BEHAVIOR-BASED COOPERATION WITH APPLICATION TO SPACE ROBOTS*

João Sequeira, Pedro Lima, M. Isabel Ribeiro, João Sentieiro
Instituto Superior Técnico / Instituto de Sistemas e Robótica
IST - Torre Norte, Av. Rovisco Pais 1,
1049-001 Lisboa, Portugal
{jseq,pal,mir,jjss}@isr.ist.utl.pt

ABSTRACT

This paper focuses on two approaches to a general functional architecture, currently being tested at the Institute for Systems and Robotics, for the control of robot teams acting in cooperation with applications to planetary exploration and satellite inspection or maintenance. This paper presents an overview of both architectures and highlights the main differences between them. Results of simulations regarding a formation control experiment are presented in the paper.

1 INTRODUCTION

Space offers many exciting challenges for robot applications. Hostile environments and complex tasks often require complex robots with a high degree of autonomy.

Multi-robot systems have long been seen as a practical and economical way to perform complex tasks. An economical argument has been used to justify the use of multiple, low cost, robots in applications such as space exploration, [Brooks et al., 1990]. A practical argument has been that intelligence can be achieved through the competition/coordination among a set of basis behaviors/roles without using complex control architectures. Similarly to many biological systems, the cooperation among robots with low-complexity leads to the emergence of complex group behaviors, resembling a form of group intelligence, and hence, to the ability to perform complex tasks.

This paper focuses on the past and current work on Cooperative Robotics carried out at the Institute for Systems and Robotics (ISR/IST). Two approaches to a general functional architecture for cooperative robotics, with applications, with suitable modifications, to space environments, including planetary exploration, are presented. The first is a task oriented architecture where each robot in the team has a specific role, assigned on-line based on the target task and an evaluation of the world state. The second approach considers the nature of robot motion in a behavioral context to obtain two fundamental operators through which the dynamics of the architecture is defined.

The paper is organized as follows. Section 2 describes the basic concepts of this work regarding functional architectures for multi-robot teams. Section 3 presents simulation results obtained with the described formation control methods. Section 4 draws some conclusions and refers to potential applications of the work to space robots.

2 COOPERATIVE ROBOTICS

Robotic teams have the advantages of redundancy, robustness and incremental operative capabilities over the single robot systems. When considering multi-robot teams, there is a broad range of scientific topics involved, from control theory (and multiple related areas such as game theory and hybrid systems) to the fundamentals of functional architecture design.

A multi-robot functional architecture must take into account two main issues.

- Task assignment: given a task to be performed by the team, how to define the tasks for each of the individual robots and hence its roles.
- Formation control: given the kinematic and dynamical constraints, how to control each robot such that a path, gener-

---

*Work sponsored by the Portuguese research programme PRAXIS XXI, project COOPERA - 2/2.1/TPAR/2087/95.

ated according the assigned role, is followed, leading to the task execution.

Two approaches to a general functional architecture for cooperative multi-robot teams are under development at ISR/IST since the mid 90's. These have in mind not only the applications to real indoors robots [Lima et al., 1999, Sequeira, 1999], but envisage applications to less structured environments (e.g., rescue after catastrophes, planetary exploration) as well.

The two approaches differ in the interpretation of the relationships between task assignment and formation control. The first approach establishes team subgroups by assigning roles to team robots; formation control must therefore be handled for each of the so formed groups. The second approach establishes a particular structure for a state space from which a number of operators (that define the motion in this state space) is derived.

2.1 A TASK ORIENTED APPROACH

Whenever multiple robots cooperate to achieve a given goal, three classes of behaviors are exhibited: *individual behaviors*, displayed by each robot in the team individually, *relational behaviors*, concerning the interactions among teammates, and *organizational behaviors*, displayed by the team as a whole. Those classes can be integrated by a 3-level *agent* team architecture introduced by Drogoul and his co-workers, [Drogoull and Collinot, 1998]. Its application to a multi-*robot* team has several distinct features summarized as follows:

- The **Organizational level** maps the *world state* (which includes the team internal state) onto a specific role assignment to each robot in the team. The mapping may be static (i.e., based only on the current world state) or take into account past information and estimates of potential future states.
- The **Relational level** is where relationships among robots are established. The robots negotiate and eventually come to an agreement about team and/or individual goals. Any team member has *relational operators*, which control relations between two or more teammates. Each operator includes a pre-conditions set and, when the conditions in the set are satisfied, establishes communications with the relational operator(s) of the designated teammates, asking them to start a negotiation process which may end up in a coordinated action among the members of the so formed temporary subteam. As a result, a *relational behavior* is displayed. Role assignments made by the organizational level are temporarily modified as the result of successful inter-robot negotiations, since they are replaced by the roles required for the execution of the coordinated action until this action is over.
- The **Individual level** operators consist of *primitive tasks* and/or *composite tasks* (primitive tasks linked by logical conditions on events). The primitive tasks are defined as sense-think-act(STA)-loops, [Lima et al., 1999], a generalization of a closed loop control system which may include motor speed control, object tracking or trajectory following control loops, to name a few. The *individual behaviors* are displayed by the robots as the result of executing *individual operators*.

A sequence of primitive tasks is traversed as the logical conditions associated with the connections among them become true. The logical conditions are defined over a predicate set, which includes predicates that check variable values and predicates that check event occurrence.

A *world model* is required to provide information to the relational and organizational levels regarding the world state. Since all computation is supposed to be distributed over the team members, with no external storage available, a distributed world model representation is required, containing all the relevant information for intra and inter-agent communication and team organization, as well as raw and processed data, for primitive tasks usage. A *distributed blackboard* has been proposed to implement the world model [Lima et al., 1999]. The *distributed blackboard* implements global shared memory and event-based communication. Each robot software includes several processes which handle the most relevant functions (e.g., coordination, guidance, self-localization). The processes write, in the blackboard, actuator data, shared variables and messages to be sent to a teammate through wireless communication. Similarly, they read, from the blackboard, sensor data, shared variables and teammate messages. Concurrent specialized processes handle all actuator, sensor and communications data, keeping the blackboard updated and labeling the data with time tags. Hence, there is no direct access of the main processes to the robot devices (actuators, sensors or wireless modems), and information (either raw or processed data) is available for any process that requests it.

Key factors for the blackboard design are:

- the information distribution per robot, which should minimize the need to communicate in order to obtain information (e.g., data obtained from processing an image should be stored in the robot where the image was acquired), and
- how to communicate information that must be shared among robots — the proposed solution is to split the blackboard information in two classes (local and global variables) and broadcast global variables whenever their values are updated.

Relational behavior is fundamental in multi-robot teams. The absence of a relational level frequently leads to situations where team behavior is poor. Consider the case of two robots with similar roles, that often conflict with each other while trying to reach a given object. The key to solve this problem is to endow the team members not only with individual goals, but also make them knowledgeable of the team goals. For an individual robot of the team, work towards meeting its individual goal(s) may include temporary modification of its role to cope with the team main goal (e.g., refraining from trying to reach a common object if it is not the closest robot to the object). This distinguishes a group of non-cooperative agents whose individual goals just happen to be the same, from a group of cooperative agents which share a common aim. The latter exhibits cooperation and coordination, while in the former the individual agents compete when the resources are scarce [Jennings, 1999].

The relational operators implement a *recipe* which is commonly agreed by all the agents of a team. This recipe is embedded in the operators and may either be prescribed initially (i.e., before joint action is started) or evolve over time. One way of changing the recipe over time consists of using reinforcement learning techniques, based on a performance function which weights the reliability (i.e., the ability to meet specifications) and the cost (computational or other) of a given recipe, [Lima and Saridis, 1996]. This applies also to the on-line selection among alternative role assignments given the world state, at the Organizational level, as well as to the on-line selection among alternative STA-loops to implement the same primitive task, at the Individual level.

2.2 A MOTION ORIENTED APPROACH

The motion of a robot is subject to kinematic, dynamic, environmental and mission constraints, all of them influencing its motion, sometimes generating conflicting maneuvering. This evidence led to the starting of a research direction at ISR/IST on the fundamental structure underlying robot control. The main goal of this research is to answer the following questions.

- How should the robot motion space (e.g., the robot configuration space or the working space) be structured, given the multiple aforementioned constraints, such that robots (operating either isolated or in a team) can be controlled?
- What are the main operations in such a space?
- How can these operations be related such that a control architecture is obtained?

The framework initially developed for the control architecture points out a basis structure in terms of two basic operators (defined on a suitable space) and of a supervisor controller, [Sequeira, 1999]. This section presents preliminary results on the inclusion, on that architecture, of an inter-robot negotiation model applied to robot teams.

The state variable considered is a pair composed of an action (i.e., a particular type of local motion) and a configuration (position and orientation) that represents the initial condition of the robot when the action is applied. Each action spans a class of trajectories contained in a bounded region of the configuration space and starting in a neighbourhood of the initial configuration. From the perspective of a mission execution, any two trajectories in such a class are equivalent, in the sense that the robot achieves its goal, and hence any of them can be chosen for the robot to follow. In a sense, each of these equivalence classes represents a behavior of the robot.

Assuming that each robot in the team is equipped with an adequate set of actions, the existence of a group structure on the space of actions is a necessary condition for the controllability of each single, behavior-controlled, robot. An operator, defined on the state space, provides the mechanism to switch between any two states under a set of conditions required to the existence of the aforementioned group structure. This operator is named *state composition*. Whenever the available actions need to be adapted due to the kinematics or the environment changes, the second operator, named *state expansion*, is used, aiming at preserving the group structure, [Sequeira, 1999, Sequeira and Ribeiro, 2000]. This relationship between these two operators define a dynamics for the control architecture in the sense that all the motion in the state space is generated by its application.

To achieve cooperation, each robot in the team has to be able to exchange information, at least with some of the other robots, in order to establish some form of coalition. This exchange is obtained by extending the state space structure used by each of the robots to the team. Similarly to single robots, the team actions and configurations define the team state space, also equipped with convenient state composition and expansion operators.

A further extension of the concept of team state arises in the case of limited communication capabilities among the robots in a team. In this case, a number of subteams (which, in a sense, represent the coalitions among the robots) may be formed, each of which moves towards the execution of its task. Furthermore, it should be emphasized that a robot may participate in more than one subteam.

The state of the team or of the subteam is formed by a suitable (problem dependent) combination of the states of the member robots. The supervisor at every robot knows how to combine the states of its teammates and, hence, knows its subteam state. Each robot's supervisor controller has thus information on the state of its own robot, on the task it is pursuing and on the state of the subteam it belongs to.

Based on the events detected and on the state information, the supervisor controller at each robot determines the action to be executed and computes an intermediate goal to be used by the robot while pursuing towards the task. Whenever a robot supervisor decides for an action, a negotiation process starts. The robot broadcasts a request to the team indicating that it needs to change state. This request is processed by the members in the same subteam (i.e., those who detected the request) each of which tries to foresee the effect this change may have in the execution of the current task by the subteam. If no robot in the subteam refuses the request, the state change is allowed and the execution of the task continues.

3 SIMULATION RESULTS

Simulation results of two missions, using different robot kinematics, are presented in this section, both using the architecture in 2.2. In both missions it is required that the team adopts a flocking behavior.

The same sets of basis actions are used in the two experiments. Each single robot is able to (1) *stop*, (2) *go to task*, and (3) *avoid another robot*. The team is able to (1) *stop* and (2) *go to task*, as no environment disturbances are considered.

In the presented experiments, the teams evolve in a 2D plane without being disturbed by any obstacle (other than themselves). Both missions are composed by four tasks, each one aiming at reaching a goal, whose location is marked by the symbol □. Figures 1 and 4 show the trajectories of each of the robots in each experiment. Figures 2 and 5 show the positions of the tasks (□) and of the intermediate goals (△) generated along the mission execution. The wide circles indicate the boundaries of the regions where a negotiation took place. Figures 3 and 6 show the trajectories of the center of mass for each subteam formed along the mission execution. The position and orientation of each robot are represented, respectively, by the symbols ○ and −. The robots are assumed to have no physical dimensions.

Each time a robot traverses one of the wide circles, visible in Figures 2 and 5, generated by itself at sparse time instants, a negotiation with the other robots is started to obtain the autorization to continue the execution of the mission. It should be emphasized that none of the figures illustrates the temporal dependencies among the motion of each of the robots. From the complexity of the interactions, namely the avoidance of obstacles, these would be difficult to represent graphically.

The first experiment (Figures 1 to 3) considers a team of four identical holonomic robots flocking along a trapezoidal pattern defined by the tasks. The high density of the plots near the task positions indicates intense maneuvering by the team. The oscillations observed in the regions between the tasks are mainly due to obstacle avoidance interactions. These cause the team action *go to task* to generate team intermediate goal positions slightly oscillating around the straight line path defined by two consecutive tasks. Figure 3 shows the trajectories for the center of mass of the subteams formed along the mission. The discontinuities in the the trajectories indicate the points where subteams were created and destroyed. No collisions occured during the mission.

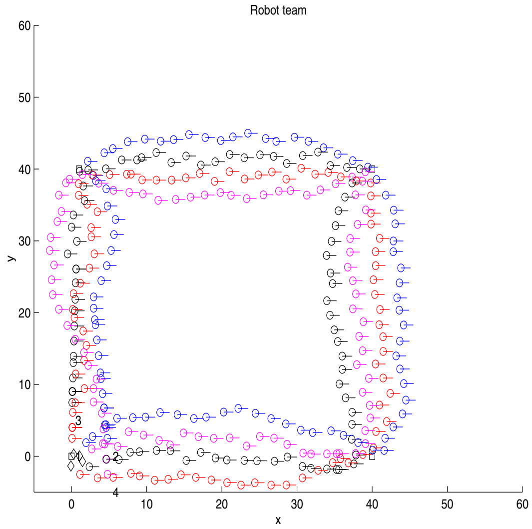

Figure 1: A team of 4 holonomic robots

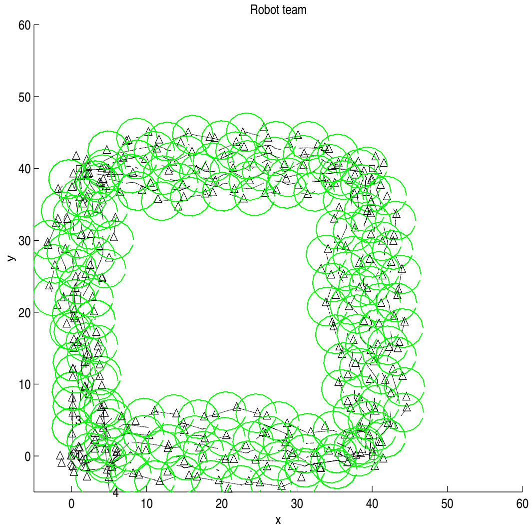

Figure 2: Intermediate goals and negociation boundaries

In the second experiment (Figures 4 to 6) a team of three identical unicycle robots is flocking along a saw-tooth pattern for the tasks. The oscillation observed in the regions between the task locations is, again, due to obstacle avoidance interactions. Note that these are much more pronounced than those in the previous experiment, consequence of the different robot kinematics. Figure 6 shows the trajectories of the center of mass of the team/subteams in the regions between tasks. As in the previous experiment they exhibit numerous discontinuities as the result of the creation and destruction of subteams. No collisions occured during the mission.

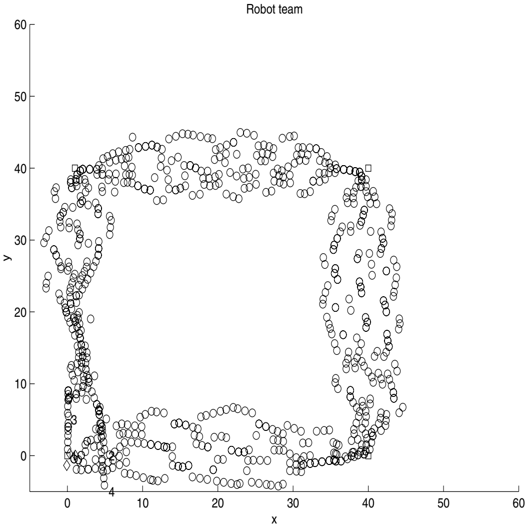

Figure 3: Trajectory of the centers of mass of the team/subteams

The number of wide circles in Figures 2 and 5 illustrates the intensive negotiations occuring along the missions. Although not visible from the plots, the negotiations occur when a robot tries to move away from the rest of the team. Furthermore, even in such simple experiments, the dynamics of creation and destruction of the subteams points out the complexity of the general cooperation problem in robotics.

4 CONCLUSIONS

Two approaches to general functional architectures under development at ISR/IST have been presented in this paper. Both architectures are being applied to real indoors cooperative robotic teams and applications to outdoors robots for rescue after urban catastrophes, such as earthquakes, or planetary exploration, are envisaged.

The flocking behavior considered in this paper can be used in relevant applications. Among these are the surveillance/inspection (aerial or terrestrial) using non rigid team formations and rescue operations using different robots, each providing the team with specific functionalities in a coordinated form.

Another envisaged application to Space Robotics of behavior-based architectures for multi-robot teams concerns satellite formations. In recent years, the usage of for-

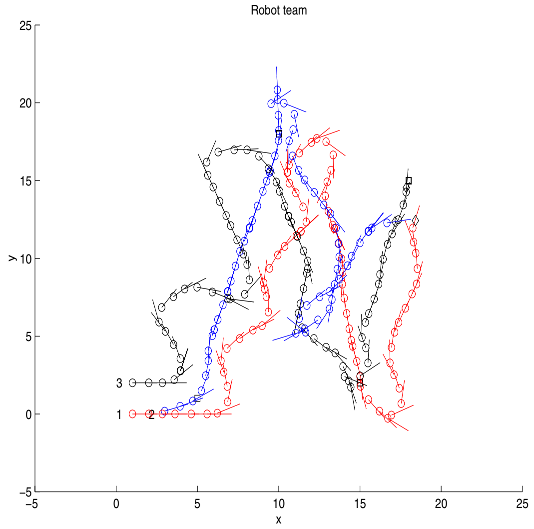

Figure 4: A team of 3 unicycle robots

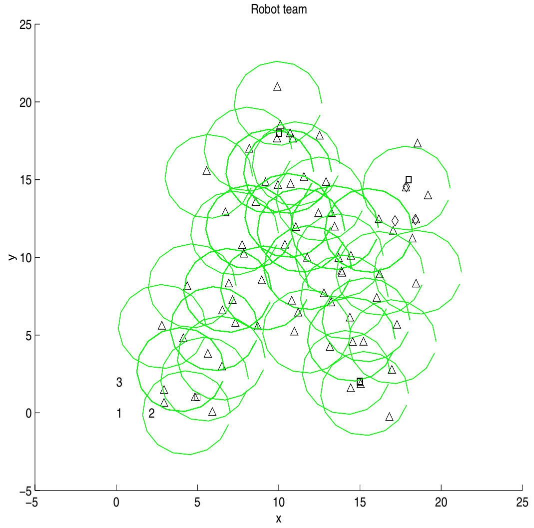

Figure 5: Intermediate goals and negociation boundaries

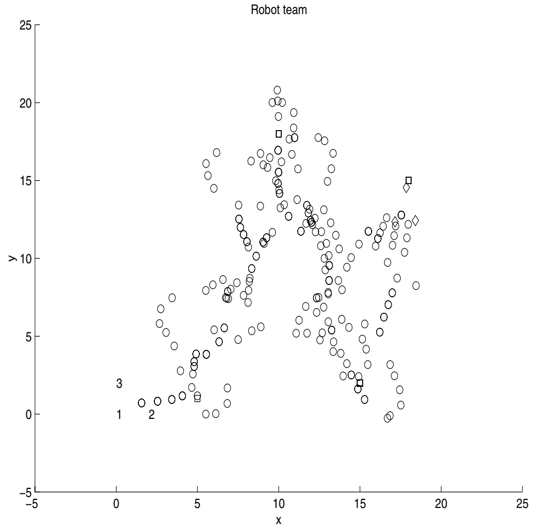

Figure 6: Trajectory of the center of mass of the team/subteams

mations of several micro-satellites to implement space telescopes and space antenna arrays has attracted increasing attention, [Robertson et al., 1999]. Given a specification (e.g., the desired telescope heading), interesting control and coordination problems arise, such as:

- what is the desired configuration for each satellite of the formation that will achieve the desired telescope or heading;
- how to move each satellite of the formation to achieve the new desired configuration in minimum time and without colliding with the other satellites;
- how to minimize the energy spent by the formation, by a suitable configuration distribution among the formation members - notice that satellites in the outer formation zone will usually spend more energy that their inner team mates. Therefore, depending on the requested manoeuvres, the choice of which satellite(s) will move will depend on their past trajectory and relative formation position.

Similar concepts apply to free-flying robot teams designed for inspection and maintenance in space of satellites and space stations (e.g., parts assemblage, screw fastening, large object manipulation).

ISR/IST has gained considerable experience in the past three years on micro-satellite attitude control and determination, [Marques et al., 2000, Clements et al., 2000], as well as on methods for position tracking along 3D trajectories for non-holonomic robots which are part of a "follow-the-leader" cooperative scheme [Tabuada and Lima, 2000], and formation controllability analysis, [Tabuada et al., 2001]. Current work includes the study of team controllability under the negotiation processes that occur under the motion oriented approach. These should be a valuable asset for the application of cooperative robotics methodologies to Space Robotics.

5 ACKNOWLEDGEMENTS

The authors would like to acknowledge Luís Custódio, Rodrigo Ventura, Paulo Tabuada and Luís Toscano the important contribution to some of the concepts expressed in this paper.

6 REFERENCES

[Brooks et al., 1990] Brooks, R., Maes, P., Matarić, M., and More, G. (1990). Lunar Base Construction Robots. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems, IROS'90.*

[Clements et al., 2000] Clements, R., Tavares, P., and P., L. (2000). Small Satellite Attitude Control Based on a Kalman Filter. In *Proceedings of the IEEE International Symposium on Intelligent Control 2000.* Patras, Greece.

[Drogoull and Collinot, 1998] Drogoull, A. and Collinot, A. (1998). Autonomous Agents and Multi-Agent Systems. In *Applying and Agent-Oriented Methodology to the Design of Artificial Organizations: A Case Study in Robotic Soccer.* Kluwer Academic.

[Jennings, 1999] Jennings, N. (1999). Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems Using Joint Intentions. *Artificial Intelligence*, (75):195–240.

[Lima and Saridis, 1996] Lima, P. and Saridis, G. (1996). *Design of Intelligent Control Systems Based on Hierarchical Stochastic Automata.* World Scientific.

[Lima et al., 1999] Lima, P., Ventura, R., Aparício, P., and Custódio, L. (1999). A Functional Architecture for a Team of Fully Autonomous Cooperative Robots. In *Proceedings of the RoboCup Workshop of IJCAI 99.* Stockholm, Sweden (also to appear in M. Veloso and H. Kitano (eds), RoboCup-99: Robot Soccer World Cup III, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2000.

[Marques et al., 2000] Marques, S., Clements, R., and Lima, P. (2000). Comparison of Small Satellite Attitude Determination Methods. In *Proceedings of the 2000 AIAA Conference on Navigation, Guidance and Control.* Colorado, USA.

[Robertson et al., 1999] Robertson, A., Inalhan, G., and How, J. (1999). Spacecraft Formation Flying Control Design for the Orion Mission. In *Proceedings of the 1999 AIAA Conference on Navigation, Guidance and Control.* Portland, OR.

[Sequeira, 1999] Sequeira, J. (1999). *Cooperation Among Robots: A Behavioural Approach Supported On Group Theory.* PhD thesis, Instituto Superior Técnico, Portugal. Departamento de Engenharia Electrotécnica e de Computadores.

[Sequeira and Ribeiro, 2000] Sequeira, J. and Ribeiro, M. I. (2000). Behaviour-Based Cooperation Between Two Mobile Manipulators. *Journal of Integrated Computer-Aided Engineering*, 7(3):193–215.

[Tabuada and Lima, 2000] Tabuada, P. and Lima, P. (2000). Position Tracking for Underactuated Rigid Bodies on SE(3). Technical Report RT-401-2000, Instituto de Sistemas e Robótica.

[Tabuada et al., 2001] Tabuada, P., Pappas, G., and Lima, P. (2001). Feasible Formations of Multi-Agent Systems. Submitted to the *American Control Conference 2001.*