Multiagent Teleautonomous Behavioral Control*

Khaled S. Ali and Ronald C. Arkin
Mobile Robot Laboratory
College of Computing
Georgia Institute of Technology
Atlanta, GA, USA 30332-0280
e-mail: kali@cc.gatech.edu, arkin@cc.gatech.edu

Abstract

Multiagent schema-based reactive robotic systems are complemented with the addition of a new behavior controlled by a human operator. This enables the whole society to be affected as a group rather than forcing the operator to control each agent individually. The operator is viewed by the reactive control system as another behavior exerting his/her influence on the society as a whole. The operator can also control the overall personality of the robot group. Simulation results are presented for foraging, vacuuming, and herding tasks. Results on real robots are presented for maneuvering robots out of a box canyon and squeezing robots through a small space. Teleautonomous operation of multiagent reactive systems was demonstrated to be significantly useful for some tasks, less so for others.

1 Introduction

Reactive multiagent robotic societies are potentially useful for a wide-range of tasks. This includes operations such as foraging and grazing (e.g., [1, 12, 7]) which have applicability in service (vacuuming and cleaning), industrial (assembly) and military (convoy and scouting) scenarios.

Although some promising results have been achieved in these systems to date [8], purely reactive systems can still benefit from human intervention. Many purely reactive systems are myopic in their approach: they sacrifice global knowledge for rapid local interaction. Global information can be useful and it is in this capacity that a human operator can interact with a multiagent control system.

A related problem in teleoperation is that the operator is potentially overwhelmed by the large amount of data required to control a multiagent system in a dynamic environment. This phenomenon is referred to as cognitive overload. The approach described in this paper provides a mechanism to significantly reduce the human operator's cognitive and perceptual load by allowing the reactive system to deal with each robot's local control concerns. Two principal mechanisms to achieve this are by allowing the operator to act either as a constituent behavior of the society or to allow him/her to supervise the societal behavioral sets and gains, acting only as needed based upon observable progress towards task completion.

In this research, the operator is allowed to control whole societies of agents; not one robot at a time, but rather controlling global behavior for the entire multiagent system. This is a straightforward extension of our work in both multiagent robotic systems [1] and teleautonomy [2]. The end product is a simple way for a commander to control large numbers of constituent elements without concern for low-level details (which each of the agents is capable of handling by themselves). In essence, the human operator is concerned with global social *strategies* for task completion, and is far less involved with the specific behavioral *tactics* used by any individual agent.

*Portions of this paper appeared in the Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB94)

2 Single agent teleautonomous control

Our previous results [2] in the integration of reactive and telerobotic control in the context of single agents provide the basis for our extension of this concept into multiagent societies. In this earlier work we have shown that a human operator can interact with a reactive robot in at least two different ways:

- **Operator as a schema**: Here the human acts as an additional behavior in the already existing collection of behaviors that are active within the robot. Using a schema-based methodology [3], each active behavior contributes a vector that is related to the agent's intentions - such as to get to a particular object, not crash into something, etc. The operator's intentions are introduced at the same level - as another schema contributing forces in the same manner as all the other behaviors do.
- **Operator as a behavioral supervisor**: In this case, the human changes the behavioral settings of the robot as it moves through the world, essentially changing its "personality". For example, the robot can become more aggressive by increasing its attraction towards a desirable object or decreasing its repulsion from obstacles.

In schema-based reactive control [3], each active behavior (schema) provides its own reaction to the environment by creating a vector response to a specific perceptual stimulus. The entire set of vector outputs created by all active schemas is summed and normalized and then transmitted to the robot for execution. No arbitration is involved, rather a blending of all active concurrent behaviors occurs. The system at this level is completely reactive, not retaining knowledge of the world or the agent's past performance.

3 Multiagent Teleautonomous Control

Our laboratory is conducting extensive research in multiagent robotic systems [1, 5, 4] both in simulation and on our 3 Denning Mobile Robots. Robotic systems are specified as a finite state acceptor that specifies the behavioral (schema) assemblages [9, 10] and the transitions between them. An example state machine for a foraging task appears in Figure 1. In this figure there exist three distinct high-level behavioral states for each agent:

- *Forage* - which consists of a high gain and long persistence **noise** schema that is used to produce wandering while having moderate inter-robot repulsion to produce dispersion coupled with significant obstacle repulsion (**avoid-static-obstacle** schemas).
- *Acquire* - which consists of using a **move-to-goal** schema to move towards a detected or reported attractor (depending on the communication strategy used [5]) with a reduced inter-robot repulsion to allow for multi-robot convergence on attractors and continued obstacle avoidance (again provided by the **avoid-static-obstacle** schema). A small amount of **noise** is still injected into the system to facilitate navigation [3].
- *Deliver* - which occurs after acquisition of the attractor and results in delivery of the object back to homebase by one or more agents. The same behaviors are used as in the *acquire* state with the goal location now being the homebase.

Space prevents a full discussion of the mechanisms for reactive multiagent control. The interested reader is referred to [1, 5] for more information.

3.1 Implementation

We have developed a multiagent teleautonomy system called TELOP. In TELOP, teleoperation is implemented both as an additional schema in the system (the operator as a schema approach) and as a method for modifying the behavioral parameters (the operator as a behavioral supervisor approach).

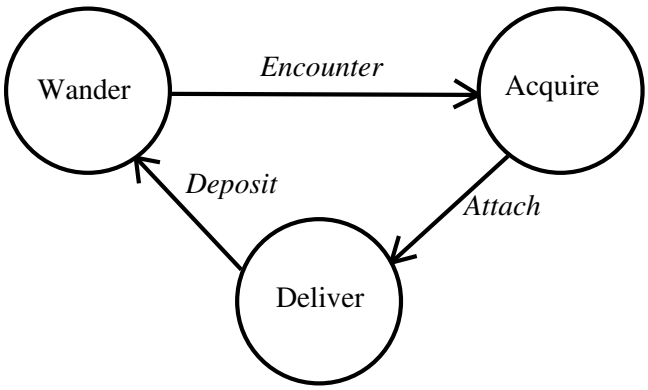


Figure 1: Behavioral States for Foraging Task.

We will discuss the operator as a schema approach first. Based on the instructions of a human agent, the **teleautonomy** schema contributes a vector in the same way as do the other schemas, such as **move-to-goal** or **avoid-static-obstacle**. Unlike the other schemas, however, which produce different vectors for each robot, the **teleautonomy** schema produces the same output for each of the robots in the team. Thus, if the human agent tells the robots to go north, then all the robots receive the same vector. The output produced by the **teleautonomy** schema is summed with all of the vectors produced by the other active schemas in each agent to produce a combined vector which determines the overall direction and rate of travel of the robot. In this way, the robots use environmental knowledge provided by the human agent in conjunction with their other goals, such as not to collide with obstacles or each other, rather than having the operator's goals completely override the robots' other behaviors.

The human agent has control over both the direction and magnitude of the vector produced by the **teleautonomy** schema. An on-screen "joystick", as shown in Figure 2, is used to input the desired direction and magnitude.

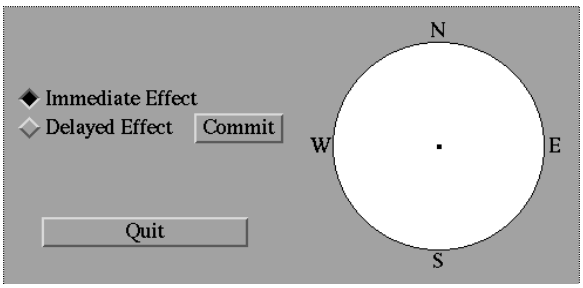


Figure 2: Main window with on-screen joystick.

When acting as a behavioral supervisor, the human operator adjusts the behavioral parameters of the society. Each of the behaviors has one or more parameters associated with it, such that the values determine exactly how the robots' will react. For instance, one parameter of the **avoid-static-obstacle** behavior is the gain. Increasing this value linearly increases the magnitude of the vector output by this behavior. This has the effect of causing the robot to exhibit a stronger aversion to obstacles. For more information about the behavioral parameters, see [3].

The human operator can also manipulate the behavioral parameters in terms of abstract personality traits. Making parameter changes in terms of personality traits allows a user, with no knowledge about the underlying behaviors and their parameters, to effectively modify the robots' behavior. Abstract parameters, which represent general kinds of behavioral or personality adjustments, are available for adjustment by the human operator. In our current system these characteristics include *Aggressiveness* and *Wanderlust*. The

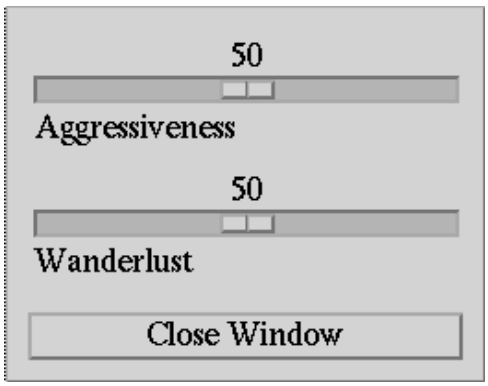


Figure 3: Personality window.

value of an abstract parameter controls the values of several individual low-level parameters. The operator uses slider bars (see Figure 3) to modify the value of an abstract personality trait, thus changing the global performance of the overall society.

For instance, the abstract parameter *Aggressiveness* determines the amount that the robot is focussed on achieving its goal. Aggressiveness controls the relative gains of the **move-to-goal** and **avoid-static-obstacle** behaviors. Increasing the *Aggressiveness* parameter results in an increase in the **move-to-goal** gain and a decrease in the **avoid-static-obstacle** gain. The effect produced is to cause the robots to be more strongly attracted to their goal location and be less repulsed by obstacles in their way, generally resulting in more direct albeit hazardous paths. Likewise, decreasing aggressiveness results in a decrease in the **move-to-goal** gain and an increase in the **avoid-static-obstacle** gain, producing safer behavior around obstacles but generally yielding longer paths.

Wanderlust represents the desire of the robot to randomly explore the terrain and how much attention is given to any goal-oriented behaviors. *Wanderlust* controls the gains of the **noise** and **formation** behaviors. Increasing the *Wanderlust* causes the robot to move more randomly and be less concerned with maintaining formation with the other robots.

4 Simulation Experiments

TELOP was tested on five different tasks. Three application tasks were tested in a simulation environment. These tasks include foraging, grazing, and herding.

4.1 Simulation Environment

The system was tested on a graphical simulation environment for three different tasks. The objects represented in the simulation environment include robots, obstacles, and attractors. Each robot's trail is depicted by a broken line. Every robot uses the same set of behaviors (a homogeneous society), but the sensory input for each is different, depending on the robot's location within the environment. The robots can sense objects within a certain radius around them. They have the ability to distinguish whether a sensed object is an obstacle, another robot, or an attractor.

The agents have a limited form of communication between themselves. A robot is capable of communicating its current behavioral state or the location of an attractor that it is acquiring or delivering [5]. The communication is simulated by using shared memory. Each agent only looks at this shared memory when there is no attractor within its sensing range.

In tasks that require the movement of attractors, more than one robot is allowed to contribute to the transport of the object at the same time. The net effect of this cooperation is simulated by having the

robots move the attractor farther during each time unit if there are more robots carrying it. The distance traveled while carrying an attractor is determined by the mass of the object and the number of robots carrying it.

4.2 Tasks

The use of teleautonomy in multiagent systems was tested in simulation for the tasks of foraging, grazing (vacuuming), and herding the robots into a pen. In all three tasks, an operator provided input at his own discretion.

In the foraging task, the robots wander around looking for attractors. When a robot finds a target object, it communicates its location to the other agents while simultaneously moving to acquire it. After its acquisition, the robot carries the attractor back to a homebase, then deposits it, and finally returns back to the task of searching for more attractors. If a robot cannot detect an attractor within its sensory radius, it checks to see if any other agent has communicated the location of another candidate goal object. If so, then the robot proceeds to acquire it.

In the grazing task, the robots are placed in an environment studded with obstacles. Initially, all of the floor that is not covered with an obstacle is "ungrazed". Each section of the floor that is ungrazed is treated as if it had an attractor on it. That is, a robot can sense an ungrazed section of floor from a distance, and it can also communicate the presence of an ungrazed section of the floor to the other robots. When an agent passes over an ungrazed region it becomes grazed. The task is completed when a certain percentage of the floor, specified in advance, has been grazed. The robots normally wander randomly until an ungrazed floor area is detected.

In the herding task, there is a pen with an opening formed of obstacles in the simulation environment. All the agents are initially outside of the pen. The robots remain in the *forage* state for the duration of the run and wander aimlessly in random directions. The robots are repulsed by the obstacles and the other robots. The task is to get all of the robotic agents inside the pen at the same time.

4.3 Results

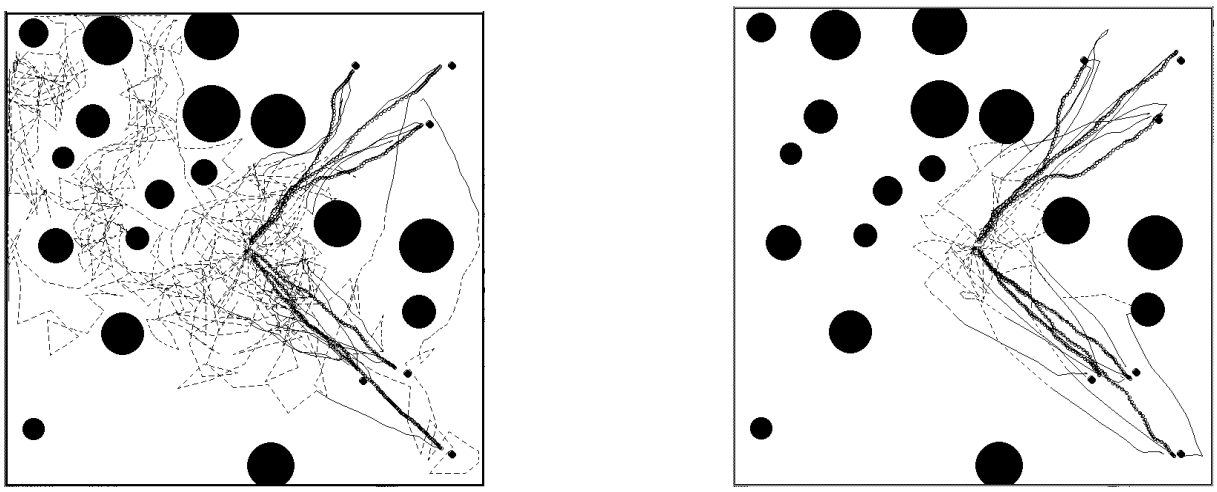
For the foraging and grazing tasks, tests were conducted that compared the total number of steps taken by the robots to complete the tasks with and without the help of a human. For the herding task, no comparison could be made between teleoperation and no teleoperation, because the likelihood of all the robots wandering into the pen by themselves at the same time is virtually nil. Interesting information was gained about this task nonetheless.

4.4 Foraging Results

In the tests conducted for the foraging task, three robots were used to gather six attractors. The density of obstacles in the environment was set to 10%. The total number of steps required to finish the task was measured both with and without teleoperation. If teleoperation is used wisely, it can significantly lower the total number of steps required to complete the task by greatly reducing the time spent in the *forage* state (i.e., the number of steps that the robots spend looking for attractors). If none of the agents currently sense an attractor, then the operator can assist by guiding the robots in one's direction. However, once the robots can sense an attractor, the operator should stop giving instructions, unless the instructions are to deal with a particularly troublesome set of obstacles. In general, the robots perform more efficiently by themselves than when under the control of a human if the agents already have an attractor in sight. The human's instructions tend to hinder the robots if they are already moving to acquire or return an attractor. Indeed, when teleoperation is used at all times, the overall number of steps required for task completion often increases when compared to no teleoperation at all. However, if the human only acts to guide the robots toward an attractor when none are currently detected, significant reductions in time for

task completion are possible. The average over six experimental runs of the total number of time steps required for task completion when teleoperation was used in this manner was 67% of the average task completion time when no teleoperation was used.

An example trace of a forage task without teleoperation is shown in Figure 4a. Another trace of the same forage task with a human operator helping the robots find the attractors when they did not have one in sensing range is shown in Figure 4b. The robots all started at the homebase in the center of the environment. In the run without teleoperation, the robots immediately found the two closer attractors at the lower right. Then they quickly found the two closer attractors at the upper right. At this point, the robots did not immediately detect the remaining two attractors. Two of the three agents proceeded by chance to the left and upper left sides of the environment, wandering unsuccessfully while seeking an attractor. Eventually, the other robot found the attractor in the lower right corner, and the other two robots moved to help with its return. After delivering it to the homebase, the robots wandered again for awhile without finding the last attractor. Finally, the last attractor was detected and successfully delivered to homebase. In the same world with the help of a human, the two protracted periods of wandering while searching for attractors are avoided. This indicates the types of environments where the use of teleoperation for the forage task is most beneficial. The greatest benefit from teleoperation can be seen when there are one or more attractors that are far from both the homebase and the start locations of the robots. Typically, this is when the robots do not sense the target objects without wandering for a while.



(a) (b)

Figure 4: Foraging task.
(a) Without Teleoperation. Note that the robots spent lots of time searching the upper left corner of the environment, but there are no attractors in that area.
(b) With Teleoperation. The robots did not spend time looking for attractors where there are none, because the human operator guided them in the direction of an attractor when the robots could not sense one themselves.

4.5 Grazing (Vacuuming) Task Results

For the grazing task, five robots were used. A sample run of a grazing task is shown in Figure 5. In this case, the robots performed poorly when a large amount of teleoperation was involved. Teleoperation only proved useful when the robots had difficulty in locating a section of ungrazed floor. When the robots had already detected an ungrazed area, they performed better without any input from the human operator. The agents' performance degraded considerably, often taking several times longer to complete the task, if teleoperation

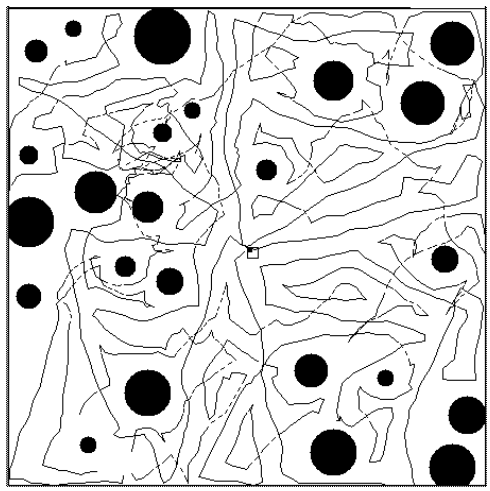


Figure 5: Grazing Task. The trails of the robots are shown for a grazing task with teleoperation.

was used when a robot had already located an ungrazed floor area. Moreover, since remaining untreated areas tend to be clustered together in large patches, the agents typically do not need to spend long periods of time looking for another ungrazed spot (which is opposite the case of the foraging task discussed above). Therefore, the use of teleoperation did not help significantly with the grazing task. When teleoperation was used solely to help the robots find ungrazed floor area when they were not already grazing, only a 4% improvement in average task completion time over six runs was observed when compared to not using teleoperation. Thus, when used wisely, teleoperation helped somewhat but not to a large extent.

4.6 Herding Task Results

For the herding task, five robots were herded into a pen that was 36 units long by 18 units wide, with a 12 unit long door in one of the longer sides. All of the robots started at one spot on the side of the pen with the door. In most test runs, the operator encountered no difficulty with this task. He was able to herd the robots into the pen with no problem. In some of the test runs, there were a few minor difficulties, such as robots wandering back out of the pen after having been herded in. However, the human operator was still able to complete the task without much frustration and in a reasonable amount of time. The results of a test run for the herding task are shown in Figure 6.

5 Robotic Experiments

Two generic tasks were tested on real robots. These tasks include directing the robots out of a box canyon and squeezing the robots through small spaces. TELOP was tested on a pair of Denning MRV-2 mobile robots, each about three feet tall with a diameter of 32 inches. Each robot is equipped with a ring of 24 ultrasonic sensors and shaft encoders. A Sun Sparcstation 5 served as the base station, running TELOP through *MissionLab*[1] [11]. The base station communicates with the robots using FreeWave radio links. The base station and human operator were on the third floor of the Manufacturing Research Center at Georgia Tech, and the robots were running on the first floor. The feedback to the operator consisted of the graphical depiction of the robots actions relayed in real-time by *MissionLab* and walkie-talkie communication between the operator and a human who was on the first floor observing the robots.

[1] *MissionLab* is a system for specifying and simulating multiagent robotic missions. *MissionLab* takes high-level military-style plans and executes them with teams of real or simulated robotic vehicles. The source code for *MissionLab* is available on the World Wide Web at the location http://www.cc.gatech.edu/ai/robot-lab/research/MissionLab/MissionLab.html.

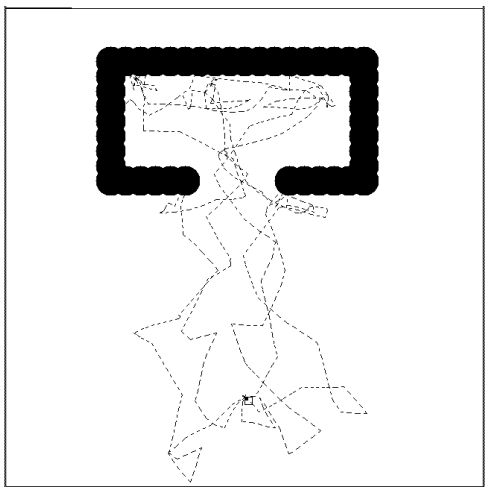


Figure 6: Herding task. Five robots were herded from a location outside the pen to the inside of the pen.

5.1 Tasks

Multiagent teleautonomy was then tested on a pair of Denning mobile robots for two other tasks including navigating the robots out of a box canyon and squeezing them through a tight space. In both tasks, the robots were using the **teleautonomy**, **avoid-static-obstacle**, **avoid-robot**, **move-to-goal**, **noise**, and column **formation** [6] behaviors.

In the first task, a box canyon, constructed from chairs, was set up in the room. The robots were started on the side of the room facing the opening of the box canyon. The robots were instructed to go to a location on the other side of the box canyon, such that the box canyon lay directly along the straight-line path from the start location to the destination. Since the robots operate purely reactively in this mode, they normally would get stuck in the box canyon and need to be helped out by the human operator.

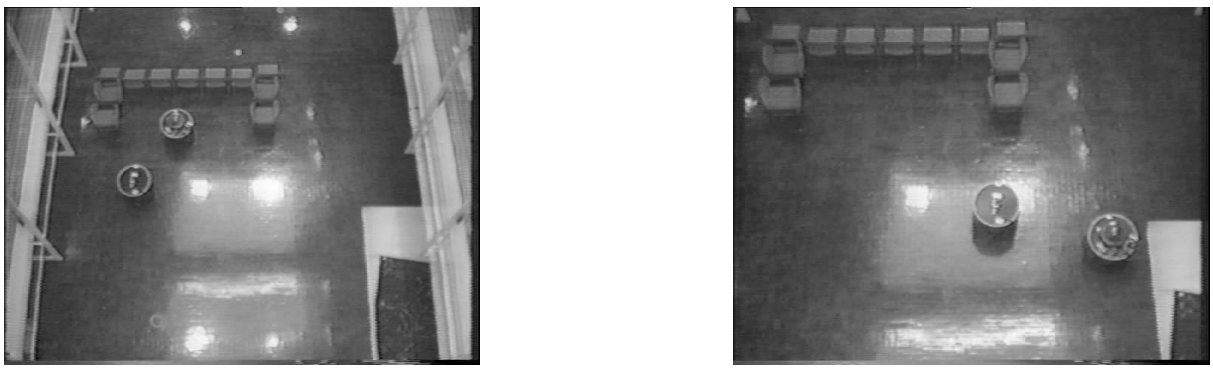
The task setup for the second task was the same as the first, except that the box canyon had a gap in it. The gap was sufficiently small so that the robots could not squeeze through it with the usual default gain setting for the **avoid-static-obstacle** behavior. The robots were provided with this default setting at the start of the task, so they would normally become stuck in the box canyon. The human operator should then be able to increase the robots' *Aggressiveness* to forcibly squeeze them through the gap.

5.2 Experimental Results

For the two tasks involving the two Denning mobile robots, the runs were videotaped, and a screen capture was taken of the tracking of the actual robots' movement from the *MissionLab* interface. *MissionLab* monitors the movement of the robots using information from their shaft encoders. This movement is plotted over an underlay depicting the task environment.

5.2.1 Box Canyon Results

As expected, the two robots got stuck in the box canyon while heading to the destination location (see Figure 7a). The human operator was able to use the on-screen joystick to steer the robots out of the box canyon, and around the side of it (see Figure 7b). After the robots were completely around the lip of the box canyon and were no longer in any danger of falling back into it, the operator released the joystick. Then the robots continued on to their destination autonomously. A trace of the robots' movement is shown in Figure 8.



(a) (b)

Figure 7: Box canyon task: (a) The robots are stuck in the box canyon. (b) The robots are being maneuvered out of the box canyon using the teleautonomy behavior.
The camera was located 4 floors above the robots, giving a birds-eye view of the action. The robots have circles of white tape on top of them to make them more visible.

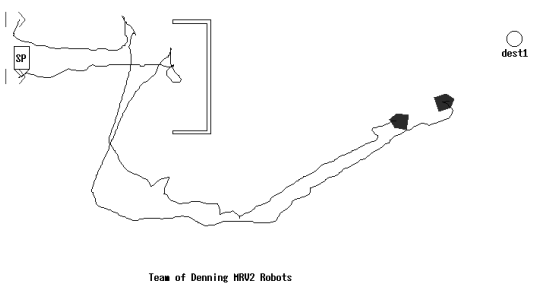


Figure 8: *MissionLab* trace of the robots movement during the box canyon task. The robots became stuck in the box canyon, and were then herded around the side of the box canyon by the human operator using the teleautonomy behavior.

(a) (b)

Figure 9: Squeezing task: (a) The robots are stuck in the box canyon with a gap. (b) The robots are being squeezed through the gap in the box canyon by making them more aggressive.
The camera was located 4 floors above the robots, giving a birds-eye view of the action.

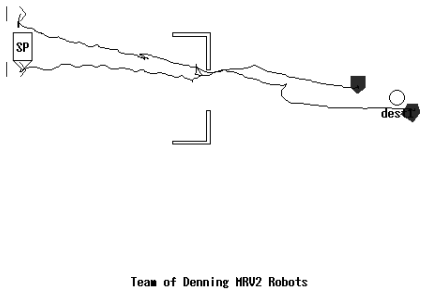


Figure 10: *MissionLab* trace of the robots movement during the squeezing task. The robots became trapped in the box canyon with a gap, but the operator squeezed them through the passage by increasing their aggressiveness.

5.2.2 Squeezing Results

The robots became trapped within the box canyon while heading to their destination (Figure 9a). This is a result of the default gain for the **avoid-static-obstacle** behavior being set too high and the gain for the **move-to-goal** behavior set too low for the robots to pass through the gap. The human operator slowly increased the robots' aggression until the robots successfully squeezed through the passageway (see Figure 9b). A trace of the robots' movement is shown in Figure 10.

6 Analysis

The use of the **teleautonomy** schema in conjunction with the robots' other behaviors proved particularly effective for the foraging task, while being less so for the grazing task (vacuuming). Herding the robots into a pen is possible without too much difficulty. During foraging, the best results were observed when teleoperation was used only to guide the robots in the direction of an attractor if one had not been previously sensed. For the vacuuming task, teleoperation was not significantly better than no teleoperation, although minor improvements were observed. The best results were again seen when teleoperation was used in

guiding the robots towards ungrazed areas that were outside the sensor (or communication) range of the agents.

Trying to herd the robots into a pen is straightforward, although some difficulties can arise. Two conceivable improvements can be used for this task regarding teleoperation. The first is to allow the operator to turn off the input from the teleoperation schema for specific robots but not for others, allowing the operator to concentrate on the outside robots without worrying what effects his actions will have on robots already inside the pen. The other improvement is to allow the human operator to completely stop a robot's movement when it is inside the pen. In this way, the output of the teleoperation schema could be thought of as producing a vector that nullifies the vectors produced by the robot's other schemas.

Another important point is that if the human operator is given unrestricted control of the magnitude of the vector produced by the teleoperation schema, it is possible for the operator to force a robot to collide with obstacles and other robots. The operator must be careful when increasing the gain of the **teleautonomy** schema so that this does not occur. It can be a delicate task to override the output of the **noise** schema, which is necessary to cause the robots to quickly move in a particular direction, while not overriding the **avoid-static-obstacle** behaviors.

When increasing the aggression of the robots, the operator should make small incremental increases until the robots squeeze through the small space. Then the operator should decrease the aggression again. If, however, the operator increases the aggression too much, the robots may charge through obstacles on their way to the goal (although this may be consistent with what the operator wants).

7 Summary

A method by which multiagent reactive robotic societal task execution can be influenced via human intervention has been demonstrated. This has been shown for a range of tasks including: improving the efficiency of foraging behavior; limited impact on improving grazing (vacuuming) activity; the ability to congregate agents in a small confined area (herding) under human guidance; maneuvering robots out of a box canyon; and squeezing robots through a small space.

A set of usability tests have been conducted on the interface of TELOP. These tests provided considerable information for designing the interface effectively. The evaluators in the usability tests were people working in our robotics laboratory. In the future, we plan to conduct at least one more set of usability tests on TELOP, using military students as evaluators.

The TELOP system has recently been integrated with the ARPA UGV Demo II architecture using the STXmcu mission control system for use on teams of HMMWVs. The teleautonomy behavior was demonstrated at a technical demo during Demo C of the UGV project in the summer of 1995. The teleautonomy behavior and the personality sliders are intended to be used in Demo II during the summer of 1996.

Acknowledgments

Support for this project is provided by the National Science Foundation under grant #IRI-9100149 and the Advanced Research Projects Agency under ARPA/ONR Grant #N00014-94-1-0215. The authors would like to thank Tucker Balch for his role in developing the simulation software used in the simulation experiments, and Jonathan Cameron and Doug MacKenzie for their role in developing *MissionLab*.

References

[1] Arkin, R.C., "Cooperation without Communication: Multi-agent Schema Based Robot Navigation", *Journal of Robotic Systems*, Vol. 9(3), April 1992, pp. 351-364.

[2] Arkin, R.C., "Reactive Control as a Substrate for Telerobotic Systems", *IEEE Aerospace and Electronics Systems Magazine*, Vol. 6, No. 6, June 1991, pp. 24-31.

[3] Arkin, R.C., "Motor Schema-Based Mobile Robot Navigation", *International Journal of Robotics Research*, Vol. 8, No. 4, August 1989, pp. 92-112.

[4] Arkin, R.C. and Hobbs, J.D., "Dimensions of Communication and Social Organization in Multi-Agent Robotic Systems", *From animals to animats 2: Proc. 2nd International Conference on Simulation of Adaptive Behavior*, Honolulu, HI, Dec. 1992, MIT Press, pp. 486-493.

[5] Balch, T. and Arkin, R.C., "Communication in Reactive Multiagent Robotic Systems", *Autonomous Robots*, Vol. 1, No. 1, pp. 27-52, 1994.

[6] Balch, T. and Arkin, R.C., "Motor Schema-based Formation Control for Multiagent Robot Teams", *1995 International Conference on Multiagent Systems*, San Francisco, CA, pp. 10-16, 1995.

[7] Brooks, R., Maes, P., Mataric, M., and More, G., "Lunar Base Construction Robots", *IEEE International Workshop on Intelligent Robots and Systems (IROS '90)*, pp. 389-392, Tsuchiura, Japan, 1990.

[8] Cao, Y., Fukunaga, A., Khang, A., and Meng, F., "Cooperative Mobile Robotics: Antecedents and Directions", *Proc. 1995 International Conference on Intelligent Robotics and Systems (IROS '95)*, August, 1995, Pittsburg, PA, Vol. 1, pp. 226-234.

[9] Lyons, D.M., and Arbib, M.A., "A Formal Model of Computation for Sensory-Based Robotics", *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 3, June 1989.

[10] MacKenzie, D. and Arkin, R.C., "Formal Specification for Behavior-based Mobile Robots", in *Mobile Robots VIII*, Boston, MA, Nov. 1993.

[11] MacKenzie, D., Cameron, J., Arkin, R., "Specification and Execution of Multiagent Missions", *Proc. 1995 Int. Conf. on Intelligent Robotics and Systems IROS '95*, Pittsburg, PA, Vol. 3, pp. 51-58, 1995.

[12] Mataric, M., "Minimizing Complexity in Controlling a Mobile Robot Population", *1992 IEEE International Conference on Robotics and Automation*, Nice, pp. 830-835.