Cooperative Multi–Robot Box–Pushing

Maja J Matarić **Martin Nilsson** **Kristian T. Simsarian**

Volen Center for Complex Systems
Computer Science Department
Brandeis University
Waltham, Ma 02254
tel: (617) 736-2708 fax: 736-2741
maja@cs.brandeis.edu

Swedish Institute of Computer Science
S-16428 KISTA
Stockholm, Sweden
fax: +46 (8) 751-7230
tel: +46 (8) 752-1570
mn@sics.se kristian@sics.se

Note: Submitted to the invited session on Collective and Cooperative Robotics, proposed by Maja J Matarić.

Abstract

This paper describes the problem of task sharing between two autonomous six–legged robots. The robots are equipped with two-way communication, object and goal sensing, and a repertoire of basic behaviors. The performance on the selected task, pushing an elongated box toward a goal region, is difficult for a single robot and improves significantly when performed cooperatively. The cooperative solution, however, requires careful coordination between the robots. We present an approach that takes advantage of a simple communication protocol to compensate for the robots' noisy and uncertain sensing and actuation, and their partial knowledge about the world. We demonstrate our approach on a series of experiments with two physical robots.

1 Introduction

The work in this paper is concerned with the problem of mobile robot task sharing. In particular, we are focusing on manipulation tasks with sufficiently challenging dynamics to require the careful cooperation of two (or more) robots.

In mobile robot domains sensing and actuation is noisy and uncertain. Consequently, robots can only have partial knowledge about the world. We explore the use of multiple robots to improve and speed up performance on manipulation tasks that might be possible with a single robot but become more efficient through cooperation. In particular, we describe a simple communication protocol that compensates for the robots' incomplete and inaccurate information about the environment while also reducing interference and resource competition between the two robots.

2 The Domain

The problem of box–pushing is used to ground the research. It is a problem domain which has both theoretical interest and practical applications. Box–pushing represents a large class of practical object manipulation tasks that appear to require tight feedback, and interaction with and control of real-world physics and dynamics. As such, it involves a basic capacity that is expected of any autonomous robot physically interacting with objects in the world.

Additionally from a theoretical standpoint, box–pushing is a variant on the canonical object manipulation problem that draws on issues in fine motion as well as high level planning and control. Box–pushing is related to the well-known "piano–movers problem" (Schwartz & Sharir 1983), in that it requires the achievement of the top–level goal of delivering the box to a particular location, as well as the maintenance of low–level requirements including obstacle avoidance, maintaining contact with the box, and maintaining forward motion.

We set up the problem so that in our domain the box–pushing task does not require cooperation, but is shown to be significantly improved by it. We are interested in evaluating the benefits of cooperative solutions relative to single–agent approaches. Consequently, we are not considering tasks that are impossible for a single robot. Furthermore, we do not attempt to derive the optimal number of agents for the task. Some interesting simulation (see for example Arkin, Balch & Nitz (1993)) demonstrates that increasing numbers of agents can result in overwhelming interference and a decrease in global efficiency.

In our scenario, the dynamics of the robot and box interaction are designed so as to increase efficiency in terms of forward movement of the box. We will demonstrate a communication and turn–taking protocol, described in a later section, that satisfies both the high–level and the low–level goals of the system.

Huberman (1990) shows that the introduction of communication can produce superlinear performance improvements on problems that otherwise allow only linear efficiency improvements through the use of cooperative solutions. We will experimentally demonstrate that our box–pushing scenario falls into that class of problems. We will give strong evidence that single–robot and non communicating two–robot solutions are inferior, in terms of performance, when matched with the communication and turn–taking protocol given in this paper.

3 Experimental Environment

3.1 The Robot Hardware

This section describes the equipment used in the experiments. A brief overview of the robots is followed by a description of our modifications, and

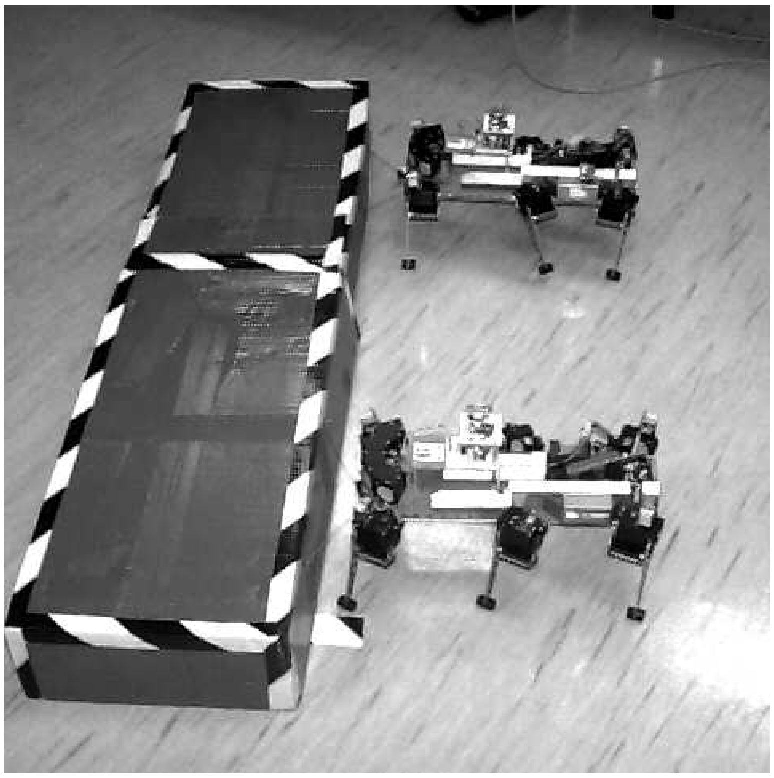

Figure 1: Two Genghis-II six–legged robots were used in the experiments. The robots are 30 cm long and 15 cm high. Each leg is driven by two servos, one for each degree of freedom. The sensors include two frontal contact whiskers and five pyroelectric sensors for detecting moving IR sources. The robots are powered by rechargeable batteries and controlled by five 68HC11 microprocessors programmed in the Behavior Language.

the communication system we have added.

For the experiments, we used two commercially available Genghis-II six-legged robots as shown in Figure 1. The robots are programmed in the Behavior Language, a parallel distributed robot programming language based on the Subsumption Architecture (Brooks 1990, Brooks 1986).

Using legged robots instead the more common wheeled locomotion has strong consequences on the dynamics of the robot–box interaction. When perpendicular to the box, the robots "push" it with their whiskers. However, at any more acute incident angle, they effectively "kick" it with whatever leg is closer. The kicking has the effect of turning the box faster than would a pushing force exerted by a flat–faced or round, wheeled robot. Consequently, the cooperative algorithm must account for the concurrent and independent displacement of the box by each of the robots.

Since the Genghis robot does not have any provisions for user–added sensors, we introduced communication facilities by adding wires directly to some unused pins of the main on–board processor. We used the wires to implement an I^2C interface, to our customized board mounted on top of the robot (see Figure 2). The user program communicates with the board via primitive operations in the Behavior Language. The bit rate for the I^2C

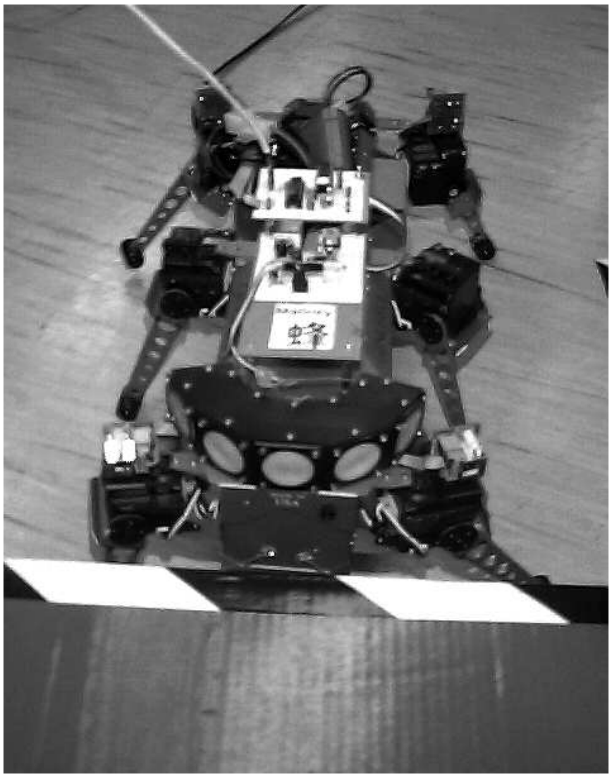

Figure 2: The communication hardware is mounted on top of the robot.

link is about 20 kbit/s, limited by the speed of the 68HC11 processor. This interface required no change to any of the resident Genghis system software.

The board we added to the robot has two main responsibilities. One is to allow us to add arbitrary sensing and actuating extensions, which can be controlled by the user program running on the Genghis main processor, via the I^2C interface. The other is to handle the communication with other robots. For these purposes, the board is equipped with two PIC16C84 processors, of which one is entirely dedicated to communication.

The communication system runs completely asynchronously with the Genghis main processor. For the experiments described here, we used a cable between robots, although the system is actually designed for radio communication. With a 5 meter cable, we could easily transmit 50 kbit/s without errors, and higher speeds than this are possible with the existing hardware, but this not meaningful due to the limitations of Genghis processor speed and bandwidth limitations of the radio link. We expect to have the non–cable radio version fully operational in the near future.

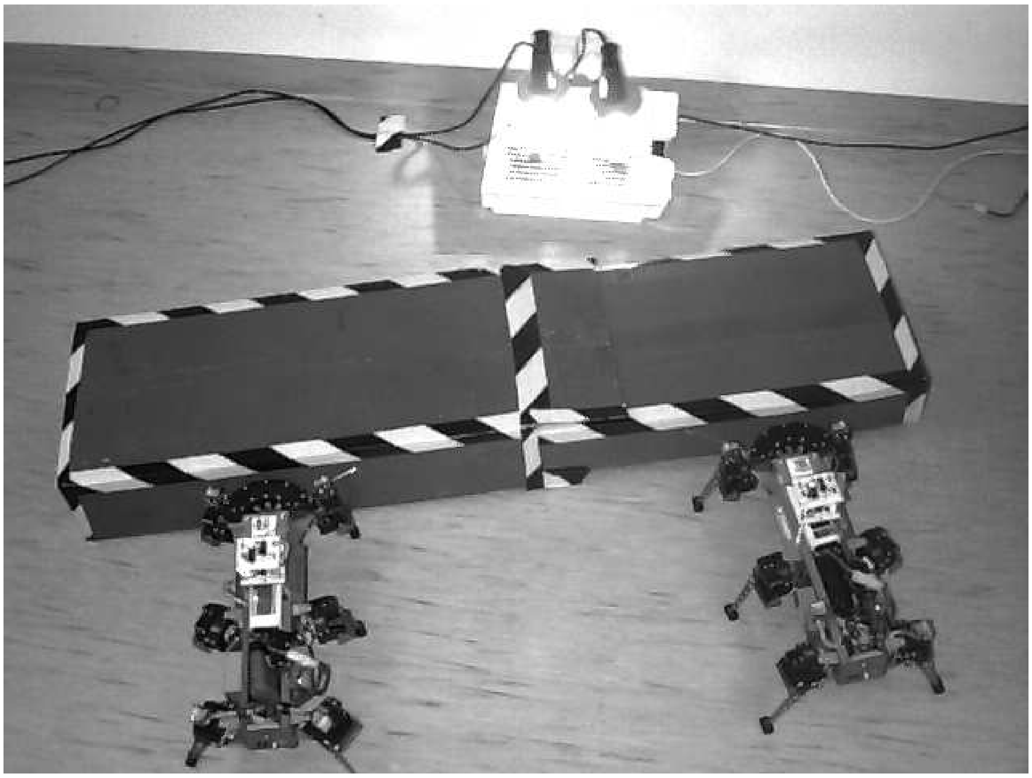

Figure 3: This photo illustrates the experimental environment. The lights indicate the location of the goal region. The box to be pushed is long enough to require force on both sides. It is also not so high as to obscure the input into the light sensors on the robots.

3.2 The Workspace

The two robots operate in a square arena, approximately 3 meters to a side. The box to be pushed is 14 cm high, 35 cm wide, and 1.3 meters long. The goal region, located at one end of the arena, is indicated by two tungsten light sources, which emit IR radiation detectable by the robot's pyroelectric sensors. The experimental scenario is illustrated in Figure 3. The arena is not fixed in that we have the ability to change the location of the goal (i.e. the light source) both prior to and during experiments. However for the experiments described in this paper the goal location remained fixed in order to give a valid comparison.

3.3 The Sensing Strategies

Two main sensing modalities were employed for this task, tactile and and photoelectric. The first corresponds to the whiskers which are used to detect contact with the box. The latter corresponds to the pyroelectric sensors used to sense the goal.

The whiskers are designed to be a 2 binary contact detectors. In this task they are employed in an algorithm that exhibits the behavior of maintaining perpendicular contact to the box. Although the information from the sensors is simple, a tight control loop based on the whisker input used to servo the body is very effective at maneuvering and maintaining contact.

The pyroelectric sensors detect heat sources. The tungsten filaments in the goal–lights create a source for adequate sensor response. The pyros detect a heat edge moving across the sensor element, thus can sense the direction of movement of the heat source. In addition, the pyro sensors are positioned in a semi-circle around the face of the robot and have a wide field of view and give additional relative-positional sensitivity. The pyro sensing algorithm used in this work looks at the absolute difference, with respect to time, of light responses, and compares this to a threshold. The highest response direction is then kept in a register made available to the main control loop and to the other robot. Implementing the algorithm in this way gives the robot memory as to the last known location of the goal. In effect, if the goal falls out of sight, the position will be known until a larger response is triggered.

The above sensing strategies produces behaviors that can easily be in conflict. For our simpler experiments, it will be shown that it is often the case that the robots will have to choose to maintain one behavior while relinquishing maintenance of the other.

4 The Approach

The interaction dynamics of a six–legged walking robot with its environment are complex. They are further complicated by the introduction of another such robot, and a physical object which both can manipulate simultaneously. In particular due to the noise, error, and uncertainly in the sensors and effectors in such a system, a predictive, functional model of the system is not available. Consequently, the designer is left with a choice of either explicitly programming in a partial model, or implicitly embedding one in the controller. Our work addresses the latter approach, by demonstrating an effective controller constructed with a simple implicit model.

We propose that for a noisy system of the kind we are describing, the effort of providing an explicit model may not be justified. Such a model would be necessarily inaccurate and possibly misrepresentative of the full complexity of the dynamics involved in the domain. While such a model could be acquired by gathering statistics from observing the system's behavior, we suggest an alternative approach that utilizes communication between the robots to simplify control and bypass prediction.

One of the most interesting issues in cooperative multi–agent systems is the extent to which each agent models the other in order to effectively interact with it. The ability to maintain explicit models of other agents and their goals has a high computational, sensory, and communication overhead (Gasser & Huhns 1989, Rosenschein & Genesereth 1985, Axelrod 1984). Our previous work on collective robot behavior (Matarić 1993, Matarić 1992), drawing on biological and ethological evidence (McFarland 1987, Gould 1982), has demonstrated that effective group behavior can be achieved in the complete absence of explicit models. The work described here follows in the same vane, exploring simple cooperative control strategies that utilize communication to minimize the need for explicit modeling and prediction.

In our scenario, the robots have no explicit knowledge of the world physics, of their own mechanics and dynamics, nor of those of the other robot. Furthermore, they do not anticipate or predict the effects of their own actions, nor those of the other robot. Instead, their control strategy is based on obtaining as much immediate sensory information as possible from their own sensors including their communication channel with the other robot. Our underlying assumption in this environment is that the dynamic nature of the robot and object interactions enforce and support a Markovian world in which history plays no significant role. For the purposes of control, the state of the system at any point in time can be completely determined by the inputs into the robots sensors.

Uncertainty and asynchrony challenge this assumption but do not violate it. Due to sensor limitations and noise, the robots perceived state can often be inaccurate. Furthermore, if the robots move simultaneously and continuously, their perceptual information, in particular that dealing with the actions of the other robot, is likely to be outdated. Consequently, they would require some history to keep track of what behavior the other robot may be currently executing. To avoid complications, this scenario has been simplified by imposing synchronized turn-taking for the robots. Thus, the Markovian assumption can be supported by imposed a turn–taking protocol in the robot's behavior. We propose a protocol that serves to simultaneously synchronize the robots' actions as well as to interleave communication for expanding each of the robot's perceptual capabilities though information sharing.

The following is the basic control loop, identical for both robots, consisting of two main concurrent processes. The robots have four primary actions, {Stop, Move-forward, Left, Right}, the set of which form the actions for A and M. After performing an action the robots each perform the behavior of maintaining contact with the box, which in effect servos the robot's body perpendicular to the box–face.

```
Whenever "my-turn" message is received:
    get own sensory data S
    get other-robot's sensory data O
    combine S and O into current joint state G
    use G to select the optimal next action A
    use A to select the optimal message-action to send M
    send M
    perform A
    maintain box contact
    send "your-turn" message

Whenever an action-message M is received:
    perform M
    maintain box contact
```

The outlined algorithm embodies an egalitarian task sharing strategy in that the robots take turns controlling the actions at each time step. This contrasts a centralized framework in which both robots would be directed from a global controller. It also differs significantly from a master–slave scenario in which one of the robots would retain control throughout the lifetime of the task, and command the other's actions at each step.

In our scenario, the two robots are completely equal both in terms of their control capabilities and in the amount of information they have about the world. Each robot depends on the other for additional information, i.e. for the state of the "other side" of the box. However, once having obtained that information, each robot can apply its own control rule, and instantiate the appropriate action.

Given this scenario, we can postulate whether the role of the other robot could be replaced with a remote sensor. However, such a sensor would only fill the perceptual but not the action aspect of the cooperative task. As described earlier, our task was selected so as to require cooperation both in the sensing and execution. Thus, a single robot with any amount of sensing would still have to repeatedly move from along the side of the box to the other to push and move it effectively.

Our two–robot system utilizes cooperation at three levels: sensing, action, and control. The control system is duplicated and independent across the two robots which increases robustness through redundancy. Having two independent robots taking turns eliminates the possibility of a one–way bias due to the controller's susceptibility to a variety of perturbations that arise in noisy and dynamic environments and from a particular location.

5 Experiments and Results

We conducted a series of experiments, increasing in sophistication, in order to compare and evaluate the different aspects of the proposed approach. In the first experiment a single robot used a simple reactive strategy to push the box toward the goal. The next experiment employed two robots using identical simple strategies without synchronization. Finally the third experiment used two robots with a cooperative communication and synchronization strategy. For all the experiments the goal and starting locations were the same.

The first set of experiments were used to evaluate the difficulty of the task for a single robot. A single robot was positioned at a distance from a box. It walked forward, found the box, and proceeded to push it toward the goal. It was given the simple strategy of maintaining contact with the box and moving toward the light. The robot performed this task from two different starting points on the box and from two different box starting points in the room in order to remove directional bias. The first robot starting point was in the middle of the box (see figure 4). From the middle starting point, without a more complex control algorithm, the robot was unable to maneuver the box toward the light. Instead the robot was forced to either maintain contact with the box or move toward the goal.

With a more complicated strategy the box could be pushed toward the goal by a single robot if it repeatedly switched sides. In the ideal case this strategy would take twice as long as when the box is pushed by two turn-taking robots, one at each end. Due to the legged locomotion employed here, the time to perform a strategy which alternates pushing sides would effectively be more than double the required time because the robot does not have great dexterity in performing lateral movements. In addition due to movement and sensing deficiencies the robot would have difficulty in determining and maintaining a centered position relative to the box.

In the next experiment the robot was placed in a position so that it could be guaranteed to reach the goal (see figure 5). In this experiment we wanted to find out how long it would take a single robot to push the box toward the goal in the best possible condition. The time is measured in *steps* which are consistent throughout the experiments and equal to approximately 6 seconds. We found that it took an average of 15.5 steps. We performed these trials in order to compare the single robot task time with the dual robot task times.

The second set of control experiments consisted of testing the performance of two independently acting box–pushing robots, each positioned and

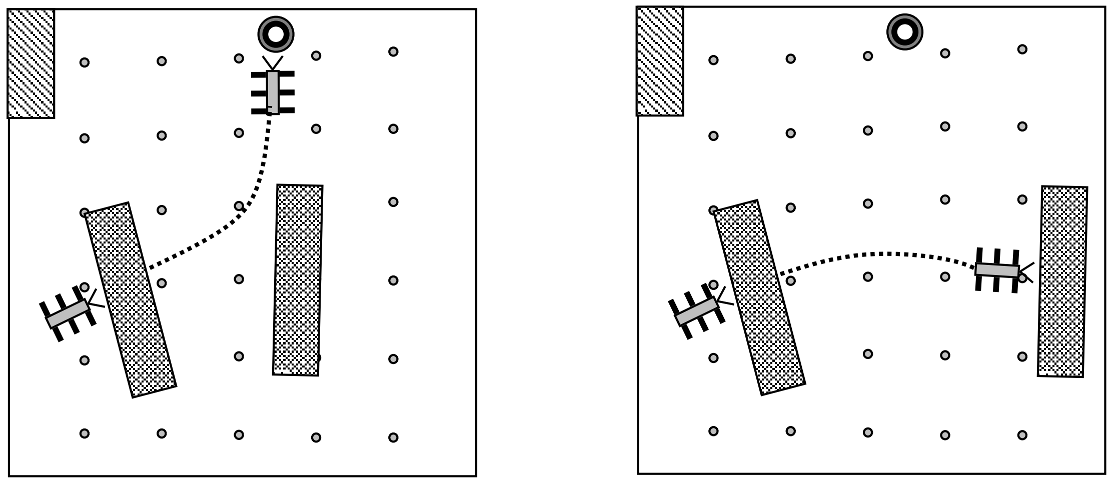

Figure 4: In all figures the starting and ending configurations for the box and robot are overlaid. The path the robot took through the trial is indicated by a dotted line. The goal is indicated by the circle at the top of the workspace. This figure shows the two observed behaviors from a number of runs when a single robot is started and positioned at the box center. The robot would choose to maintain contact with the box and lose track of the goal or abandon the box and move toward the goal. The average number of steps to reach the light was 10.

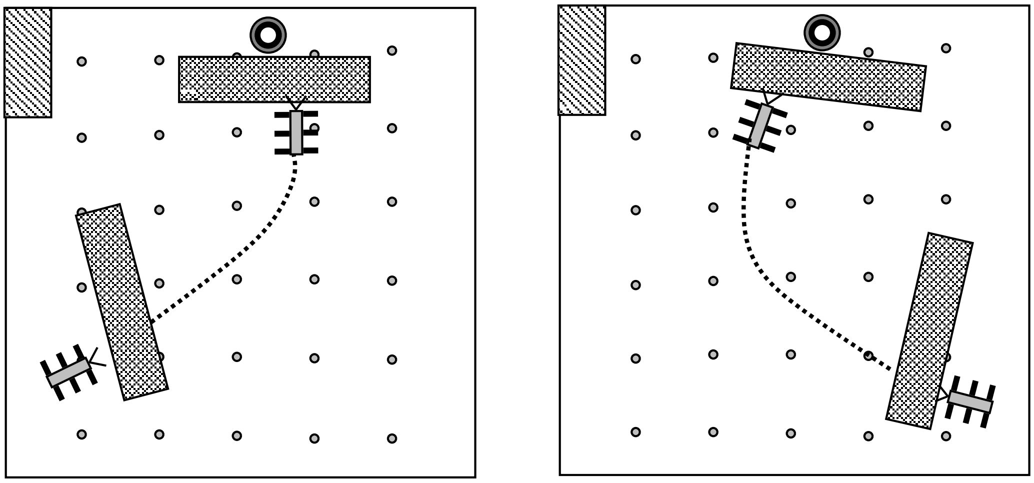

Figure 5: In this case the robot was positioned in order to guarantee reaching the goal with the box. The average number of steps taken was 15.5. This experiment was performed in order to give a comparison to the dual robot task times.

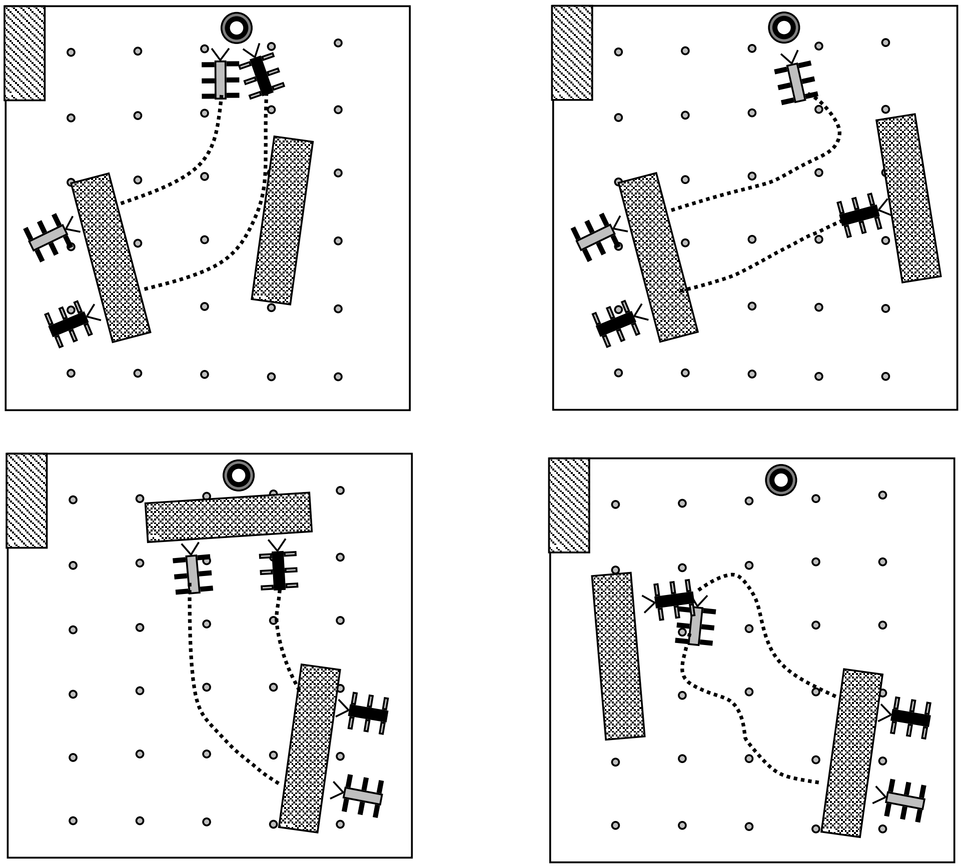

Figure 6: Prototypical trials of two non-communicating robots pushing the box toward the goal. In this case the robots could not synchronize and followed different behaviors. From left to right, in the first case both robots abandoned the box for the goal. In the second case, one robot abandoned the box for the light while the other pushed it out of the arena. In the third case, the robot successfully pushed the box toward the goal and in the forth case, the outside robot abandoned the box and interfered with the progress of the inside robot. When successful, the average number of steps to reach the goal was 8.

initiated at one end of the box. It might appear reasonable to expect that the two independent robots should, while simply servoing to the light, deliver the box to the goal. However, without communication, the actions of the two robots often have the effect of clobbering the other's actions while pursuing their own goals, and the box becomes turned to one side or the other instead of being pushed directly toward the light (see figure 6). It is also the case that one robot may abandon the box for the light or alternatively maintain contact with the box and forgo the goal of moving toward the light after the box has been pushed past an unrecoverable point. Once the box falls "out of bounds" it is never recovered since both robots continue to push and servo consistently while the goal is no longer actively visible. In those scenarios, either the box is turned too far, or one of the robots permanently loses contact with it. In addition a stable starting point where the robots would be guaranteed success could not be found. In all cases, what is needed is a cooperative turn–taking strategy for reaching the goal, such as the one we describe next.

In the final set of trials, the robots were give the control strategy described in the previous section. In all runs the robots successfully maneuvered the box to the goal (see figure 7). Because the robots work in conjunction for

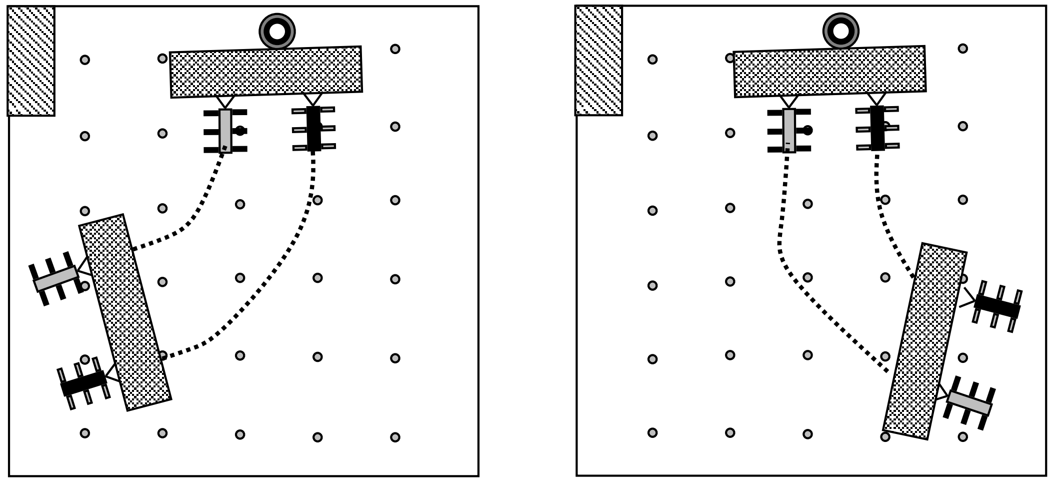

Figure 7: In this case the robots were synchronized and communicating. Each time they were successful in pushing the box toward the goal by cooperatively performing the joint basic behaviors of turning and walking. The average time to reach the goal was 6 steps.

turns and do not "clobber" each other's moves or get in each other's way it is possible for the robots to take the shortest path to the goal as well as push at the same time. Therefore it is not surprising that the two communicating robots can perform the task in much less than half the time for one robot and 25% less time than the case where two independent robots pushed the box. Not only was the cooperative demonstrated to be faster it was also successful for all cases we ran.

6 Related Work

While there is a large body of work on abstract multi–agent systems, very few physical multi–robot systems have been implemented. This section reviews only those systems that have dealt with cooperative object manipulation tasks.

The systems that are most related to ours are those designed to be cooperative. In these systems, the two or more robots are aware of each other's existence, and can sense and recognize each other directly or through communication. In contrast to our work, Caloud, Choi, Latombe, LePape & Yim (1990), Noreils (1992) and Noreils (1993) remain faithful to the state–based framework, and apply a traditional planner–based control architecture to a box–moving task implemented with two robots in a master–slave configuration. At the other end of the control spectrum, Kube (1992) and Kube & Zhang (1992) describe a series of simulations of robots performing a collection of simple behaviors which are being incrementally transferred to physical robots. The behaviors included locating and pushing a box. Donald, Jennings & Rus (1993) report on the theoretical grounding for implementing a cooperative manipulation task with a pair of mobile robots. In an approach most related to ours, Parker (1993) and Parker (1994) describes a behavior–based task–sharing architecture for controlling groups of heterogeneous robots, and demonstrates it on a set of physical robots performing toxic waste cleanup and box–pushing.

7 Summary and Conclusions

This paper has addressed the problem of cooperative box–pushing with two autonomous six–legged robots. The chosen task was achievable by a single robot but could be performed much more effectively with two robots. However, the non–trivial dynamics of the task require careful coordination in any cooperative solution.

We presented a cooperative strategy that takes advantage of a simple communication protocol to compensate for the robots' noisy and uncertain sensing and actuation, and their partial knowledge about the state of the box. Our solution is fully distributed in that the robots share sensory and control information at each turn.

The cooperative box–pushing method for communicating robots we demonstrated is superior to strategies not employing communication and cooperation we have tested. In particular, our experiments have shown the existence of situations where a dual robot cooperative strategy is twice as effective in task time over a single robot performing the task in the same conditions, and approximately 25% faster than non–cooperating robots when they succeed. In addition, our method has proven reliable in situations where the simple algorithms fail.

References

Arkin, R. C., Balch, T. & Nitz, E. (1993), Communication of Behavioral State in Multi-agent Retrieval Tasks, *in* 'IEEE International Conference on Robotics and Automation', pp. 588–594.

Axelrod, R. (1984), *The Evolution of Cooperation*, Basic Books, New York.

Brooks, R. A. (1986), 'A Robust Layered Control System for a Mobile Robot', *IEEE Journal of Robotics and Automation* **RA-2**, 14–23.

Brooks, R. A. (1990), The Behavior Language; User's Guide, Technical Report AIM-1127, MIT Artificial Intelligence Lab.

Caloud, P., Choi, W., Latombe, J., LePape, C. & Yim, M. (1990), Indoor Automation with Many Mobile Robots, *in* 'IROS-90', Tsuchiura, Japan, pp. 67–72.

Donald, B. R., Jennings, J. & Rus, D. (1993), Experimental Information Invariants for Cooperating Autonomous Mobile Robots, *in* 'Proceedings, International Symposium on Robotics Research', Hidden valley, PA.

Gasser, L. & Huhns, M. N. (1989), *Distributed Artificial Intelligence*, Pitman, London.

Gould, J. L. (1982), *Ethology; The Mechanisms and Evolution of Behavior*, W. W. Norton & Co., New York.

Huberman, B. A. (1990), 'The Performance of Cooperative Processes', *Physica D* **42**, 38–47.

Kube, C. R. (1992), Collective Robotic Intelligence: A Control Theory for Robot Populations, Master's thesis, University of Alberta.

Kube, C. R. & Zhang, H. (1992), Collective Robotic Intelligence, *in* 'From Animals to Animats: International Conference on Simulation of Adaptive Behavior', pp. 460–468.

Matarić, M. J. (1992), Minimizing Complexity in Controlling a Collection of Mobile Robots, *in* 'IEEE International Conference on Robotics and Automation', Nice, France, pp. 830–835.

Matarić, M. J. (1993), Kin Recognition, Similarity, and Group Behavior, *in* 'Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society', Boulder, Colorado, pp. 705–710.

McFarland, D. (1987), The Oxford Companion to Animal Behavior, *in* 'Oxford, University Press'.

Noreils, F. R. (1992), An Architecture for Cooperative and Autonomous Mobile Robots, *in* 'IEEE International Conference on Robotics and Automation', pp. 2703–2710.

Noreils, F. R. (1993), 'Toward a Robot Architecture Integrating Cooperation between Mobile Robots: Application to Indoor Environment', *The International Journal of Robotics Research* **12**(1), 79–98.

Parker, L. E. (1993), Learning in Cooperative Robot Teams, *in* 'Proceedings, IJCAI-93 Workshop on Dynamically Interacting Robots', Chambery, France, pp. 12–23.

Parker, L. E. (1994), Heterogeneous Multi–Robot Cooperation, PhD thesis, MIT.

Rosenschein, G. & Genesereth, M. R. (1985), Deals Among Rational Agents, *in* 'IJCAI-85', pp. 91–99.

Schwartz, J. & Sharir, M. (1983), 'On the Piano Mover's problem. II. General techniques for computing topological properties of real algebraic manifolds', *Advances in Applied Math.*