Broadcast of Local Eligibility for Multi-Target Observation

Barry Brian Werger and Maja J Matarić

USC Robotics Research Lab, Los Angeles, CA 90089-0781

Abstract. While numerous researchers have investigated group behavior of robots which are each controlled in a behavior-based manner, none have yet thoroughly investigated the possibilities of extending the *port-arbitrated behavior* (PAB) paradigm across networks of robots. We present an extension to the well-defined PAB techniques of behavioral interaction which provides standard abstractions for nessaging, inhibition, and suppression over IP networks. The Broadcast of Local Eligibility is a general technique built from these abstractions that allows fully-distributed, flexible team coordination. We present a BLE approach to the CMOMMT multi-target observation problem, implemented on a team of physical robots.

1 Introduction

While numerous researchers have investigated group behavior of robots which are each controlled in a behavior-based manner, none have yet thoroughly investigated the possibilities of extending the port-arbitrated behavior [12] paradigm across networks of robots. While it has often been hypothesized that there need be no distinction between inter-robot and inter-behavior communication e.g., [2], to our knowledge no previous system has provided standard tools that allow port-based messaging, suppression, and inhibition between behaviors on separate networked robots.

Our intention is to demonstrate that behavior-based systems restricted to well-defined port-arbitrated interactions can scale to higher levels of competence than is generally assumed. Specifically, we show that when the port-arbitration paradigm is extended across networks, the resulting systems are able to dynamically reconfigure themselves in order to efficiently allocate resources in response to task constraints, environmental conditions, and system resources. We introduce the Broadcast of Local Eligibility (BLE) as a general tool for coordination between robots, and then demonstrate its application to the CMOMMT [8] multi-target observation task.

PAB: Port-Arbitrated Behavior-Based Control In PAB systems, controllers are written in terms of *behaviors*, which are groups of concurrent processes that share a public interface. This interface is composed of *ports*, which are registers that each hold a single data item (e.g., an integer, float, string, or complex data structure).

Ports in different behaviors are linked together by *connections*, which are unidirectional data paths between a *source port* and a *destination port*. A port can have any number of incoming and outgoing connections. When data is written to a port, either directly from a process within the behavior or indirectly through a connection, it is generally propagated along all of that port's outgoing connections. We say "generally," because data flow can be modified by special connections; whenever a message m is propagated along a connection $C_{s,d}$ from port s to port d, if $C_{s,d}$ is:

- *Normal*, then m is written to d, and is propagated along all of d's outgoing connections
- *Suppressive*, then m is not written to d, and for a specified period no incoming connections will be able to write to d
- *Inhibitory*, then m is not written to d, and for a specified period no messages will be propagated out from d
- *Input Overriding*, then d is *suppressed*, except that messages arriving along $C_{s,d}$ (including m) are written to d and propagated as normal
- *Output Overriding*, then d is *inhibited*, except that messages arriving along $C_{s,d}$ (including m) are propagated along all $C_{d,x}$ as normal

It is through these mechanisms of suppression and inhibition that subsumption hierarchies, as well as other forms of arbitration, can be efficiently and intuitively implemented. Since connections are external to the behaviors, behavior code is easily re-usable, and interaction between behaviors can be modified dynamically. The port abstraction enforces a data-driven approach to programming that "grounds" computation in sensor readings and effector actions. The PAB approach allows a clean, uniform interface between system components (behaviors) at all levels that abstracts away many issues of timing and communication; the "black boxes" of behaviors may contain reactive mappings or deliberative planners. While our research focuses on non-deliberative approaches, we believe that PAB interaction between system components can help reduce the complexity of the components themselves, whatever their type.

Ayllu: Port-Arbitration over IP Networks Ayllu, a C-based language for behavior-based control [12] has been developed to facilitate implementation of distributed PAB algorithms. Connections between ports, of all the types listed in Section 1, can be made either locally or over IP, and can be specified to broadcast. In addition, Ayllu adds specialized port types, such as *max ports* which filter arriving messages for maximum values. Max ports are sufficient to build systems that are arbitrarily scalable.

2 Broadcast of Local Eligibility

We now introduce our Broadcast of Local Eligibility (BLE) approach to multi-robot coordination. The BLE mechanism involves a comparison of lo-

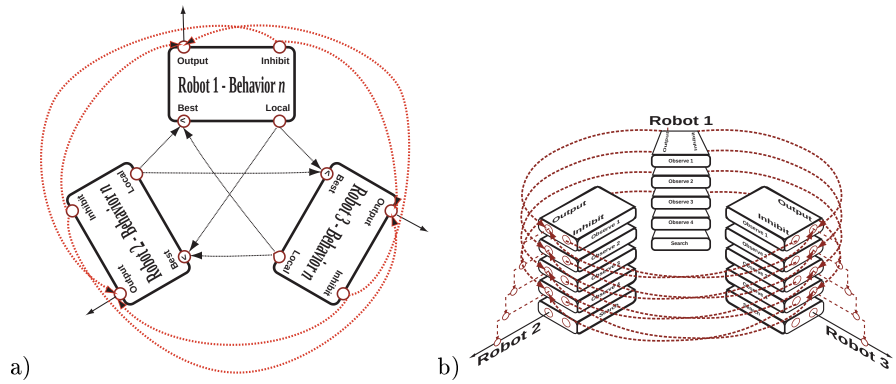


Fig. 1. *a) Cross-Inhibition*: A cross-inhibited peer group. *b) Cross-Subsumption*: The structure of a cross-cubsumptive system. Some lines are omitted for clarity; each “layer” is connected as a).

cally determined eligibility with the best eligibility calculated by a peer behavior on another robot. When a robot’s local eligibility is best for some behavior B_n which performs task T_n, it inhibits the peer behaviors (that is, behaviors B_n) on all other robots, thereby “claiming” task T_n. Since this inhibition is an active process, failure of a robot which has claimed a task results in the task being immediately “freed” for potential takeover by another robot.

Since BLE is based on broadcast messages and receiving ports that filter their input for the “best” eligibility (see Section 2.1), BLE-based systems are inherently scalable. Up to the limit of communication bandwidth, any number of BLE-enabled robots can be added to a system. As we will also see in Section 2.2, BLE allows heterogeneous robots to efficiently allocate themselves to appropriate tasks.

2.1 BLE-Enabled Behaviors

BLE action selection requires that each BLE-arbitrated behavior include three ports named *Local*, *Best*, and *Inhibit*(see Figure 1a). Useful behaviors will generally have additional ports for task-related input and output. We generically refer to the BLE-arbitrated output of a behavior as *Output*, though the actual output may be through any number of ports of arbitrary name. The *Best*port is a emphmax port, accepting only values that are larger than the its current value.

2.2 Cross-Inhibition of Behaviors

Cross-inhibition refers to the process of arbitration between *peer behaviors,* instances of the same BLE behavior on different robots. Given that there is some behavior instance B_n (which performs task T_n) on each robot, cross-inhibition results in the selection of at most a single robot to perform T_n. The selected robot is the one that is most eligible (according to local criteria) for the task. There may be multiple sets of cross-inhibiting behaviors active at the same time; Section 2.3 below discusses one manner in which local arbitration between different cross-inhibited behaviors can take place.

As illustrated in Figure 1a, the *Local*port of each robot's behavior B_n broadcasts a locally-computed eligibility estimate to the *Best*port of each other robot's behavior B_n. Each *Best*port maintains the maximum of the eligibility messages it has received in the current cycle. Whichever robot has a local eligibility better than or equal to the *Best*it receives writes to its *Inhibit*port, causing inhibition of behavior B_n in the other robots.

Cross inhibition is particularly well-suited to heterogeneous systems, in which not all robots are able to perform all tasks. Robots in which some behavior B_n is not instantiated will naturally never inhibit B_n in other robots and claim T_n; thus if robots locally instantiate only behaviors appropriate to their capabilities nothing more needs to be done in order to assign heterogeneous robots to appropriate tasks. If the local arbitration (see Section 2.3 below) gives priority to the tasks each robot is specialized for, then this assignment of robots to tasks should be very efficient.

2.3 Cross-Subsumption

Cross-inhibition arbitrates only between peer behaviors on different robots; some local mechanism must arbitrate between different behaviors on the same robot. We believe that simple subsumption, when combined with BLE, is sufficient for flexible, scalable, and robust team cooperation in many tasks. We call the combination of cross-inhibition and local subsumption *cross-subsumption*.

In cross-subsumption, each robot has a local subsumption hierarchy. Each layer of this hierarchy may be cross-inhibited (as in Figure 1a), resulting in a system similar to the one diagrammed in Figure 1b. As a result, each robot is controlled by its behavior B_n which has the highest priority of any behavior which is generating output. A behavior that is not generating output fails to do so either because its current input is unsuitable for the task (e.g., some necessary object is not in the field of view), or because its output is cross-inhibited. Thus, each robot will claim the highest-priority task that it is most suitable for.

3 The CMOMMT Task

We have tested our BLE approach on a multi-target observation task known as CMOMMT (Cooperative Multi-robot Observation of Multiple Moving Targets) introduced by [8], and a prioritized variation that we call W-CMOMMT.

CMOMMT is an NP-hard problem that requires strong cooperation [7] for good performance. It has the benefit of simple formulation and analysis, and implemented systems for comparison.

3.1 Definition of CMOMMT

Our version of the CMOMMT problem is defined as follows. Given S : a bounded, enclosed region; R : a team of m robots with noisy, limited sensors; and $O(t)$: a set of n targets $o_j(t)$ such that $In(o_j(t), S)$ is true, where $In(o_j(t), S)$ means that target $o_j(t)$ is within S at time t, define an $m \times n$ matrix $A(t)$ where

$$a_{ij}(t) = \begin{cases} 1, \text{if robot } r_i \text{ is observing target } o_j \\ \quad \text{at time } t \\ 0, \, otherwise \end{cases} \tag{1}$$

(a robot is *observing* a target when the target is in the robot's field of view and within a certain distance)

and define a logical OR operator over a vector H :

$$\bigvee_{i=1}^{k} h_i = \begin{cases} 1, \text{if there exists an } i \text{ such that } h_i = 1 \\ 0, \, otherwise \end{cases} \tag{2}$$

The goal of the CMOMMT is then to maximize

$$Observation = \frac{\sum_{t=1}^{T} \sum_{j=1}^{m} \bigvee_{i=1}^{r} a_{ij}(t)}{t \times m} \tag{3}$$

that is, to maximize the time during which each target in S is being observed by at least one robot. We assume that the area covered by the sensors of the robots is much smaller than the total area to be monitored and that targets move slower than the robots. The original formulation of the problem [8] assumes that robots share a known global coordinate system; we replace this with the assumption that the robots can visually distinguish each target from the others. Thus our formulation focuses on *task space* where Parker's formulation, predictive tracking, and *local force-vector* algorithm [8] tend to be more oriented towards physical space.

We also introduce a prioritized version of the problem which we call Weighted CMOMMT, or W-CMOMMT. Given: W : a vector of weights such that w_i reflects the priority of target o_i, the goal of W-CMOMMT is to maximize

$$W\text{-}Observation = \frac{\sum_{t=1}^{T} \sum_{j=1}^{m} w_j \bigvee_{i=1}^{r} a_{ij}(t)}{t \times m \times \sum_{v=1}^{m} w_v} \tag{4}$$

4 Experiments

We have implemented controllers for CMOMMT on a team of three ActivMedia Pioneer 2DX robots. These are differentially-steered wheeled bases with on-board sonar (for obstacle avoidance) and vision (for identifying and tracking targets). The video cameras have a 45-degree field of view. Each robot is connected to a wireless ethernet LAN, and programmed using Ayllu [12].

4.1 The Experimental Environment

Current experiments take place in an 18 by 22 foot corral. Targets are colored paper cylinders which experimenters move by hand in a fixed pattern at an average speed of about 2 feet/minute. The targets all begin at one end of the enclosure, and move in a criss-cross pattern that varies from a very dispersed to a very condensed formation (see Figure 2a). Trials are run with three robots and four targets.

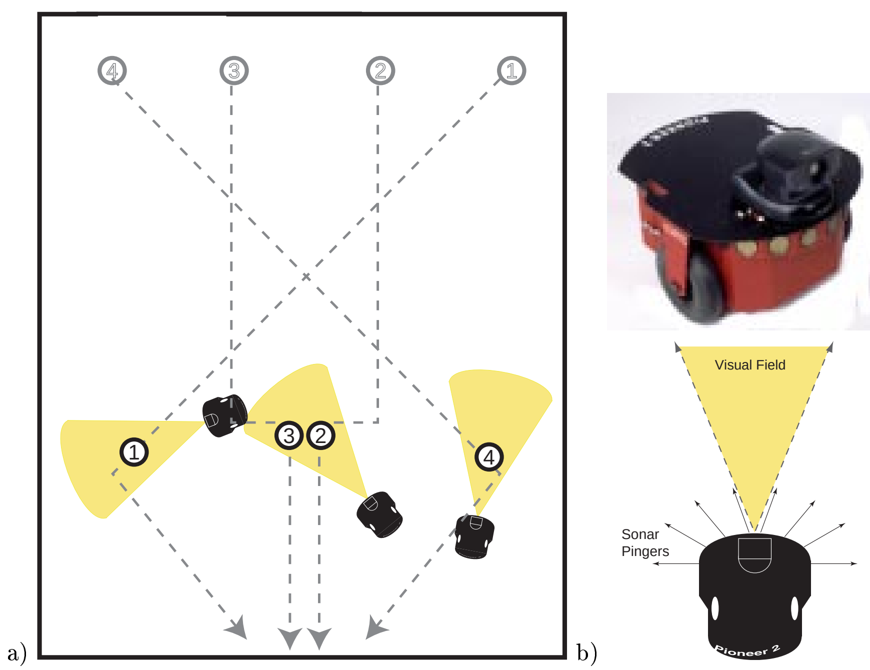


Fig. 2. a) *Experimental Environment*: The 18 by 22 foot corral. Robots are shown with observation ranges; fields of view extend further in a similar cone. Targets are numbered circles. Light grey targets and dashed lines indicate initial positions and paths of targets. b) *Robotic Testbed*: Three Pioneer 2DX robots.

4.2 Robot Behaviors

Four controllers have been implemented for comparison in CMOMMT: a BLE controller, a local greedy controller, a local subsumption controller, and a random controller. Each of these controllers is implemented using the same behaviors, with differences in behavior arbitration.

Common Behaviors A single behavior on each robot controls translational motion to maintain a safe velocity based on the distance to sonar-detected obstacles. The task-oriented behaviors specified below only control rotational motion of the robots. Two classes of behavior are implemented:

Observer behaviors: the target-observing behaviors rotate the robot towards a specific target in its field of view. This, combined with the common velocity control behavior, causes the robot to approach a specific target and maintain a distance of approximately 1 foot. One observer behavior is instantiated for each target to be tracked. The observation range of the robots is approximately four feet, and the robots are able to perceive targets up to fifteen feet away, depending on lighting conditions in different parts of the corral.

Search behavior: the search behavior is a random wander.

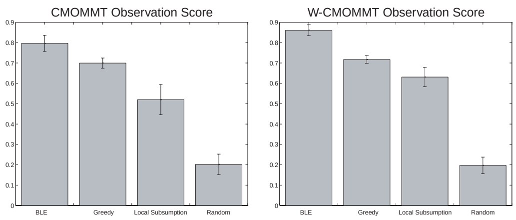
BLE Coordination: The BLE controller is a subsumption hierarchy of *Observer* behaviors, with the Target 1 observer having highest priority. Each is then joined into a cross-inhibiting peer group which consists of *Observer*s of the same target on each robot (Figure 1a), such that the controller becomes a cross-subsumption hierarchy (Figure 1b. The highest-priority behavior that is not cross-inhibited controls the robot - that is, each robot approaches and tracks the highest-priority target it sees that is not being observed by another robot.

The local evaluation function for each *Observer* behavior is proportional to the height of its associated target in the visual field (an approximation of distance). It favors observation of multiple targets by increasing for each additional target viewed in observation range.

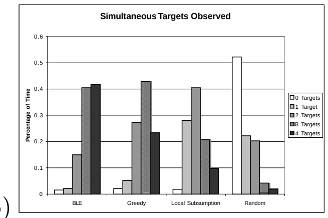
The wander behavior is active when all other behaviors are either cross-inhibited or unable to perceive any targets.

Local Subsumption Only The Local Subsumption (LS) controller is the same as the BLE controller, but connections are not made across the peer groups so that no cross-inhibition takes place. The robot approaches and tracks the highest-priority target it sees, or wanders if it sees no targets.

Local Greedy The Local Greedy (Greedy) controller has neither cross-inhibition nor local subsumption; instead, the behavior with the highest evaluation function controls the robot. The robot approaches and tracks whatever



a)



b)

Fig. 3. a) *Average Observation and W-Observation scores by algorithm.* Error bars span 2 standard deviations. b) *Simultaneous Observation* The percentage of time at which 4, 3, 2, 1, or no targets were observed, averaged over trials.

target is most salient (as influenced by perceptual uncertainty) in the field of view, or wanders if no target is perceived.

Random In the Random controller, only the random wander behavior is active, at all times.

4.3 Results

Five trials of approximately 12 minutes each were run for each of the BLE, Greedy, and LS controllers; two trials of the Random controller were run for a baseline.

The most important measures are of course the *Observation* and *W-Observation* metrics of Section 3.1. As seen in Figure 3a, on CMOMMT the BLE approach, averaging 0.7963, scored significantly higher ($p = 0.0017$ on a pairwise t-test) than the Greedy approach at 0.69940 and the LS approach at 0.51995 ($p < .0001$). Using the W-CMOMMT metric, the relative performance of BLE was even better, scoring 0.860984 to the Greedy score of 0.717251 ($p < 0.00001$) and the LS score of 0.630928.

The distribution of the robots across the targets can be clarified with information on simultaneous target coverage, illustrated in Figure 3b. On average, the BLE approach observed all four targets 41.68 percent of the time, and observed at least three targets 82.17 percent of the time. The Greedy approach averaged four targets only 23.27 percent of the time, and at least three targets 66.12 percent of the time. LS trailed with observation of four targets averaging 9.71 percent, and three or more, 30.38 percent. Thus, BLE achieved better distribution than either Greedy or LS. Surprisingly, BLE achieved marginally higher observation of the highest-priority target than Local Subsumption (95.74 vs. 95.62 percent).

It can be seen from the target motion patterns of Figure 2a that during the last third of each trial, targets 2 and 3 were consistently close enough to be observed by a single robot. While the BLE approach resulted in a stable configuration of all four targets being observed for the majority of the final third of *every trial*, neither the Greedy nor the LS approaches maintained

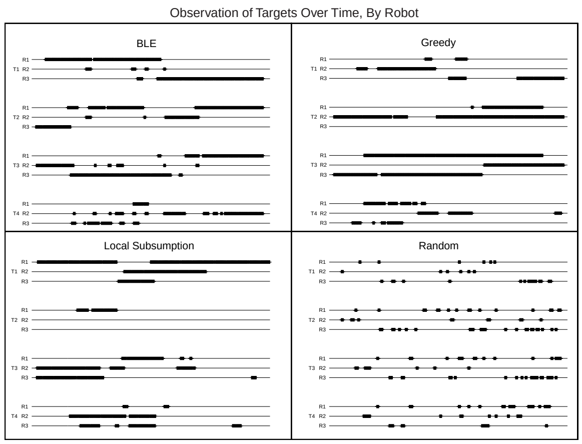


Fig. 4. *Observation Over Time, by Algorithm*

such a stable full observation in even a single trial. Figure 4 illustrates typical patterns of observation (for each algorithm we have chosen a trace of the trial which scored closest to the average). In the BLE approach, the three highest-priority targets are covered fairly constantly, although the observing robots switch off; and overlap of observation is minimal. The stable four-target observation for the last third of the trial can be seen, with robot 1 covering both targets 2 and 3. In the Greedy trace, there are clearly both larger periods of overlap and larger periods in which some targets are not covered at all. In LS, as expected, the highest-priority target was well and redundantly covered, while others were not.

In all trials, periods in which a particular robot seems not to be observing anything often reflect a blocked robot which is tracking a target, but not close enough to observe. This situation was common to Greedy and LS trials where robots often "queued up" behind other robots observing a salient target. Further, our collision avoidance, resulting only from the translational velocity control described in Section 4.2, did not deal effectively with robots approaching each other from the side, as when both were trying to get close to the same target; this resulted in occasional collisions during the LS and Greedy trials. The *task-space* separation of the BLE approach proved to be very effective in preventing both of these *physical-space* problems of interference.

Further, observation of the different approaches in action led to the realization that the BLE approach was effective in overcoming perceptual limitations of the robots. While in the Greedy and LS trials robots tended to cluster around targets that were "better perceived" (due to details of the color-tracking implementation, and environment), exacerbating the physical-space problems described above, in the BLE trials, highly visible targets were quickly observed, freeing robots to persue less-salient targets.

5 Related Work

Pioneering work with groups of behavior-based robots involved *weakly cooperative* tasks such as flocking and foraging [6] and clustering [5]. [3] modified such foraging strategies to include coordinated territorial division, and [4] investigated role-based task division in foraging systems, with and without explicit coordination. [1] discuss how communication is "foreign" to their behavior-selection methods. [9] discuss levels of abstraction that allow entire robots to be seen as single behaviors in a behavioral system, but rely on "observer" behaviors which come close to a centralized solution. The BeRoSH system [10] extends a subsumption approach to allow inter-robot communication as input to behaviors (but not for inhibition/suppression). Task performance is fully distributed, but one robot "host" must coordinate,

The ALLIANCE architecture [7] comes closest to our approach. Main differences include the need in ALLIANCE for motivational behaviors to store information about other individual robots, the lack of uniform inter-behavior communication (inter-robot communication only takes place between motivational behaviors, and cannot be arbitrated as other behavioral communication can be), and ALLIANCE's monitoring of time other robots have spent performing behaviors rather than BLE's local eligibility estimates. Motivational behaviors must be re-written (rather than re-connected) to reflect changes in system structure, and, as the gateways between robots, restrict the ways in which robots can interact. However, direct comparison of the two systems is not possible (or even useful) as Ayllu and BLE are aimed at a lower level of abstraction than ALLIANCE. It is clear that ALLIANCE could be implemented using Ayllu and BLE, and that for certain tasks it would likely be more convenient; but our interest is in examining the range of tasks that can be covered by our small set of clean, standard, "stateless" [11] language-level abstractions.

6 Conclusion

Experimentation has shown that the PAB paradigm, and BLE in particular, is able to support fully distributed, efficient coordination of teams of robots using simple and general low-level components. The resulting systems are scalable, robust, and flexible, adapting to changing environmental conditions and resource availability. Cross-subsumption can assign heterogeneous robots to tasks appropriately with no need for explicit negotiation or recognition. PAB is a principled approach, providing standard, well-defined abstractions for behavior coordination. Behavior are fully encapsulated, facilitating "bottom up" system design and testing.

References

1. T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, December 1998.
2. R. A. Brooks. *Cambrian Intelligence*. MIT Press, 1999.
3. Miguel Schneider Fontán and Maja J. Matarić. A study of territoriality: the role of critical mass in adaptive task division. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, Cape Cod, September 1996.
4. Dani Goldberg and Maja J Matarić. Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings of AAAI-97*, pages 637–642, Providence, RI, July 1996.
5. Owen Holland and Chris Melhuish. Stigmergy, self-organisation, and sorting in collective robotics. *Artificial Life*, 5:2, 2000.
6. Maja J Matarić. Designing and understanding adaptive group behavior. *Adaptive Behavior*, 4:1:51–80, December 1995.
7. L. E. Parker. Alliance: An architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14, 1998.
8. L. E. Parker. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing*, 5:5–19, 1999.
9. J. Salido, J. M. Dolan, J. Hampshire, and P. Khosla. A modified reactive control framework for cooperative mobile robots. In *Sensor Fusion and Decentralized Control in Autonomous Robotic Systems, Proceedings of SPIE 3209*, 1997.
10. Z.-D. Wang, E. Nakano, and T. Matsukawa. Designing behavior of a multiple robotic system for cooperative object manipulation. In *Proceedings of the International Symposium on Microsystems, Intelligent Materials, and Robots*, 1995.
11. Barry B. Werger. Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams. *Artificial Intelligence*, 110:293–320, 1999.
12. Barry Brian Werger. Ayllu: Distributed port-arbitrated behavior-based control. Technical Report 99-01, Ullanta Performance Robotics, 1999. Submitted to DARS 2000.