Appeared in *Proceedings of the 4th International Conference on Information Systems Analysis and Synthesis (ISAS '98)*, vol. 3, pages 89-94.

Distributed Control of Multi-Robot Teams: Cooperative Baton Passing Task

Lynne E. Parker
Center for Engineering Science Advanced Research, Oak Ridge National Laboratory
P. O. Box 2008, Oak Ridge, TN 37831-6355 USA

ABSTRACT

This research addresses the problem of achieving fault tolerant cooperation within small- to medium-sized teams of heterogeneous mobile robots. We describe a novel behavior-based, fully distributed architecture, called ALLIANCE, that utilizes adaptive action selection to achieve fault tolerant cooperative control. The robots in this architecture possess a variety of high-level functions that they can perform during a mission, and must at all times select an appropriate action based on the requirements of the mission, the activities of other robots, the current environmental conditions, and their own internal states. Since such cooperative teams often work in dynamic and unpredictable environments, the software architecture allows the team members to respond robustly and reliably to unexpected environmental changes and modifications in the robot team that may occur due to mechanical failure, the learning of new skills, or the addition or removal of robots from the team by human intervention. After presenting ALLIANCE, we describe our implementation of this architecture on a team of physical mobile robots performing a cooperative baton passing task. These experiments illustrate the ability of ALLIANCE to achieve adaptive, fault-tolerant cooperative control amidst dynamic changes during the task.

Keywords: Multi-robot teams, cooperative control, ALLIANCE, distributed robotics

1. INTRODUCTION

Achieving cooperative robotics is desirable for a number of reasons. First, many robotic applications are inherently distributed in space, time, or functionality, thus requiring a distributed solution. Second, it is quite possible that many applications could be solved much more quickly if the mission could be divided across a number of robots operating in parallel. Third, by duplicating capabilities across robot team members, one has the potential of increasing the robustness and reliability of the automated solution through redundancy. Finally, it may actually be much cheaper and more practical in many applications to build a number of less capable robots that can work together at a mission, rather than trying to build one robot which can perform the entire mission with adequate reliability.

Achieving cooperative robotics, however, is quite challenging. Many issues must be addressed in order to develop a working cooperative team, such as action selection, coherence, conflict resolution, and communication. Furthermore, these cooperative teams often work in dynamic and unpredictable environments, requiring the robot team members to respond robustly, reliably, and adaptively to unexpected environmental changes, failures in the inter-robot communication system, and modifications in the robot team that may occur due to mechanical failure, the learning of new skills, or the addition or removal of robots from the team by human intervention.

Previous research in heterogeneous mobile robot cooperation includes: [1] which proposes a three-layered control architecture that includes a planner level, a control level, and a functional level; [2] which describes an architecture that includes a task planner, a task allocator, a motion planner, and an execution monitor; [3] which describes an architecture called ACTRESS that utilizes a negotiation framework to allow robots to recruit help when needed; and [4], which uses a hierarchical division of authority to address the problem of cooperative fire-fighting. However, these approaches deal primarily with the task selection problem and largely ignore the difficult issues for physical robot teams, such as robot failure, communication noise, and dynamic environments. In contrast, our research emphasizes the need for fault tolerant and adaptive cooperative control as a principal characteristic of the cooperative control architecture.

This paper describes an architecture that we have built for heterogeneous mobile robot control that emphasizes fault tolerant, adaptive cooperation. This architecture, called ALLIANCE, is designed for small- to medium-sized teams of robots. We then describe the results of implementing this architecture on a team of four mobile robots performing a cooperative baton passing task. This is implementation is significant because it is the first time that a distributed control approach to cooperative baton passing has been successfully implemented on a physical robot team.

2. APPROACH

The cooperative control mechanism is based upon the ALLIANCE framework, which enables fault tolerant, adaptive multi-robot action selection. Shown in Figure 1, this architecture is a hybrid approach to robotic control that incorporates a distributed, real-time reasoning system utilizing behavioral motivations above a layer of low-level behavior-based control mechanisms. This architecture for cooperative control utilizes no centralized control; instead, it enables each individual robot to select its current actions based upon its own capabilities, the capabilities of its teammates, a previous history of interaction with particular team members, the current state of the environment, and the robot's current sensory readings. ALLIANCE does not require any use of negotiation among robots, but rather relies upon broadcast messages from robots to announce their current activities. The ALLIANCE approach to communication and action selection results in multi-robot cooperation that gracefully degrades and/or adapts to real-world problems, such as robot failures, changes in the team mission, changes in the robot team, or failures or noise in the communication system.

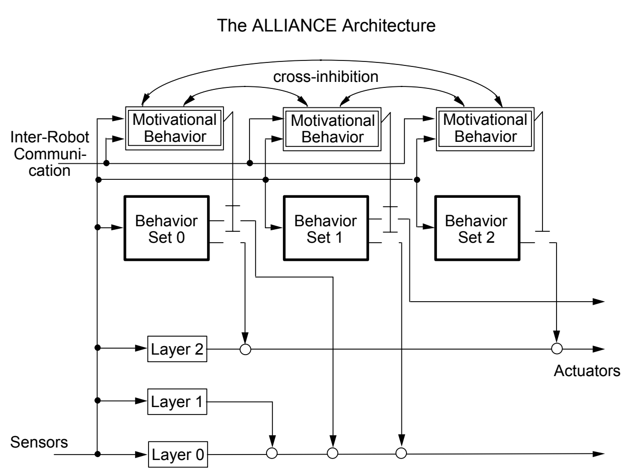

Figure 1. The ALLIANCE architecture.

Unlike typical behavior-based approaches, ALLIANCE delineates several "behavior sets" that are either active as a group or hibernating. Figure 1 illustrates three such behavior sets. The *j*th behavior set, a_{ij}, of a robot r_i corresponds to those levels of competence required to perform some high-level task-achieving function, such as finding a toxic waste spill, moving a toxic waste spill, or reporting the progress of the robot team to a human monitor. When a robot activates a behavior set, we say that it has selected the task corresponding to that behavior set. Since different robots may have different ways of performing the same task, and will therefore activate different behavior sets to perform that task, we define the function $h_i(a_{ij})$, for all robots, r_i, on the team, to refer to the task that robot r_i is working on when it activates its *j*th behavior set. Because of the alternative goals that may be pursued by the robots, the robots must have some means of selecting the appropriate behavior set to activate. Thus, controlling the activation of each of these behavior sets is a *motivational behavior*. Due to conflicting goals, only one behavior set per robot should be active at any point in time. This restriction is implemented via cross-inhibition of motivational behaviors, represented by the arcs at the top of Figure 1 in which the activation of one behavior set suppresses the activation of all other behavior sets. However, other lower-level competences such as collision avoidance may be continually active regardless of the high-level goal the robot is currently pursuing. Examples of this type of continually active competence are shown in Figure 1 as layer 0, layer 1, and layer 2.

The primary mechanism for achieving adaptive action selection in this architecture is the *motivational behavior*. At all times during the mission, each motivational behavior receives input from a number of sources, including sensory feedback, inter-robot communication, inhibitory feedback from other active behaviors, and internal motivations called *robot impatience* and *robot acquiescence*. The output of a motivational behavior is the activation level of its corresponding behavior set, represented as a non-negative number. When this activation level exceeds a given threshold, the corresponding behavior set becomes active.

Intuitively, a motivational behavior works as follows. Robot r_i's motivation to activate any given behavior set a_{ij} is initialized to 0. Then, over time, robot r_i's motivation $m_{ij}(t)$ to activate behavior set a_{ij} increases at a fast rate as long as the task corresponding to that behavior set (i.e. $h_i(a_{ij})$) is not being accomplished, as determined from sensory feedback. However, we want the robots to be responsive to the actions of other robots, adapting their task selection to the activities of team members. Thus, if a robot r_i is aware that another robot r_k is working on task $h_i(a_{ij})$ then r_i should be satisfied for some period of time that the task is going to be accomplished even without its own participation, and thus go on to some other applicable action. Its motivation to activate behavior set a_{ij} still increases, but at a slower rate. This characteristic prevents robots from replicating each other's actions and thus wasting needless energy. Of course, detecting and interpreting the actions of other robots (often called *action recognition*) is not a trivial problem, and often requires perceptual abilities that are not yet possible with current sensing technology. As it stands today, the sensory capabilities of even the lower animals far exceed present robotic capabilities. Thus, to enhance the robots' perceptual abilities, ALLIANCE utilizes a simple form of broadcast communication to allow robots to inform other team members of their current activities, rather than relying totally on sensory capabilities. At some pre-specified rate, each robot r_i broadcasts a statement of its current action, which other robots may listen to or ignore as they wish. No two-way conversations are employed in this architecture.

Each robot is designed to be somewhat impatient, however, in that robot r_i is only willing for a certain period of time to allow the communicated messages of another robot to affect its own motivation to activate a given behavior set. Continued sensory feedback indicating that a task is not getting accomplished thus overrides the statements of another robot that it is performing that task. This characteristic allows robots to adapt to failures of other robots, causing them to ignore the activities of a robot that is not successfully completing its task.

A complementary characteristic in these robots is that of acquiescence. Just as the impatience characteristic reflects the fact that other robots may fail, the acquiescence characteristic indicates the recognition that a robot itself may fail. This feature operates as follows. As a robot r_i performs a task, its willingness to give up that task increases over time as long as the sensory feedback indicates the task is not being accomplished. As soon as some other robot r_k indicates it has begun that same task and r_i feels it (i.e. r_i) has attempted the task for an adequate period of time, the unsuccessful robot r_i gives up its task in an attempt to find an action at which it is more productive. Additionally, even if another robot r_k has not taken over the task, robot r_i may give up its task anyway if it is not completed in an acceptable period of time. This allows r_i the possibility of working on another task that may prove to be more productive rather than becoming stuck performing the unproductive task forever. With this acquiescence characteristic, therefore, a robot is able to adapt its actions to its own failures.

The behavior-based design of the motivational behaviors also allows the robots to adapt to unexpected environmental changes which alter the sensory feedback. The need for additional tasks can suddenly occur, requiring the robots to perform additional work, or existing environmental conditions can disappear and thus relieve the robots of certain tasks. In either case, the motivations fluidly adapt to these situations, causing robots to respond appropriately to the current environmental circumstances.

3. IMPLEMENTATION

We have successfully implemented our ALLIANCE approach in a number of laboratory implementations, including "mock" hazardous waste cleanup [5], bounding overwatch [6], janitorial service [6], box pushing [7], and cooperative object tracking [8]. Here, we report our implementation on a team of four Nomad mobile robots performing a cooperative baton passing task.

The robot team

Our robot team consists of four Nomad robots. These robots are wheeled platforms approximately 80 cm (32 inches) high and 44 cm (17.6 inches) wide. They are equipped with a variety of sensors, including a ring of 16 sonar sensors toward the top of the robot, a ring of 16 infrared sensors about 2/3 of the way down the robot, odometric sensors, a flux-gate compass, and an indoor laser-based 2D global positioning system. The robots also have a laser ranging system and have been recently equipped with a color vision system, although these last two sensors were not used in the experiments reported here.

Each robot has a simple manipulator with a 1D pincher-style gripper that allows it to grasp certain types of objects. An infrared break-beam sensor within the gripper allows the robot to detect when an object is inside its gripper. The manipulator is a 1D telescoping device that allows the robot to reach to the floor or up to a height of about 120 cm (4 feet).

The robots are equipped with a radio ethernet system that enables the robots to talk with each other. Our implementation of the communication system is described below. A voice synthesizer is also available onboard the robots to allow them to communicate messages to human observers.

Task description

Figure 2 is a diagram of the typical layout of the cooperative baton passing task, viewed from above. The cooperative baton passing task requires a team of robots to move several batons from a starting location to a goal location. In this case, the "baton" is a stiff roll of paper approximately 60 cm (2 ft) long and 4 cm (1.6 inches) in diameter. The starting and goal locations are separated by a several physical barriers that impede a robot from moving directly from the start location to the goal location. These barriers are wooden fences approximately 37.5 cm (15 inches) high, which can be detected by the robots with their infrared sensors. The barriers effectively divide the working area into "bays" of operations. Ideally, there would be one or more robots working in each bay. Since the robots cannot easily reach both the start and goal locations, they must pass the batons over the barriers in a relay fashion to move each baton from the starting location to the goal location. In our implementation, the goal location is a bin, into which the last robot drops the baton.

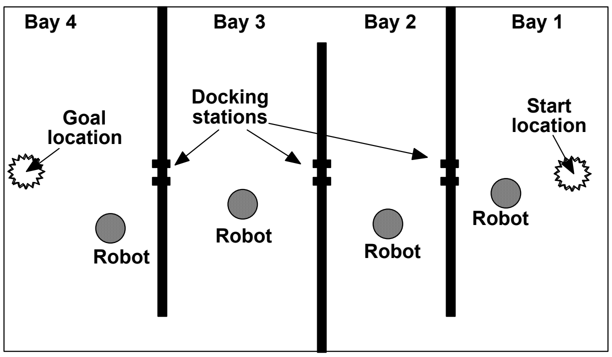

Figure 2. Diagrammatic layout of cooperative baton passing task.

We note, however, that the barriers do not completely block the path, since an opening is available at one end of each barrier to allow robots a path around the barriers if needed. However, the path around the barriers is longer and thus more time-consuming compared to passing the batons over the barriers. Thus, as long as more than one robot is operational, it is advantageous to pass the batons across the barriers to other robots rather than a single robot trying to move the baton the entire way on its own.

To ease the robot sensing requirements of the task, docking stations are defined at which the baton transfer should take place across each barrier. In future implementations, visual sensing will allow robots to more easily detect the location of other robots, thus enabling the elimination of these docking stations.

At the beginning of the task, the robots are informed of the layout of the room, the location of the start and goal locations, and the location of the docking stations. A human is stationed at the start location to hand the batons (one at a time) to the robot in the first bay.

Inter-Robot Communication

For this task to operate successfully, robots need to know several items of information regarding the state of their robot teammates. Specifically, robots need to know:

- The bay(s) in which other robots are operational;
- Whether another robot is ready to pass a baton; and,
- The docking station at which a baton is awaiting transfer.

In addition, during an actual transfer of a baton from robot 1 to robot 2, the two robots must know:

- When robot 2 has grasped the baton; and
- When robot 1 has let go of the baton.

This information is made available to robot team members through the radio ethernet system onboard each robot. Each robot broadcasts messages to its teammates during the mission using a pre-defined message language and syntax. The message format is:

sender-id, message-type, parameter-1, parameter-2, …

Sender-id identifies the sender of the message, where each robot has a unique ID number. The human controller also has a unique ID, and can send messages to the robot team when needed.

Message-type indicates the purpose of the message, and may be followed by one or more parameter values that provide further information. Possible combinations for this application are:

- "D" *dock-number height*: object available at docking station with ID *dock-number*, with baton positioned for grasping at height *height*
- "G" *dock-number*: object grabbed at docking station *dock-number*
- "R" *dock-number*: object released at docking station *dock-number*
- "A" *bay-number*: robot performing operations in bay with ID of *bay-number*

Note that when a robot is broadcasting that a baton is ready to be passed at a certain dock, the robot also provides a height at which the baton can be grasped. Since robots cannot re-adjust the grasp of the baton once it is inside the gripper, and they cannot passively sense the position of the baton within another robot's gripper (since vision has not yet been implemented), the robots alternate grasping the baton at high and low locations along the baton. This prevents the robots from running out of places to grasp the baton as it moves across multiple barriers.

Determination of Passing Batons vs. Circumnavigating Barriers

Before a robot can pass a baton from its current bay to a robot in the next bay, it must know that a robot is currently operational inside the next bay. This information is obtained through the periodic announcements of each robot of messages of the type “A” described earlier, which indicate the bay in which a robot is currently operating. If a robot has not heard a recent announcement by another robot that it is operating in a particular bay, then the first robot will assume that a baton transfer is not possible, and will circumnavigate the current barrier to move the baton along toward the goal location. Once inside the next bay, the robot will have to make a similar decision to determine whether to pass the baton across the next barrier or to circumnavigate the barrier again.

Any additional robots that are available on the team (i.e. over and above the number of bays in the task) will also use this broadcast message to affect its own impatience levels. Thus, if no robot is currently operating within a certain bay, then the additional robot will become impatient to perform that operation, and will move to the appropriate bay to take over operations there. This allows the robot team to successfully reallocate its tasks and respond to changes in the robot team to ensure that the task is performed as efficiently as possible.

Sensing for Baton Transfer

From a sensing perspective, the most difficult aspect of this task is aligning the robots appropriately for the transfer to occur. Because of the limited sensing available for our experiments, the robot transfer can be likened to the difficulty of two blind people with thick gloves trying to touch each other’s noses without using their voices to localize the other’s position. To make the task feasible, we enabled the robots to use their 2D indoor global positioning system, their infrared sensors, plus knowledge of the location of the docking stations to align themselves at a certain distance and orientation to the given docking station. Once a robot, say *robot-1*, with a baton aligns itself for the baton transfer, it announces the ID of the docking station to its teammates.

When another robot, say *robot-2,* is available to pick up the baton in the bay opposite the docking station, it approaches the docking station from the other side. When *robot-2* is appropriately oriented to the docking station, it begins advancing towards the *robot-1* until it senses the baton within its gripper with the break-beam infrared sensor in its gripper. When the baton is inside the gripper, *robot-2* announces that it has grasped the object (using the “G” message type described earlier). *Robot-1* then knows that it is safe to let go of the baton. Once *robot-1* has released the baton, it announces that fact using the “R” message type. *Robot-2* is then free to transfer the baton across its bay to the next robot.

When a robot in the last bay obtains a baton, it carries it to the goal location and drops it inside the bin.

4. RESULTS

We have run a large number of experimental trials of this task using the implementation described above. We have varied the robot team size from two to four, and have videotaped a number of runs. A typical example of a four-robot experiment is shown in Figures 3 through 5. Figure 3 shows the beginning of the task, when the human gives the first baton to one of the robots on the team. Figure 4 shows one of the robots transferring the baton across the barrier to an awaiting second robot. Figure 5 shows all four robots in operation as they perform the baton-moving task, with several batons in process across the barriers.

In these experiments, the robots successfully performed their task for the large majority of the test runs. The primary failure mode occurred due to sensory errors in aligning the robots for the robot-to-robot baton transfer. At times, the robots would be misaligned side-to-side, or would be either too close or too far from each other to make a successful transfer. Nearly all of these errors could be corrected with a vision system that enables compensation for the limited sensing available in the experiments reported here.

Figure 3. Human gives baton to first robot on team.

Figure 4. Robot transfers baton over barrier to another robot team member.

Figure 5. Four robots performing the cooperative baton passing task.

5. CONCLUSIONS

We have described our ALLIANCE architecture and its use in the implementation of a cooperative baton-passing task. This implementation shows the breadth of tasks for which ALLIANCE is useful, and illustrates how ALLIANCE can be used for more tightly-coupled tasks involving shared object manipulation.

Current research is aimed toward multi-robot learning and the automatic design and generation of cooperative robot behaviors.

6. REFERENCES

[1] F. Noreils, “Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment.,” *The International Journal of Robotics Research*, 12, pp. 79-98, 1993.

[2] W. C. P. Caloud, J-C. Latombe, C. Le Pape, and M. Yim, “Indoor automation with many mobile robots,” Proceedings of IEEE International Workshop on Intelligent Robots and Systems, Tsuchiura, Japan, pp. 67-72, 1990.

[3] K. O. H. Asama, A. Matsumoto, Y. Ishida, and I. Endo, “Development of task assignment system using communication for multiple autonomous robots,” *Journal of Robotics and Mechatronics*, 4, pp. 122-127, 1992.

[4] M. G. P. Cohen, D. Hart, and A. Howe, “Real-time problem solving in the Phoenix environment,” University of Massachusetts, Amherst, COINS Technical Report 90-28, 1990.

[5] L. E. Parker, “ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation,” *IEEE Transactions on Robotics and Automation*, 14, pp. 220-240, 1998.

[6] L. E. Parker, “On the Design of Behavior-Based Multi-Robot Teams,” *Journal of Advanced Robotics*, 10, pp. 547-578, 1996.

[7] L. E. Parker, “ALLIANCE: An Architecture for Fault Tolerant, Cooperative Control of Heterogeneous Mobile Robots,” Proceedings of IEEE/RSJ/GI International Conference on Intelligent robots and Systems (IROS '94), Munich, Germany, pp. 776-783, 1994.

[8] L. E. Parker, “Cooperative Motion Control of Multi-Target Observation,” Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '97), Grenoble, France, pp. 1591-1598, 1997.