SocRob 2013
Team Description Paper

João Messias, Aamir Ahmad, João Reis, Miguel Serafim, and Pedro Lima

Institute for Systems and Robotics, Instituto Superior Técnico
Av. Rovisco Pais 1049-001 Lisboa, Portugal
{jmessias,aahmad,jreis,pal}@isr.ist.utl.pt
http://socrob.isr.ist.utl.pt

Abstract. This paper describes the status of the SocRob MSL robotic soccer team as required by the RoboCup 2013 qualification procedures. The team's latest scientific and technical developments, since its last participation in RoboCup MSL, include further advances in cooperative perception; novel communication methods for distributed robotics; progressive deployment of the ROS middleware; improved localization through feature tracking and Mixture MCL; novel planning methods based on Petri nets and decision-theoretic frameworks; and hardware developments in ball-handling/kicking devices.

1 Introduction

The SocRob (Society of Robots) project was established in 1997 at the Institute for Systems and Robotics at Instituto Superior Técnico (ISR/IST), Technical University of Lisbon, with the goal of studying cooperative robotics and multiagent systems. The SocRob robotic soccer team (formerly ISocRob) is one of the project's case studies. It 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 describes scientific and technical developments carried out by the team since 2011, when it last participated in RoboCup MSL. When appropriate, we cite the team publications on the described topics.

2 Scientific and Technical Challenges

Cooperative Perception With Closed Loop Formation Control

A method for perception driven multi-robot formation control was proposed and implemented. Particle filter-based (PF) cooperative object tracking, developed in [2], was applied as a feedback module in a multi-robot formation control loop. The simulation and real robot results demonstrated the success of its implementation on the SocRob team, as well as in another MSL team (5DPO). The work was part of a joint research project involving these two teams. The penalization weight-based minimization of the formation controller's cost function lets us control different objectives, e.g, inter-robot

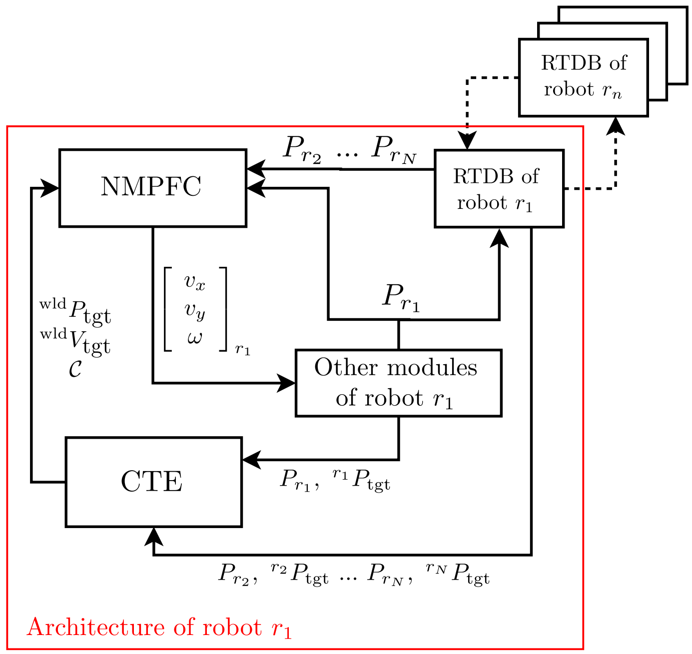

Fig. 1. The formation control loop describing the control-estimator module integration. The above flow diagram represents the architecture of robot r_1 in a team of N robots where the n^{th} robot is denoted by r_n. P_{r_n} denotes the robot r_n's world frame pose (position + orientation) as obtained by its self-localization mechanism (implemented separately from the formation control loop). $^{r_n}P_{\mathrm{tgt}}$ denotes the target's detected position (observation measurements) in the robot r_n's local frame. $^{\mathrm{wld}}P_{\mathrm{tgt}}$ and $^{\mathrm{wld}}V_{\mathrm{tgt}}$ denote the target's cooperatively estimated world frame position and velocity, respectively. $\mathcal{C}$ denotes the target's cooperatively estimated position covariance matrix. The vector $\begin{bmatrix} v_x & v_y & \omega \end{bmatrix}_{r_1}^{\top}$ denotes the velocity set points for the robot r_1, which is the output of the NMPFC at that robot. The block named 'other modules of robot r_1' denotes that robot's low level control and sensor units, e.g, robot wheel controller and target detector (using camera images). This flow diagram is reprinted from page 45 of [10] with modifications in the variable nomenclature.

or target-robot collision avoidance, while creating and maintaining the robot-team formation to minimize the uncertainty of the cooperatively tracked target. Figure 1 describes the integration of the cooperative target estimator (CTE),developed by us, and the nonlinear model predictive formation controller (NMPFC), developed by 5DPO [10] to achieve the formation control loop. This work was accepted to be published in the proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA 2013), Karlsruhe, Germany [3].

Multi-robot Unified Cooperative Localization and Object Tracking

New algorithms and methods for integrated cooperative perception have been introduced and implemented in SocRob. In 2011 we presented the visually shared object-based cooperative robot localization mechanism [8] and the cooperative ball tracking technique [2]. Since both of these methods rely on individual robot's ball tracking and self-localization estimates, there needs to be an integrated mechanism to perform both tasks cooperatively. We introduce an offline method for multi-robot unified cooperative localization and object tracking (UCLT) based on graph optimization. The method treats the tracked object as a moving landmark in addition to the previously known static landmarks in the environment. It first constructs a pose graph which consists of nodes and edges connecting those nodes. The nodes either consists of the states to be estimated, e.g, robots' pose and the tracked ball's 3D positions or the known and fixed states, e.g, static landmarks' positions. The edges correspond to the odometry or observation measurements made by the robots. A least square-based optimization routine, available in the g^2o framework [7], is adapted to perform the optimization of the aforementioned pose graph. The optimized graph is the configuration of nodes that best describes the measurements made by all the robots. The proposed method and its experimental evaluation was accepted to be published in the proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA 2013), Karlsruhe, Germany [4].

ROS Integration and Communications

During the last year, SocRob's implementation was changed to run based on ROS middleware[1]. ROS is architecturally not completely different from the MeRMaID middleware used before. It is, however, much better maintained and gives easy access to many auxiliary and debugging features. Using ROS, it is now much easier to inspect a running system, record data for offline use, access logging data, among other features that allow us to improve the software much faster. ROS also provides some stable and ready to use components, which are gradually replacing some of our custom solutions, while some of our better solutions will eventually become contributions to the community.

One of the aspects in which ROS did not provide a suitable solution was communication. While in one ROS system the communication facilities are great, in a team of five robots it is necessary to have five ROS systems, because the connection is very unreliable and there is a bandwidth limitation. Therefore, we implemented one of the best communication solutions for MSL, the RA-TDMA protocol[11]. Furthermore, while carefully analyzing our communication needs, we decided to augment this solution with a new mechanism, designed to reliably transmit urgent messages, needed to synchronize robots in dynamic game situations[6]. The solution will soon be made available to the community.

Task Modeling and Plan Representation Using Petri Nets

SocRob tasks are modeled using a discrete event approach, and plans are represented by Petri nets [5]. The task model includes controllable events, representing decisions made by the robot, and uncontrollable events performed by other robots or that result from the environment physics. Plans are hierarchical organized so as to execute roles and individual/cooperative behaviors. The basic component of our task models are primitive actions. Our approach enables modeling a robot task, analyzing its qualitative and quantitative properties and using the Petri net representation for actual plan execution.

For modeling purposes, a Petri net model of the environment, capturing the complexity of the environment dynamics, is composed with the (multi-)robot controller model to obtain a single closed-loop Petri net representing the whole task model, i.e., the model of the (multi-)robot system situated in its environment. The Petri net models of the robot controller and of the environment are separately obtained from the automatic composition of simple and modular models manually defined. An example of a controller model for the Score_Goal task is depicted in Figure 2.

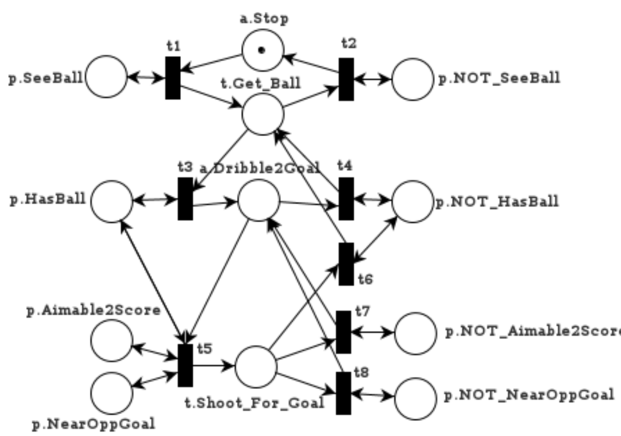

Fig. 2. Petri net model of the Score_Goal task plan. p. prefixes correspond to predicates (associated to transitions); a. and .t prefixes correspond to primitive actions and macro-actions, respectively (associated to places which are marked by the environment model, not shown).

Ordinary Petri net and generalized stochastic Petri net (GSPN) views of the model are used to retrieve logical/qualitative and (probabilistic) performance/quantitative properties of the robot task plans, respectively. Furthermore, we introduced a method to identify the parameters of the stochastic Petri net models from real data, improving the significance of the model. Analysis is applied to the closed loop models. Stochastic Petri net models are used for analysis only (of quantitative properties), while ordinary Petri nets are used both for qualitative analysis and execution. The GSPN view models action effects uncertainty both as plain transition probabilities and as stochastic timed transitions, where transition probabilities are indirectly modeled by the stochastic time elapsed between the start of the action and its end, due to some uncontrollable event; in the end, both models boil down, under some light requirements, to an equivalent Markov Decision Process, that can be solved using existing techniques, from dynamic programming to reinforcement learning. The framework provides a design-analysis-design approach, which leads to improved task plans before executing the plan in the real robots.

Mixture-MCL Based on Feature Detection

Since 2008, the SocRob team has used a custom Monte Carlo Localization algorithm for robot self-localization, and for cooperative localization based on common features (e.g the ball). This algorithm has now been extended, by applying the Mixture-MCL approach introduced in [12].

Our MCL algorithm weighed each particle $\hat{q}_i = [\hat{x}_i, \hat{y}_i, \hat{\theta}_i]$ according to a measure of average distance between a set of points z, detected over the field markings for each camera image, and the closest lines (or circles) in the field according to that particle. The weight measure was then such that $w \sim \Pr(z|\hat{q}_i)$. However, each cloud of "points-on-lines" z did not convey sufficient information to robustly distinguish characteristic field features (such as corners ir the midfield circle) from regular field lines, which naturally hinders localization results, particularly with unreliable odometry or after robot "kidnapping" situations.

In order to take advantage of these characteristic field features, we've extended our vision algorithms to extract information regarding line segments and circles from each line point cloud z. For each z, line segments are extracted by grouping line points into subsets of linear regressors, minimizing total least-squares error. A similar approach is taken for circle detection - in that case, however, the radius of each possible circle is known *a priori* (it is either the midfield circle or a corner arc). The results of this process are shown in Figure 3.

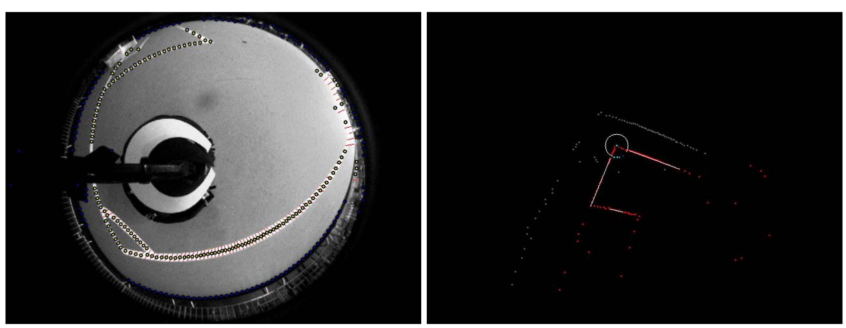

Fig. 3. Left: detection of points on field markings from a camera image; right: extracted linear and circular features from the respective point cloud.

Using this feature-based information f, it is then possible to determine $\Pr(q|f)$, the *dual* of the sensor model, in real time. We do this by taking into account the error associated with each linear/circular fit, using it to define parametric probability density functions over the configuration space of the robot. Each actual feature r_i in the field (the real lines and circles making up the field), when matched with a detected feature

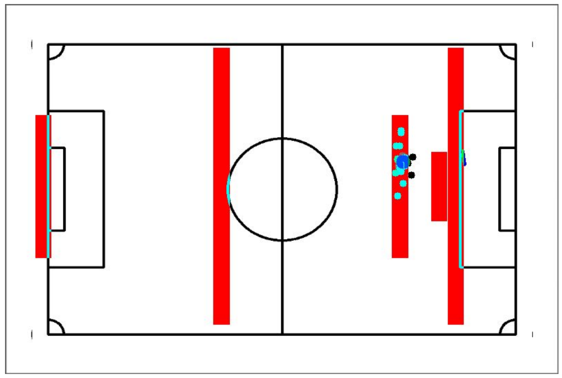

Fig. 4. Building the model $\Pr(q|f)$ from feature data. In this case, only a single line segment was detected (shown to the right of the posture estimate). The range of possible postures for which the detected feature matches an actual linear feature in the soccer field is shown in red. Compass data was used to disambiguate other alternatives. These red areas correspond to the modes of the dual sampling distribution, which are used to sample new particles.

f_i extracted from an image, uniquely describes a set of parametric modes $\Pr(q|f_i)$: a linear feature produces a set of modes which are uniform along its tangent direction, and Gaussian along its normal direction as well as in the relative orientation; a circular feature produces a set of polar modes which are uniform along the relative azimuth of the robot, and Gaussian over the distance to the feature as well as the relative orientation of the robot. We calculate the possible modes for each match (r_i, f_i) , and then discard the modes which have are not consistent with eachother, based on the Kullback-Leibler divergence between the respective distributions of each component of q. In doing so, we obtain a multi-modal representation of $\Pr(q|f)$ which can be used to sample particles $\hat{q}_i$ at each prediction step of a *dual*-MCL algorithm. These particles are then weighed based on the probability that the robot actually moved to each of those particular postures based on its last posture estimate, and resampled accordingly (*i.e.* the dual-MCL algorithm samples the observation model and weighs with the odometry model). The full Mixture-MCL algorithm combines standard (or *forward*) MCL steps with dual-MCL steps, drawn randomly according to a predetermined *mixing factor*. Our resulting implementation allows for an elegant solution of the "kidnapped" robot problem, while using an order of magnitude less particles than our previous MCL solution.

Asynchronous Decision Theoretic Planning

The SocRob team has recently been serving as a case-study for the practical application of decision-theoretic frameworks such as multiagent Markov Decision Processes (MDPs) and Partially Observable Markov Decision Process (POMDPs). This stands as an alternative approach to task modeling and plan execution, which is typically done manually via Petri Nets, as discussed previously. We have shown previously that efficient planning for simple in-game cooperative tasks can be also accomplished through the use of (PO)MDP methods [9]. Such decision-theoretic plans result naturally from the optimization of the sequence of actions for each robot in the team, in the presence of uncertainty regarding the outcome of each action and subsequent information gathered from sensors. However, multiagent (PO)MDPs typically assume that every agent in the team is acting synchronously, that is, that every robot is selecting its actions and reporting its observations to other robot at the same, periodic time instants. In contrast, the operation of a team of robots in a highly dynamic environment (such as RoboCup MSL) requires that agents react immediately to detected events, and so team-wide synchronous action selection is not desirable. We have developed novel approaches to decision-theoretic modeling which lift this assumption of synchrony, and are therefore particularly suited to the real-time control of teams of robots. Our new decision-theoretic models are event-driven and semi-Markovian, while being solvable by existing (PO)MDP methods.

Electromagnetic Kicker Development

An upgraded version of the electromagnetic kickers used in the SocRob robots is under development. Our kicker system stores its energy in 100V capacitors, until it is released to a solenoid, generating cinetic energy to produce the kick. This upgrade will reduce the time for charging the capacitors, shortening the time between kicks. It is being achieved by using a Current-mode Boost converter with a higher duty cycle than previously possible, which will be adjusted through current feedback, maximizing the energy transfer.

References

1. ROS Wiki. http://www.ros.org/wiki/.
2. A. Ahmad and P. Lima. Multi-robot cooperative object tracking based on particle filters. In *Proc. of the European Conference on Mobile Robots (ECMR 2011)*, Örebro, Sweden, Sep 2011.
3. A. Ahmad, T.P. Nascimento, A. S. Conceição, Moreira A.P., and P. U. Lima. Perception-driven multi-robot formation control. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.
4. A. Ahmad, G. D. Tipaldi, W. Burgard, and P. U. Lima. Cooperative robot localization and target tracking based on least square minimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.
5. H. Costelha and P. Lima. Robot Task Plan Representation by Petri nets: Modelling, Identification, Analysis and Execution. *Autonomous Robots*, 33(4):337–360, 2012.
6. João Reis. Distributed Communications System for Multi-Robot Systems. Master's thesis, Instituto Superior Técnico, Av. Rovisco Pais, 1, October 2012.
7. R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.
8. P. Lima, P. Santos, R. Oliveira, A. Ahmad, and J. Santos. Cooperative localization based on visually shared objects. In *Proc. of RoboCup2010 Symposium*, Singapore, 2010.
9. J. Messias, M. Spaan, and P. Lima. Multi-robot planning under uncertainty with communication: a case study. In *AAMAS 2010 Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains*, Toronto, Canada, 2010.

10. Tiago Pereira Nascimento. *Coordinated Multi-Robot Formation Control*. Phd, Porto University, 2012.
11. F. Santos, L. Almeida, and L.S. Lopes. Self-configuration of an Adaptive TDMA wireless communication protocol for teams of mobile robots. In *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, pages 1197–1204. IEEE, 2008.
12. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1):99–141, 2001.