particle filter based method [7] behavior control uses Petri Net plan representation language [11], coordination is based on a distributed task assignment [4]. Source code used in RoboCup 2006 and a document describing modules that have replaced GT2004 ones are available on-line[3].

In the following we briefly describe our work in preparation of RoboCup 2007.

3.1 Localization

Our approach to localization uses a probabilistic approach based on particle filters. In [7] we have compared different solutions based on particle filters investigating the use of two different strategies: the well known Sample Importance Resampling (SIR) filter, and the Auxiliary Variable Particle filter (APF). As a result of the experiments performed we have detected situations where one strategy is better than the other as well as hints about the use of sensor resetting, that is common in this kind of implementations. We are currently working on integrating in the localization technique information about the game state (or in general about the task state) aiming at choosing the localization strategy and setting that is more suitable for the current situation.

In addition, a new procedure for line detection has been implemented and it will be integrated in the localization process. The lines will be used more than previous years for localization, thus a more robust line detection system is needed.

Finally, due to changed structure of the goal, it is possible to use the goal's poles as landmark. This kind of feature is not unique, because we have now two poles, but they are more reliable than the previous configuration. We will integrate such observations in the localization process as well.

3.2 Learning

Soccer robots, and in particular legged ones, require a fine tune of the parameters specially in the implementation of behaviors and basic control actions and in the strategic decisional processes.

In [1] we have performed experiments on layered learning approaches for learning optimal parameters of basic control routines, behaviors and strategy selection. We compared three different methods in the different layers: genetic algorithm, Nelder-Mead, and policy gradient. Moreover, we study how to use a 3D

[3]Download section in `spqr.dis.uniroma1.it`

simulator for speeding up robot learning. The results of this experimental work on AIBO robots have been not only the realization of improved behaviors, but also a more effective learning methodology that makes use of a simulator.

3.3 3D Simulator

A 3D Simulator for legged robots (and AIBO in particular) has been presented in [10]. This extension has been included in the official USARSim release[4].

This tool has been extremely useful to tune behaviors of the robot and for debugging purposes. We have extended the framework in order to run multiple robot in the simulated environment running the same code that is executed on the real robot. Due to the extensive computational requirements of USARSim, we are able to run only a 3 vs. 3 game. However, this is sufficient to test many kinds of behaviors including different strategies for coordination.

4 ISocRob

ISocRob's main research challenges at the moment concern teamwork issues, with a special focus on:

- **Cooperative Navigation**, particularly formation control, so as to distribute the robots across the field dynamically, e.g., by prescribing at each step the desired relative distances and orientations between teammates (see one of ISocRob's qualification videos), but also cooperative localization, where the information about the localization of team members can be improved by mutual observation using Bayesian methods.

- **Cooperative Plan Execution**, mostly related to relational behaviors, where each involved robot executes part of the plan while regularly synchronizing it with the related teammates (see Section 1).

The low-level supporting modules are, whenever possible, borrowed from other teams, e.g., we will be using TeamChao's color segmentation and recognition, active vision, and locomotion modules (originally developed by the French Team "Les 3 Mousquetaires"), as well as low level actions, e.g., kicks, and SPQR's communications module.

ISocRob's current development concept consists of developing behaviors for the robots and testing them in a realistic simulation environment. In order to do that, a considerable amount of work has been done developing OpenSDK (http://opensdk.sf.net), a free open-source implementation of Sony's OPEN-R Development Kit. It aims at compiling and running code that uses that API on a normal PC, to reduce the development cycle time and to ease the transition to a new legged robot platform. The ISocRob team has been responsible by a considerable part of OpenSDK development work, which enables running the code in the USARSim simulator, by feeding it with SONY's Open-R primitive actions and fetching the sensor inputs (camera, joint angles, etc..). The OpenSDK code is already in place, but the protocol between USARSim and OpenSDK is being written. Moving from the simulator to the actual robots should consist of replacing calls to OpenSDK plus USARSim by calls to Open-R.

4.1 Middleware

This year, ISocRob team has developed MeRMaID (Multiple-Robot Middleware for Intelligent Decision-making), a robot middleware to support the fast implementation of behavior-based plans, either designed by humans, or generated by automatic planners [6]. MeRMaID's supporting concepts are *primitive actions* (the atomic element of a behavior, usually consisting of some calculations plus a call to a navigation primitive or the direct activation of an actuator), *behaviors* (macros of primitive actions grouped together using some appropriate representation, e.g., state machines or Petri nets), *predicates* (Boolean relations over the domain of world objects), *events* (an instantaneous occurrence which denotes changes of (logical conditions over)

[4]sourceforge.net/projects/usarsim

predicates, as well as events received from sources external to a robot through a communication channel, such as those sent by the referee box), and *roles* (subsets of behaviors, possibly dynamically selected). These definitions imply that *predicates* are implemented by information processing monitoring algorithms that achieve Boolean classification from raw data (e.g., by detecting whether a given object is seen by the robot or not).

The current block diagram which describes MeRMaID is depicted in Figure 2. In the figure, the **World Info** stores the relevant information about the world, such as robot postures, ball position or current score. This information can be obtained either by sensor (fusion) 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 code implementing a behavior. The **Behavior Coordinator** selects which behavior to run next, while the Team Organizer selects which role the player will perform.

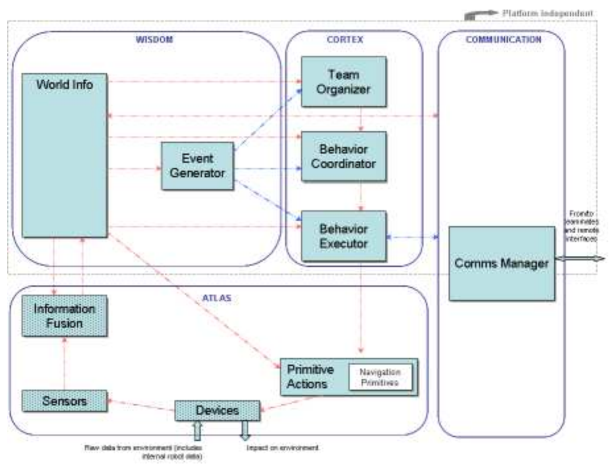

Figure 2: MeRMaID block diagram.

MeRMaID's framework allows several solutions concerning the actual implementation of the different decision levels. Currently, we have available tools to design plans based on Petri nets (including a graphical user interface (GUI)), state machines (including a GUI) and Fuzzy Decision Making (using XML). One of SPQR's team students is currently working in Lisbon with the ISocRob team and, as part of his work, he is also making available SPQR's Petri net design tool for MeRMaID. Though the concepts concerning plan representation by Petri nets are quite similar, there are slight differences in the actual implementation which makes the availability of both tools convenient.

4.2 Formation control

Work on formation control was developed for the 2006 challenges, which is now being integrated in the code of the team to be run during games. The goal is to distribute the robots across the field dynamically, e.g., by prescribing at each step the desired relative distances and orientations between teammates, instead of assigning field regions to each player, depending on its role.

The algorithm used consists of dynamically selecting a leader (usually the robot closer to, or who has the ball) and a geometric configuration for the formation composed of the selected leader and the remaining players. The leader can change dynamically, and the formation configuration is deformable, so as to take obstacle avoidance into account. Currently, the robots rely on their self-localization estimates and on wireless communications to exchange them and compute relative distances and orientations.

References

[1] A. Cherubini, F. Giannone, and L. Iocchi. Layered learning for a soccer legged robot helped with a 3D simulator. Submitted to RoboCup Symposium, 2007.

[2] P. Cohen and H. Levesque. Teamwork. *Nous*, 35:487–512, 1991.

[3] L. Iocchi. Robust color segmentation through adaptive color distribution transformation. In *Proc. of RoboCup Symposium*, 2006.

[4] L. Iocchi, D. Nardi, M. Piaggio, and A. Sgorbissa. Distributed coordination in heterogeneous multi-robot systems. *Autonomous Robots*, 15(2):155–168, 2003.

[5] P. Lima and L. Custodio. Artificial intelligence and systems theory applied to cooperative robots. *International Journal of Advanced Robotic Systems*, 3, 2004.

[6] P. Lima, N. Ramos, M. Barbosa, and H. Costelha. Multiple-robot middleware for intelligent decision-making. Technical Report RT-701-07, Institute for Systems and Robotics, Instituto Superior Tcnico, 1049-001 Lisbon, Portugal, 2007.

[7] L. Marchetti, G. Grisetti, and L. Iocchi. A comparative analysis of particle filter based localization methods. In *Proc.of RoboCup Symposium*, 2006.

[8] T. Röfer et al. German team robocup 2004, 2004. Available from `www.germanteam.org`.

[9] B. Vecht and P. Lima. Formulation and implementation of relational behaviours for multi-robot cooperative systems. In *Proc. of RoboCup 2004 Symposium*, Lisbon, Portugal, 2004.

[10] M. Zaratti, M. Fratarcangeli, and L. Iocchi. A 3d simulator of multiple legged robots based on usarsim. In *Proc. of RoboCup Symposium*, 2006.

[11] V. A. Ziparo and L. Iocchi. Petri net plans. In *Proc. of Fourth International Workshop on Modelling of Objects, Components, and Agents (MOCA)*, pages 267–290, Turku, Finland, 2006. Bericht 272, FBI-HH-B-272/06.