Stagnation Recovery Behaviours for Collective Robotics

C. Ronald Kube and Hong Zhang
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1
kube@cs.ualberta.ca   zhang@cs.ualberta.ca

May 31, 1994

Abstract

Accomplishing useful tasks with a collection of decentralized mobile robots will require control methods that deal effectively with a number of unique problems that impede the system's progress. Reactive control architectures can easily cause the problems of stagnation and cyclic behaviour, both characterized by a lack of progress in achieving a task. In this paper we present one possible solution to stagnation recovery, motivated from the study of group transport in ants and demonstrate its use in a box-pushing task. By using stagnation recovery behaviours, which are triggered by a lack of progress in the task-achieving activity of the system, the collective system can monitor its own advancement in a decentralized manner. A set of such behaviours are progressively ordered using timeouts, with each set designed for a specific recovery strategy. The stagnation recovery behaviours have been tested in simulation with the results to be mapped onto a set of ten autonomous robots presently under construction.

1 Introduction

Collective Robotics is a new approach to designing task-achieving systems using a collection of simple autonomous mobile robots. Tasks are accomplished through the collective effort of a group of robots. Among the ways of controlling such a group of homogeneous robots, the Animat approach [20], which models whole but simple animal-like systems, provides a computational model suitable for studying both perception and motor control. Systems that employ reactive control using sensory-response mechanisms can suffer from stagnation and cyclic behaviour both characterized by a lack of task progression. Potential solutions to stagnation recovery for a multi-robot system may be found among the social insects, Nature's example of decentralized control. Using simulation, we have studied stagnation recovery behaviours, as they apply to a box-pushing task, which we plan to test on a system of 10 mobile robots currently under construction.

Using a collection of robots to achieve a specific task is an approach that has been applied to such problems as material handling [4, 16, 7], map making [15], formation marching [19, 13, 11, 14], and box-pushing [3, 1, 8]. Such a novel approach requires a rethinking of the original problem in terms of a distribution of both actuation and sensing. Robots capable of only local sensing and without a way of explicit communication can communicate through the task itself, while carrying out a task such as moving a common load. Any movement of the load can be immediately sensed by all robots carrying or pushing the load. Thus, although explicit communication between robots is not used in our approach, information is communicated to all the robots through the task itself.

When a multi-robot system is controlled using a reactive architecture, with simple sensory-response control mechanisms, then a method to monitor the task's progression is required to serve as a way of positive feedback. Without such a deadlock avoidance mechanism, stagnation may occur in which the task ceases to progress. For example, in a box-pushing task the net force applied by the robots may equal zero if the robots are evenly distributed around the perimeter of the box. In such a case, a robot might attempt indefinitely to push the box unsuccessfully. An equivalent problem can be found in Nature among ants displaying a group transport behaviour [12]. How do ants equipped with simple sensory-response behaviours deal with the stagnation that results when the item they are transporting becomes stuck? The answer may serve to motivate possible solutions for robots whose feasibility can be studied in simulation and tested by mapping the simulation results onto real robots.

In Section 2, we examine group transport by ants and the strategies they employ when faced with impediments to task progression. Such strategies include realignment and repositioning used to increase the applied force required to move an object. In Section 3, we examine several cases of stagnation in a box-pushing task, and propose a mechanism which can be used as part of a set of stagnation recovery behaviours. The behaviours are then simulated with examples to demonstrate the technique. Finally, Section 4 discusses why stagnation recovery is useful in reactive systems that use a stimulus response control mechanism. We also describe our collection of 10 robots currently under construction and compare it to our previous system composed of five robots designed to locate and collectively push a box too heavy for a single robot.

2 Group Transport by Ants

Nature has graciously provided us, by way of the social insects, with an example multi-agent system whose decentralized control is based solely on locally sensed information. Moreover, ants exhibit a group transport behaviour, used in both food and prey retrieval tasks, in which stagnation problems arise and are solved using simple recovery strategies. These strategies have motivated our approach to stagnation recovery in collective robotics.

Group transport is the cooperative movement of a load by two or more ants. Very few studies have examined this behaviour found almost exclusively in ants, but those that have show group transport to be an efficient way of moving a load with a small workforce [12, 5, 17, 18]. Food is generally consumed within the safe confines of the nest and must be first torn apart before consumption. Ants must either transport the food item as a whole from its location or dismantle it into small enough pieces to be carried back to the nest by an individual. The efficiency of group transport is evident in Moffett's experiment using a large piece of cereal carried by 14 ants, a food item which would have required 498 ants had individual pieces been carried solitarily [12]. Franks [5] has also determined the efficiency of group transport with ants capable of moving items which are more than the sum of pieces carried by the individual ants in a group. Since items are always carried at a standard retrieval speed, Franks hypothesizes that this superefficiency is obtained by a group's ability to overcome the rotational forces necessary to balance a food item.

A detailed study of the movement patterns involved in group transport was carried out by Sudd [17, 18] in which it was concluded that although the behaviour of ants in a group transport was similar to that of single ants, group transport showed cooperative features [17]. When an ant attempts to move a food item it first tries to carry it. If the item is restrained in any way the ant will next attempt to drag it. Sudd [18] suggests that the resistance to transport determines whether to carry or drag the item. After some seconds are spent on resistance testing, the ant will try and realign the orientation of its body without releasing the item. This has the effect of altering the direction of applied force and may be sufficient to move the food item. If the item still cannot be moved the ant will release its grasp and reposition itself by grasping at another spot. If this final attempt does not result in movement the ant will recruit other ants to the food site. The lighter the load the longer an ant will attempt to move it. Sudd cites an ant will spend up to four minutes before recruitment takes place for items less than 100mg, and up to one minute for items greater than 300mg.

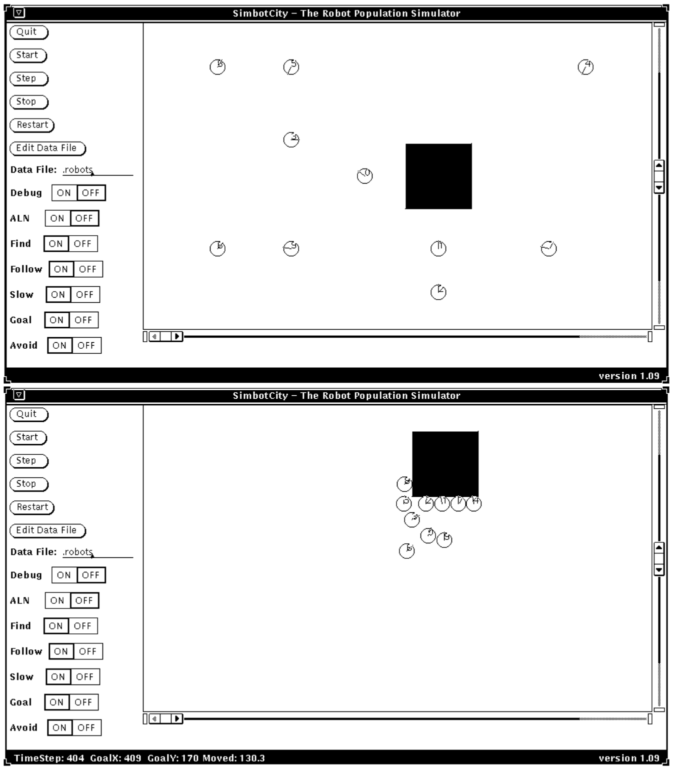
The strategies of realigning, and repositioning are used by ants in the group if during transport the item gets stuck, and therefore movement stagnates. Once movement begins, the rate of transport increases as time passes due to the increase in frequency of spatial rearrangement, which Sudd [18] suggests results from the ants' response to the reactive forces communicated through the item being transported. Although no numerical data was gathered, Sudd [17] suggests that realignment occurred more frequently than repositioning, which suggests a priority might exist between the two behaviours although sensitivity to increased frictional forces would also explain this observation.

3 Simulation

The strategies employed by ants to handle task stagnation—a condition that occurs when an item being carried gets stuck during a group transport task—can be viewed as a stored behaviour designed to overcome the difficulty. Activated as a response to increased frictional forces, the behaviours are used by ants both in group transport and during individual transport of food items. These behaviours appear to be ordered in their application. For example, Sudd [17] notes realignment seemed to occur more frequently than repositioning with the former being applied as the first response to the increase in frictional forces.

Can such an ordered set of behaviours, triggered by the lack of task progression be a useful stagnation recovery mechanism in a reactive architecture? To answer this question we will make use of our simulation environment and a box-pushing task [9].

In box-pushing the objective is to locate and move a large box too heavy to be moved by a single robot. The task will require the collective efforts of several robots pushing in a common direction (see Figure 1). We will model the forces on the box as the sum of a single robot applying a unit force at an angle to the box side. This will produce a resultant force vector and a torque applied about the box's center. If the resultant force is greater than a user defined threshold the box will translate in an



Data File: .robots

TimeStep: 404 GoalX: 409 GoalY: 170 Moved: 130.3

Figure 1: The initial configuration of the cooperative box-pushing task (left) and after 404 simulation steps in which the box has been moved 130 units upwards. The robots (circles) must locate and push the large box, which is too heavy to be moved by a single robot; therefore requiring the cooperative effort of at least 2 robots pushing on the same side.

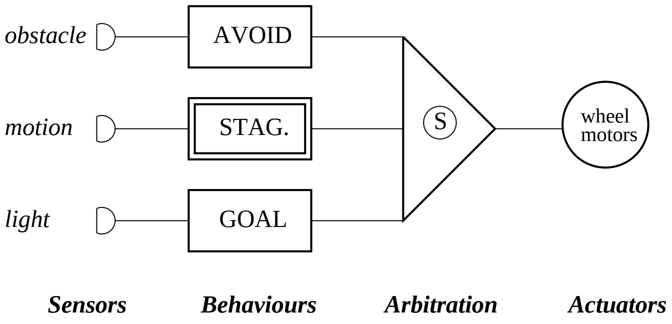


Figure 2: The box-pushing robot's behaviour based control architecture. The stagnation behaviour (labelled STAG.) is a compound behaviour consisting of several behaviour strategies designed to overcome the stagnation condition. The behaviour arbitration method is either a subsumption network (circle with S) or an Adaptive Logic Network.

XY plane. Likewise a user set torque threshold will cause rotation of the box (see Figure 4).

Each robot's control system (see Figure 2) consists of a goal behaviour, designed to direct the robot towards the box, and an avoid behaviour steering the robot away from obstacles. The goal behaviour receives input from a ring of photocells and turns the robot towards the goal by a fixed angle at each timestep using the following simple rule:

```
if the left side is brighter then move-left
else move-right
```

The avoid behaviour receives input from a left and right obstacle sensor as well as a forward pointing contact bumper. At each timestep the behaviour steers the robot away from locally perceived obstacles using the following rule:

```
if contact or obstacle-left then move-right
elseif obstacle-right then move-left
```

These simple rules are similar to those employed by Goss *et al.* [6] in their work on collective movement.

Behaviour arbitration is handled using either a subsumption network [2] or an Adaptive Logic Network (ALN) [10]. By themselves, these behaviours allow the collection of robots to locate and push the box. If an equal distribution of robots push the box the resulting net force is insufficient to overcome the weight of the box and the system stagnates. To overcome this condition, a compound behaviour is added which consists of a set of stagnation recovery behaviours (see Figure 3). The stagnation recovery behaviours each contain a counter that is reset each time the robot moves. Each counter has a

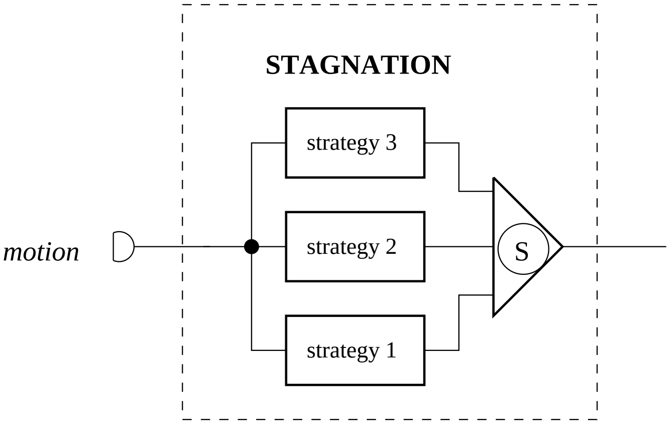


Figure 3: The compound stagnation behaviour consists of one or more (3 illustrated) behaviour strategies designed to overcome the stagnation condition.

threshold that can be set, and if reached (i.e. a timeout) the behaviour is activated. Using thresholds, behaviours can be ordered. For example, a realignment behaviour with a time threshold $T_1 = x$ where x is some constant, and a repositioning behaviour with a time threshold of $T_2 = 4x$ will take 4 times as much time before being invoked. As long as the robot is moving the behaviour does not become active since its threshold counter is constantly being reset to zero. Since the behaviours alter the robot's orientation or position by a small random amount, the robots behave asynchronously.

In the box-pushing task stagnation refers to any configuration in which robots are in contact with the box and the box is not moving. For example, consider a minimum configuration with two robots on the same side pushing with a unit force at an orientation of 45° and the weight of the box set at 1.50 Figure 5. The resulting force calculations (see Figure 4) using a unit force and equations 1 and 2 are $F_x = 1.414$ and $F_y = 0.0$. Since $F_x \leq 1.50$ stagnation has occurred.

$$F_x = \sum_{i=1}^{n} f_{ix} \tag{1}$$

$$F_y = \sum_{i=1}^{n} f_{iy} \tag{2}$$

$$f_{ix} = cos(\theta_R) \tag{3}$$

$$f_{iy} = sin(\theta_R) \tag{4}$$

Once the realignment timeout threshold is reached the behaviour changes the direction of the applied force. Since the realignment is random this may increase or decrease the resultant force. Figure 6 shows a simulation in which

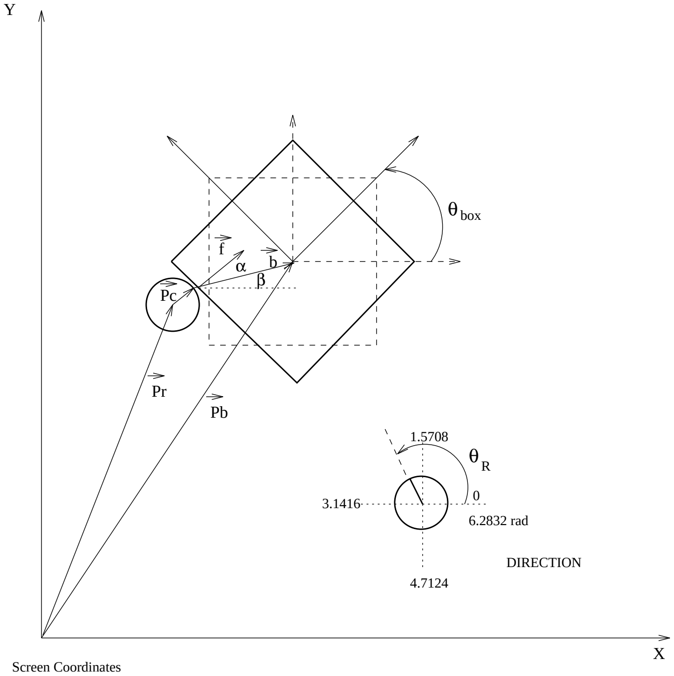
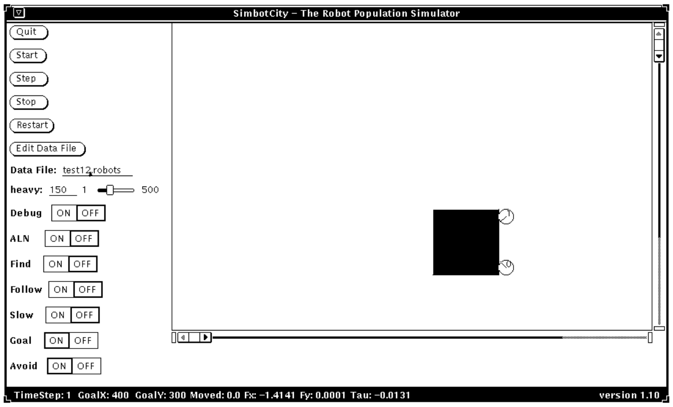


Figure 4: The model used to calculate the box force vector.



Data File: test12.robots

heavy: 150 1 500

Debug ON [✓ OFF]

ALN ON [✓ OFF]

Find ON [✓ OFF]

Follow ON [✓ OFF]

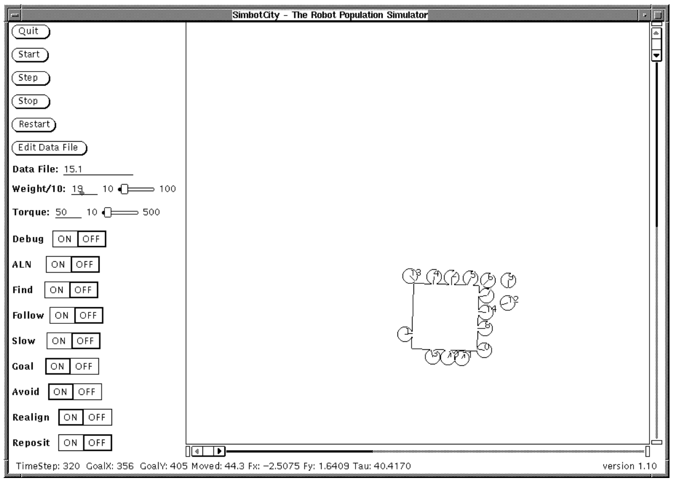
Slow ON [✓ OFF]

Goal [✓ ON] OFF

Avoid [✓ ON] OFF

TimeStep: 1 GoalX: 400 GoalY: 300 Moved: 0.0 Fx: –1.4141 Fy: 0.0001 Tau: –0.0131 version 1.10

Figure 5: A minimal stagnation condition in which each robot is applying a unit force at 45° resulting in a net force of 1.414 which is insufficient to move the box.



Data File: 15.1
Weight/10: 19 10 100
Torque: 50 10 500
Debug ON OFF
ALN ON OFF
Find ON OFF
Follow ON OFF
Slow ON OFF
Goal ON OFF
Avoid ON OFF
Realign ON OFF
Reposit ON OFF
TimeStep: 320 GoalX: 356 GoalY: 405 Moved: 44.3 Fx: −2.5075 Fy: 1.6409 Tau: 40.4170
version 1.10

Figure 6: A stagnation condition in which realignment was sufficient to reestablish movement. In this case the realignment behaviour becomes active and changes the angle at which the robot applies the force.

the system of 15 robots stagnates after having already moved the box from its initial position. In this case realignment was sufficient to resolve the stagnation and move the box 200 units.

For cases where realignment is not sufficient to establish box movement the repositioning behaviour becomes active. In this case the repositioning behaviour causes the robot to assume a new position on the box. Repositioning is also random and has the potential of generating a more drastic change in force on the box.

Figure 7 is a plot of the results of four experiments testing the stagnation recovery behaviours. The controller tested in the first experiment (labelled *goal & avoid*) uses the goal and avoid behaviors. The task is to move the box 200 units from its initial position within 2000 simulation timesteps. The plot shows the percentage of successful task execution versus the number of robots out of 25 trials. Each trial uses a different initial configuration of robots with random placement in the environment. The second experiment (labelled *realign*) adds the realignment stagnation behaviour to the previous controller. This has the effect of increasing the task success rate. The third experiment (labelled *reposition*) adds the repositioning stagnation behaviour to the first controller (*goal & avoid*). This behaviour‘s timeout threshold is set to be 4 times that of the realignment behaviour and by itself is not as effective at resolving the stagnation condition when the number of robots is low[1]. The last experiment adds both the realignment and repositioning behaviours and only affects the case of 5 robots resulting in an averaging effect of the separately applied cases.

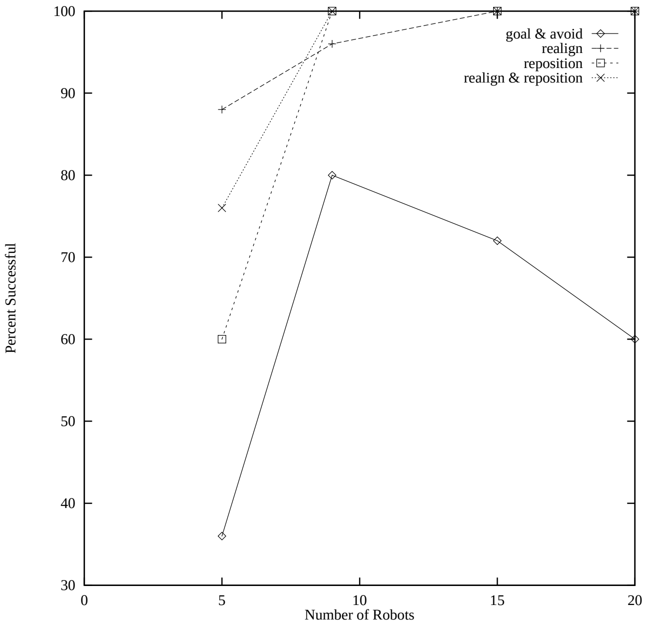


Figure 7: The results of four different controllers on the box-pushing task. Each controller was tested for 25 trials with the percentage of successful tasks shown as a function of the number of robots. Success was defined as pushing the box 200 units from its initial position within 2000 timesteps.

Figure 8 shows the average number of timesteps of successful trials as a function of the number of robots. In general the combination of realignment and repositioning decreases the number of steps to complete the task and slows the sharp increase in execution time as the number of robots increase. Although the first controller (*goal & avoid*) takes on average fewer steps for the cases of 5, 9, and 15 robots Table 1 (and Figure 7) show they were only successful 30, 80 and 72 percent of the time. All trials taken are for a specific box weight and it is still yet to be shown whether this result generalizes for varying box weights and sizes.

By designing behaviours that require the positive feedback provided by the task's successful progression we create a mechanism that may prove useful for detecting and correcting stagnation conditions in the task's execution. Instead of using positive feedback to reinforce a behaviour's activation we are using it to suppress it. For example, the GOAL behaviour is activated when its sensor locates the goal (in this case a brightly lit box). On the other hand, a wheel motion sensor is used to keep the stagnation behaviours unactivated as long as wheel motion is present, thus suppressing the behaviour. Used in this

---

[1] The number of robots used is very task dependent and for the case of box-pushing depends on the size of the box.

manner, positive feedback may both encourage successful task-achieving behaviour as well as provide a mechanism for dealing with the nonproductive activity characterized by a lack of progress.

Control Behaviours	*Number of Robots Participating in Task*			
	5	9	15	20
goal & avoid	36%	80%	100%	100%
realign	88%	96%	100%	100%
reposition	60%	100%	100%	100%
realign & reposition	76%	100%	100%	100%

Table 1: Percentage of successful times taken to move the box 200 units.

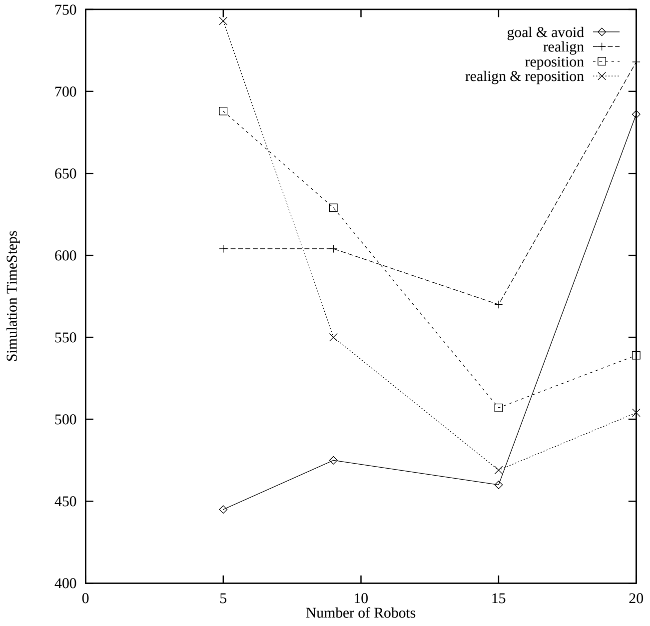


Figure 8: The results of four different controllers on the box-pushing task. Shown are the average number of steps taken to complete the task of moving the box 200 units from its initial position as a function of the number of robots.

4 Discussion

Designing intelligent robots is an elusive scientific pursuit whose solution promises to change the way we use technology in a fundamental way. Multi-robot systems are an alternate approach being explored at both the macro level on the factory floor and the micro level in future applications of micromachine technology. Accomplishing useful tasks with such a collective system has been shown by Nature's social insects to be a feasible approach using decentralized control.

Controlling a collection of robots in a decentralized manner without explicit communications between the robots requires each robot to execute control decisions based on locally sensed information only. If a robot's control mechanism is based on a sensory-response reactive architecture then being able to sense progress as well as a lack of it is mandatory for the collective system to exhibit a goal directed behaviour. When the collective system stagnates, failing in its task-achieving behaviour, then the responsibility for sensing this condition must fall to the individual robot, and a recovery method must be embedded in the control architecture of the robot.

Just as progress may be used to activate a behaviour, a lack of it may serve to activate recovery behaviours. In the box-pushing task both realignment and repositioning were shown to be useful stagnation recovery behaviours. In our previous system of 5 box-pushing robots [8] these stagnation resolution methods were not present and the system would stagnate if an equal distribution of forces occurred around the box. Both methods make use of random changes in either orientation or position. We are currently exploring a more informed change in force based on the orientation of the box.

In order to test the stagnation recovery behaviours on real robots we are constructing a system of 10 microrobots each equipped with a box contact sensor (see Figure 9). The box contains a bright white light source and will be sensed as a combination of the box contact sensor and a light sensor value. If both these conditions are true and no robot wheel motion occurs, the robot's stagnation recovery threshold counters will increment. Once the threshold is reached the stagnation recovery behaviour will be activated. The system's box-pushing behaviour will be used to create a transport behaviour so the system may move a box to a given location, a task useful in a material handling system.

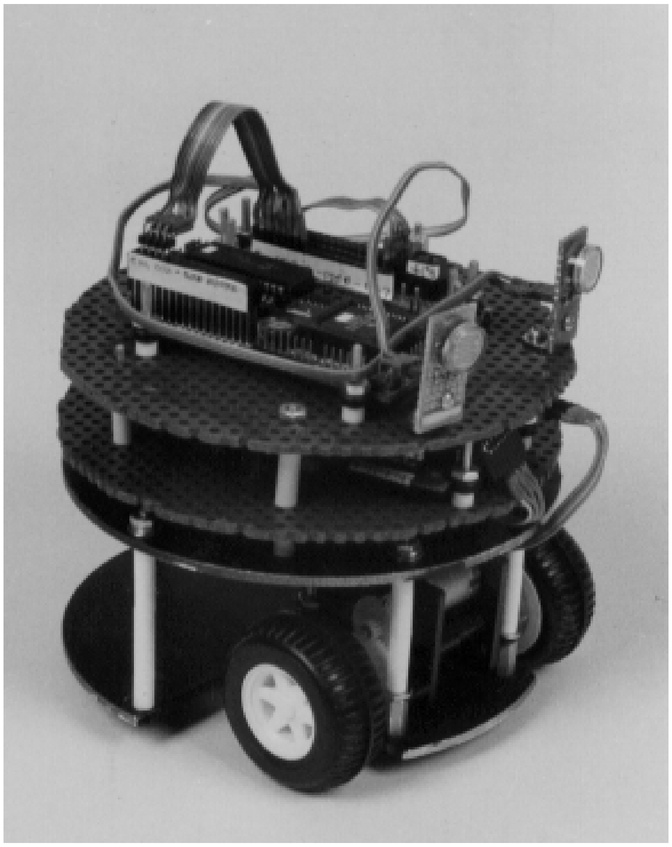


Figure 9: New prototype box-pushing robot.

Stagnation recovery behaviours are one additional mechanism useful in our reactive approach to task-achieving collective robotics. However, such a mechanism is unable to identify the other form of stagnation (i.e. cyclic behaviours) in the system since a repetition of the same motion commands does not necessarily imply a lack of progress. Detecting these cycles will help eliminate another hurdle to achieving useful tasks collectively, and we are currently investigating this important issue.

References

[1] Hajime Asama, Maki K. Habib, Isao Endo, Koichi Ozaki, Akihiro Matsumoto, and Yoshiki Ishida. Functional distribution among multiple mobile robots in an autonomous and decentralized robot system. In *1991 IEEE International Conference on Robotics and Automation*, pages 1921–1926, 1991.

[2] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, March 1986.

[3] Philippe Caloud, Wonyun Choi, Jean-Claude Latombe, Claude Le Pape, and Mark Yim. Indoor automation with many mobile robots. In *1990 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 67–72, 1990.

[4] Keith L. Doty and Ronald E. Van Aken. Swarm robot materials handling paradigm for a manufacturing workcell. In *1993 IEEE International Conference on Robotics and Automation*, pages 778–782, 1993. Vol. 1.

[5] Nigel R. Franks. Teams in social insects: Group retrieval of prey by army ants. *Behavioral Ecology and Sociobiology*, 18:425–429, 1986.

[6] S. Goss, J. L. Deneubourg, R. Beckers, and J.L. Henrotte. Recipes for collective movement. In *Proccedings of the Second European Conference on Artificial Life*, pages 400–410, 1993.

[7] Yoshio Kawauchi, Makoto Inaba, and Toshio Fukuda. A principle of distributed decision making of cellular robotic system (cebot). In *1993 IEEE International Conference on Robotics and Automation*, pages 833–838, 1993. Vol. 3.

[8] C. Ronald Kube and Hong Zhang. Collective robotic intelligence. In *Second International Conference on Simulation of Adaptive Behavior*, pages 460–468, December 7-11 1992.

[9] C. Ronald Kube and Hong Zhang. Collective robotics: From social insects to robots. *Adaptive Behavior*, 2(2):189–219, 1994.

[10] C. Ronald Kube, Hong Zhang, and Xiaohuan Wang. Controlling collective tasks with an ALN. In *1993 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 289–293, 1993.

[11] Maja J. Mataric. Minimizing complexity in controlling a mobile robot population. In *1992 IEEE International Conference on Robotics and Automation*, pages 830–835, 1992.

[12] M. W. Moffett. Cooperative food transport by an asiatic ant. *National Geographic Research*, 4(3):386–394, 1988.

[13] Fabrice R. Noreils. Coordinated protocols: An approach to formalize coordination between mobile robots. In *1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 717–724, 1992.

[14] Lynne E. Parker. Designing control laws for cooperative agent teams. In *1993 IEEE International Conference on Robotics and Automation*, pages 582–587, 1993. Vol. 3.

[15] Karansher Singh and Kikuo Fujimura. Map making by cooperating mobile robots. In *1993 IEEE International Conference on Robotics and Automation*, pages 254–259, 1993. Vol. 2.

[16] Daniel J. Stilwell and John S. Bay. Toward the development of a material transport system using swarms of ant-like robots. In *1993 IEEE International Conference on Robotics and Automation*, pages 766–771, 1993. Vol. 1.

[17] J. H. Sudd. The transport of prey by an ant, *pheidole crassindoa*. *Behaviour*, 16(3-4):295–308, 1960.

[18] J. H. Sudd. The transport of prey by ants. *Behaviour*, 25(3-4):234–271, 1965.

[19] P. K. C. Wang. Navigation strategies for multiple autonomous mobile robots moving in formation. *Journal of Robotic Systems*, 8(2):177–195, 1991.

[20] Stewart W. Wilson. The animat path to AI. In *First International Conference on Simulation of Adaptive Behavior*, pages 15–21. MIT Press, 1990.