Autonomous Robots, 12, 3 (2002)

Distributed Algorithms for Multi-Robot Observation of Multiple Moving Targets

LYNNE E. PARKER

ParkerLE@ornl.gov
Center for Engineering Science Advanced Research (CESAR)
Computer Science and Mathematics Division
Oak Ridge National Laboratory
P. O. Box 2008, Mailstop 6355, Oak Ridge, TN 37831-6355 USA

Abstract. An important issue that arises in the automation of many security, surveillance, and reconnaissance tasks is that of observing the movements of targets navigating in a bounded area of interest. A key research issue in these problems is that of sensor placement — determining where sensors should be located to maintain the targets in view. In complex applications involving limited-range sensors, the use of multiple sensors dynamically moving over time is required. In this paper, we investigate the use of a cooperative team of autonomous sensor-based robots for the observation of multiple moving targets. In other research, analytical techniques have been developed for solving this problem in complex geometrical environments. However, these previous approaches are very computationally expensive – at least exponential in the number of robots — and cannot be implemented on robots operating in real-time. Thus, this paper reports on our studies of a simpler problem involving uncluttered environments — those with either no obstacles or with randomly distributed simple convex obstacles. We focus primarily on developing the on-line distributed control strategies that allow the robot team to attempt to minimize the total time in which targets escape observation by some robot team member in the area of interest. This paper first formalizes the problem (which we term *CMOMMT* for *Cooperative Multi-Robot Observation of Multiple Moving Targets*) and discusses related work. We then present a distributed heuristic approach (which we call *A-CMOMMT*) for solving the *CMOMMT* problem that uses weighted local force vector control. We analyze the effectiveness of the resulting weighted force vector approach by comparing it to three other approaches. We present the results of our experiments in both simulation and on physical robots that demonstrate the superiority of the *A-CMOMMT* approach for situations in which the ratio of targets to robots is greater than 1/2. Finally, we conclude by proposing that the CMOMMT problem makes an excellent domain for studying multi-robot learning in inherently cooperative tasks. This approach is the first of its kind for solving the on-line cooperative observation problem and implementing it on a physical robot team.

1. Introduction

Many security, surveillance, and reconnaissance tasks require autonomous observation of the movements of targets navigating in a bounded area of interest. A key research issue in these problems is that of sensor placement — determining where sensors should be located to maintain the targets in view. In the simplest version of this problem, the number of sensors and sensor placement can be fixed in advance to ensure adequate sensory coverage of the area of interest. However, in more complex applications, a number of factors may prevent fixed sensory placement in advance. For example, there may be little prior information on the location of the area to be observed, the area may be sufficiently large that economics prohibit the placement of a large number of sensors, the available sensor range may be limited, or the area may not be physically accessible in advance of the mission. In the general case, the combined coverage capabilities of the available robot sensors will be insufficient to cover the entire terrain of interest. Thus, the above constraints force the use of multiple sensors dynamically moving over time.

There are many application motivations for studying this problem. Targets to be tracked in a multi-robot or multi-agent context include other mobile robots, items in a warehouse or factory, people in a search and rescue effort, and adversarial targets in surveillance and reconnaissance situations. Even medical applications, such as moving cameras to keep designated areas (such as particular tissue) in continuous view during an operation, require this type of technology [18].

In this article, we investigate the use of a cooperative team of autonomous sensor-based robots for applications in this domain. Our primary focus is on developing the distributed control strategies that allow the team to attempt to minimize the total time in which targets escape observation by some robot team member in the area of interest, given the locations of nearby robots and targets. Because we are interested in real-time solutions to applications in unknown and dynamic environments, we do not view the problem from the perspective of computing geometric visibility, which is largely a problem of planning from a known world model. Instead, we investigate the power of a weighted force vector approach distributed across robot team members in simple, uncluttered environments that are either obstacle-free or have a random distribution of simple convex obstacles. In other work [32], we describe the implementation of this weighted force vector approach in our ALLIANCE formalism ([26], [30]) for fault tolerant multi-robot cooperation. In this article, we focus on the analysis of this approach as it compares to three other approaches: (a) fixed robot positions (*Fixed*), (b) random robot movements (*Random*), and (c) *non-weighted* local force vectors (*Local*).

Of course, many variations of this dynamic, distributed sensory coverage problem are possible. For example, the relative numbers and speeds of the robots and the targets to be tracked can vary, the availability of inter-robot communication can vary, the robots can differ in their sensing and movement capabilities, the terrain may be either enclosed or have entrances that allow targets to enter and exit the area of interest, the terrain may be either indoor (and thus largely planar) or outdoor (and thus 3D), and so forth. Many other subproblems must also be addressed, including the physical tracking of targets (e.g., using vision, sonar, infrared, or laser range), prediction of target movements, multi-sensor fusion, and so forth. Thus, while our ultimate goal is to develop distributed algorithms that address all of these problem variations, we first focus on the aspects of distributed control in robot teams with equivalent sensing and movement capabilities working in an uncluttered, bounded area.

The following section defines the multi-target observation problem of interest in this paper, which we term *CMOMMT* for *Cooperative Multi-Robot Observation of Multiple Moving Targets*. We then discuss related work in this area, followed by a description of the distributed, weighted local force vector approach, which we call *A-CMOMMT*. Next, we describe and analyze the results of the *A-CMOMMT* approach in both simulation and on physical robots, compared to three other policies for cooperative target observation – *Local*, *Fixed*, and *Random*. We present a proof-of-concept implementation of our approach on a team of four Nomad robots. Finally, we discuss the merits of the CMOMMT problem as a general domain for multi-robot learning involving inherently cooperative tasks.

2. *CMOMMT* problem description

The problem of interest in this paper — *Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT)* — is defined as follows. Given:

$\mathcal{S}$: a two-dimensional, bounded, enclosed spatial region

$\mathcal{V}$: a team of m robot vehicles, $v_i, i = 1, 2, ...m$, with 360^0 field of view observation sensors that are noisy and of limited range

$\mathcal{O}(t)$: a set of n targets, $o_j(t)$, $j = 1, 2, ..., n$, such that target $o_j(t)$ is located within region $\mathcal{S}$ at time t

We say that a robot, v_i, is *observing* a target when the target is within v_i's *sensing range* (defined explicitly below).

Define an $m \times n$ matrix $B(t)$, as follows:

$$B(t) = [b_{ij}(t)]_{mxn} \text{ such that } b_{ij}(t) = \begin{cases} 1 & \text{if robot } v_i \text{ is } \textit{observing} \text{ target } o_j(t) \text{ in } \mathcal{S} \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

Then, the goal is to develop an algorithm, which we call *A-CMOMMT*, that maximizes the following metric A:

$$A = \sum_{t=1}^{T} \sum_{j=1}^{n} \frac{g(B(t), j)}{T}$$

where:

$$g(B(t), j) = \begin{cases} 1 & \text{if there exists an } i \text{ such that } b_{ij}(t) = 1 \\ 0 & \text{otherwise} \end{cases}$$

under the assumptions listed below. In other words, the goal of the robots is to maximize the average number of targets in $\mathcal{S}$ that are being observed by at least one robot throughout the mission that is of length T time units. Note that we do not assume that the membership of $\mathcal{O}(t)$ is known in advance. Also note that if we later wanted to use an adaptive algorithms that improved its performance over time, this metric could be changed to sum over a recent time window, rather than the entire mission. This would eliminate the penalty for the early performance of processes that improve over time.

In addressing this problem, we define $sensor\_coverage(v_i)$ as the region visible to robot v_i's observation sensors, for $v_i \in \mathcal{V}$. Then we assume that, in general, the maximum region covered by the observation sensors of the robot team is much less than the total region to be observed. That is,

$$\bigcup_{v_i \in \mathcal{V}} sensor\_coverage(v_i) \ll \mathcal{S}.$$

This implies that fixed robot sensing locations or sensing paths will not be adequate in general, and that instead, the robots must move dynamically as targets appear in order to maintain their target observations and to maximize the coverage.

We further assume the following:

- The robots have a broadcast communication mechanism that allows them to send (receive) messages to (from) each other within a limited range. The range of communication is assumed to be larger than the sensing range of the robots, but (potentially) smaller than the diameter of $\mathcal{S}$. This communication mechanism will be used only for one-way communication. Further, this communication mechanism is assumed to have a bandwidth of order $O(mn)$ for m robots and n targets[1].
- For all $v_i \in \mathcal{V}$ and for all $o_j(t) \in \mathcal{O}(t)$, $max\_vel(v_i) > max\_vel(o_j(t))$, where $max\_vel(a)$ gives the maximum possible velocity of entity a, for $a \in \mathcal{V} \cup \mathcal{O}(t)$. This assumption allows robots an opportunity to collaborate to solve the problem. If the targets could always move faster, then they could always evade the robots and the problem becomes trivially impossible for the robot team (i.e., assuming "intelligent" targets).
- The robot team members share a known global coordinate system[2].

In some situations, the observation sensor on each robot is of limited range and is directional (e.g., a camera), and can only be used to observe targets within that sensor's field of view. However, in this article, we report the results that assume an omni-directional 2D sensory system (such as a ring of cameras or sonars, the use of a global positioning system, or a single omnidirectional camera such as those recently prototyped by SRI, CMU/Columbia, etc.), in which the robot sensory system is of limited range, but is available for the entire 360^o around the robot.

3. Related work

Research related to the multiple target observation problem can be found in a number of domains, including art gallery and related problems, multi-target tracking, and multi-robot surveillance tasks. Nearly all of the previous work in this area involves the development of centralized algorithms for complex geometric environments using ideal sensors. While these previous approaches provide theoretically precise solutions to complex geometrical problems, they are extremely computationally expensive (at least exponential in the number of robots) and do not scale well to numbers of sensors and targets greater than one or two. The previous solutions also do not provide a on-line, real-time solutions to the cooperative observation problem. Our work differs in that, while we do not solve for complex geometric environments, we do provide a heuristic on-line solution that works in simple environments, and that scales well to larger numbers of targets and robots in the midst of sensor and effector errors.

While a complete review of the previous research is not within the scope of this paper, we will briefly outline the previous work that is most closely related to the topic of this paper. The work most closely related to the *CMOMMT* problem falls into the category of the *art gallery* and related problems [25], which deal with issues related to polygon visibility. The basic art gallery problem is to determine the minimum number of guards required to ensure the visibility of an interior polygonal area. Variations on the problem include fixed point guards or mobile guards that can patrol a line segment within the polygon. Several authors have looked at the static placement of sensors for target tracking in known polygonal environments (e.g., [8]). These works differ from the *CMOMMT* problem, in that our robots must dynamically shift their positions over time to ensure that as many targets as possible remain under surveillance, and their sensors are noisy and of limited range.

Sugihara *et al.* [36] address the *searchlight scheduling problem*, which involves searching for a mobile "robber" (which we call *target*) in a simple polygon by a number of fixed searchlights, regardless of the movement of the target. They show that the problem of obtaining a search schedule for an instance having at least one searchlight on the polygon boundary can be reduced to that for instaces having no searchlight on the polygon boundary. They also present a necessary and sufficient condition for the existence of a search schedule for exactly two searchlights in the interior. They do not address issues of computational complexity.

Suzuki and Yamashita [37] address the *polygon search* problem, which deals with searching for a mobile target in a simple polygon by a mobile searcher. They examine two cases: one in which the searcher's visibility is restricted to k rays emanating from its position, and one in which the searcher can see in all directions simultaneously. Their work assumes a single searcher. The paper presents necessary and sufficient conditions for a polygon to be searchable by various searchers. They introduce a class of polygons for which the searcher with two flashlights is as capable as the searcher with a point light source, and give necessary and sufficient conditions for polygons to be searchable by a searcher with two flashlights.

LaValle *et al.* [19] introduce the visibility-based motion planning problem of locating an unpredictable target in a workspace with one or more robots, regardless of the movements of the target. They define a visibility region for each robot, with the goal of guaranteeing that the target will eventually lie in at least one visibility region. A formal characterization of the problem and some problem instances are presented. For a simply-connected free space, a logarithmic bound on the minimum number of robots needed is established. A complete algorithm for computing the motion strategy of the robots is presented, based on searching a finite cell complex constructed based on critical information changes. In this approach, the search space is at least exponential in the number of robots, and appears to be NP-hard.

LaValle *et al.* [18] addresses the related question of maintaining the visibility of a moving target in a cluttered workspace by a single robot. They are able to optimize the path along additional criteria, such as the total distance traveled. They divide the problem into two cases: predictable targets and partially predictable targets. For the predictable case, an algorithm that computes the optimal solution is presented; this algorithm is exponential in the dimension of the robot configuration space. For the partially predictable targets, two strategies are developed. One maximizes the probability that the target

remains in view in a subsequent time step, and the second maximizes the minimum time in which the target could escape the visibility region. For these approaches, the search space is also exponential.

Another large area of related research has addressed the problem of multi-target tracking (e.g., Bar-Shalom [2], [3], Blackman [7], Fox *et al.* [13]). This problem is concerned with computing the trajectories of multiple targets by associating observations of current target locations with previously detected target locations. In the general case, the sensory input can come from multiple sensory platforms. Other work related to predicting target movements includes stochastic game theory, such as the hunter and rabbit game ([5], [6]), which is the problem of determining where to shoot to minimize the survival probability of the rabbit. Our task in this paper differs from this work in that our goal is not to calculate the trajectories of the targets, but rather to find dynamic sensor placements that minimize the collective time that any target is not being observed by at least one of the mobile sensors.

In the area of multi-robot surveillance, Everett *et al.* [10] have developed a coordinated multiple security robot control system for warehouse surveillance and inventory assessment. The system is semi-autonomous, and utilizes autonomous navigation with human supervisory control when needed. They propose a hybrid navigational scheme which encourages the use of known "virtual paths" when possible. Wesson *et al.* [39] describe a distributed artificial intelligence approach to situation assessment in an automated distributed sensor network, focusing on the issues of knowledge fusion. Durfee *et al.* [9] describe a distributed sensor approach to target tracking using fixed sensory locations. As before, this related research in multi-robot surveillance does not deal with the issue of interest in this article — the dynamic placement of mobile sensors.

4. Approach

4.1. Overview

Our proposed approach to the *CMOMMT* problem is based upon the same philosophy of control as our previously-developed ALLIANCE architecture for fault tolerant multi-robot control ([30]). In this approach, we incorporate a distributed, real-time reasoning system utilizing motivations of behavior to control the activation of task achieving control mechanisms. For the purposes of fault tolerance, we utilize no centralized control, but rather enable each individual robot to select its own current actions. We do not make use of negotiation among robots, but rather rely upon broadcast messages from robots to announce their current activities. This approach to communication and action selection, as embodied in the ALLIANCE architecture, results in multi-robot cooperation that gracefully degrades and/or adapts to real-world problems, such as robot failures, changes in the team mission, changes in the robot team, or failures or noise in the communication system. This approach has been successfully applied in the ALLIANCE architecture to a variety of cooperative robot problems, including mock hazardous waste cleanup ([27], [30]), bounding overwatch ([29], [28]), janitorial service [29], box pushing [26], and simple cooperative baton-passing [31], implemented on both physical and simulated robot teams.

In [32], we provide the details of the behavior organization that implements our *A-CMOMMT* approach in the ALLIANCE architecture. In the current article, we focus on an analysis of the approach and compare it to three other control mechanisms to determine its usefulness in solving the *CMOMMT* problem. In the *A-CMOMMT* approach, robots use weighted local force vectors that attract them to nearby targets and repel them from nearby robots. The weights are computed in real-time and are based on the relative locations of the nearby robots and targets. The weights are aimed at generating an improved collective behavior across robots when utilized by all robot team members.

In the following subsections, we provide more details of this approach, describing the issues of target and robot detection, computation of the local force vectors, force vector weights, and the combination of the weighted force vectors.

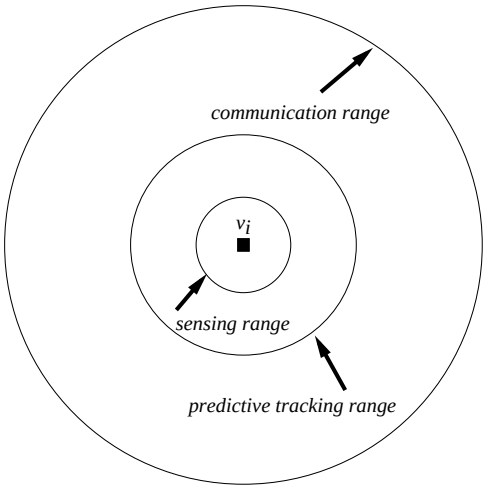


Fig. 1. Definition of the *sensing range*, *predictive tracking range*, and *communication range* of a robot v_i. Although the exact range values may change from application to application, we assume that the relative ordering of range distances remains the same.

4.2. Target and robot detection

Ideally, robot team members would be able to passively observe (e.g., through visual image processing) nearby robots and targets to ascertain their current positions and velocities. Research fields such as machine vision have dealt extensively with this topic, and have developed algorithms for this type of passive position calculation. However, since the physical tracking and 2D positioning of visual targets is not the focus of this research, we instead assume that robots use a global positioning system (such as GPS for outdoors, or the laser-based MTI Conac indoor positioning system [15] that is in use at our laboratory) to determine their own position and the position of targets within their sensing range, and to communicate this information to the robot team members within their communication range.

In our approach, each robot communicates to its teammates the position of all targets within its field of view. To clarify this idea, Figure 1 depicts three ranges that are defined with respect to each robot v_i. The innermost range is the *sensing range* of v_i, within which the robot can use a sensor-based tracking algorithm to maintain observation of targets in its field of view. The middle range is the *predictive tracking range* of the robot v_i, which defines the range in which targets localized by other robots $v_k \neq v_i$ can affect v_i's movements. Thus, a robot can know about two types of targets: those that are directly sensed or those that are "virtually" sensed through predictive tracking. The outermost range is the *communication range* of the robot, which defines the extent of the robot's communicated messages. For simplicity, we assume that the sensing, predictive tracking, and communications ranges are identical (respectively) across all robot team members.

When a robot receives a communicated message regarding the location and velocity of a sighted target that is within its predictive tracking range, it begins a predictive tracking of that target's location, assuming that the target will continue linearly from its current state. This predictive tracking will then give the robot information on the likely location of targets that are not directly sensed by the robot, so that the robot can be influenced not only by targets that are directly sensed, but also by targets that may soon enter the robot's sensing range. We assume that if not enough information is available to disambiguate distinct targets (e.g., due to a high density of targets, a low frequency of sensor observations, a low communication bandwidth, etc.) then existing tracking approaches (e.g., Bar-Shalom [3]) should be used to uniquely identify each target based upon likely trajectories.

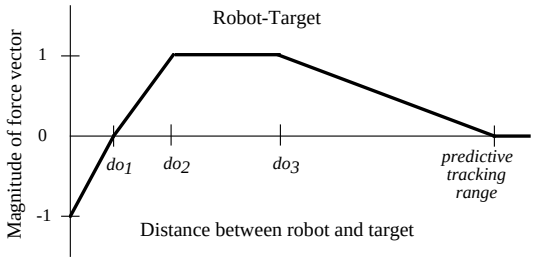


Fig. 2. Function defining the magnitude of the force vectors acting on a robot due to a nearby target.

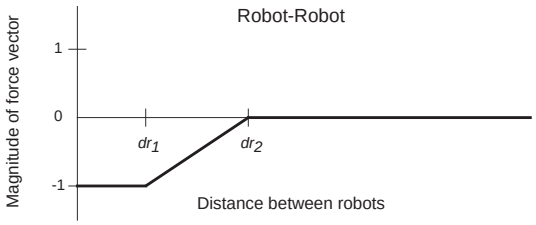


Fig. 3. Function defining the magnitude of the force vector acting on a robot due to another nearby robot.

Note that this communicated information is *not* equivalent to global information. We do not assuem that all robots have information bout all targets, since this assumption does not scale as the number of targets and robots increases. In the following, robots only use information about nearby targets and nearby robots in their control approach. Thus, our approach is not equivalent to a global controller.

4.3. Local force vector calculation

In performing their mission, the robots should be close enough to the targets to be able to take advantage of their (i.e., robots') more sophisticated tracking devices (e.g., cameras), while remaining dispersed to cover more terrain. Thus, the local control of a robot team member is based upon a summation of force vectors which are attractive for nearby targets and repulsive for nearby robots. The function in Figure 2 defines the relative magnitude of the attractive forces of a target within the predictive tracking range of a given robot. Note that to minimize the likelihood of collisions, the robot is repelled from a target if it is too close to that target ($distance < do_1$). The range between do_2 and do_3 defines the preferred tracking range of a robot from a target. In practice, this range will be set according to the type of tracking sensor used and its range for optimal tracking. In the work reported here, we have not studied how to optimize the settings of these thresholds, leaving this to future work. The attraction to the target falls off linearly as the distance to the target varies from do_3. The attraction goes to 0 beyond the predictive tracking range, indicating that this target is too far to have an effect on the robot's movements. The robot *sensing range* will lie somewhere between do_3 and the predictive tracking range.

Figure 3 defines the magnitude of the repulsive forces between robots. If the robots are too close together ($distance < dr_1$), they repel strongly. If the robots are far enough apart ($distance > dr_2$), they have no effect upon each other in terms of the force vector calculations. The magnitude scales linearly between these values.

4.4. Weighting the force vectors

Using only local force vectors for this problem neglects higher-level information that may be used to improve the team performance. Thus, we enhance the control approach by weighting the contributions of each target's force field on the total computed field. The weight w_{lk} reduces robot r_l's attraction to a nearby target if that target is within the field of view of another nearby robot. Using these weights helps reduce the overlap of robot sensory areas toward the goal of minimizing the likelihood of a target escaping detection. Here, each robot assumes that its teammates have the same sensing range as its own.

If robot v_l detects another robot v_j nearby and within sensing range of target o_k, then w_{lk} should be set to a low value. In the simplest case, since we define (in section 2) a robot v_l to be *observing* a target o_k when it is within v_l's sensing range, we could assign w_{lk} to be zero whenever another robot is within sensing range of o_k. However, this will increase the likelihood that a target will escape detection, and thus it may give better results to set $w_l k$ to some non-zero value.

The proper setting of w_{lk} is also dependent upon the estimated density of targets in the vicinity. If targets are sparsely located in the area, then the robot team risks losing track of a higher percentage of targets if any targets are ignored. On the other hand, if targets are densely distributed, then the risks are lower. In related work, we are currently exploring the proper computation of these weights based upon these issues. The results of this related work will be reported in the future.

These weights have the effect of causing a robot to prefer the observation of certain targets over others. In more complex versions of the *CMOMMT* problem, robots could also learn about the viewing capabilities of their teammates, and discount their teammates' observations if that teammate has been unreliable in the past.

4.5. Combination of weighted local force vectors

The weighted local force vectors are combined to generate the commanded direction of robot movement. This direction of movement for robot v_l is given by:

$$\sum_{k=1}^{n} w_{lk}\,\mathbf{f_{lk}} + \sum_{i=1, i\neq l}^{m} \mathbf{g_{li}}$$

where $\mathbf{f_{lk}}$ is the force vector attributed to target o_k by robot v_l and $\mathbf{g_{li}}$ is the force vector attributed to robot v_i by robot v_l. To generate an (x, y) coordinate indicating the desired location of the robot corresponding to the resultant force vector, we scale the resultant force vector based upon the size of the robot. The robot's speed and steering commands are then computed to move the robot in the direction of that desired location. Both of these computed commands are functions of the angle between the robot's current orientation and the direction of the desired (x, y) position. The larger the angle, the higher the commanded rate of steering and the lower the commanded speed. For small angles, the speed is a function of the distance to the desired (x, y) location, with longer distances translating to faster speeds, up to a maximum robot speed. A new command is generated each time the force vector summation is recomputed. While this approach does not guarantee smooth robot paths, in practice, we have found that the force vector summations yield a desired (x, y) location that moves relatively smoothly over time, thus resulting in a smooth robot motion in practice.

We note here that the velocity and steering command can be overwritten by an *Avoid Obstacles* behavior, which will move the robot away from any obstacle that is too close. This is achieved by treating any such obstacle as an absolute force field that moves the robot away from the obstacle.

It is also important to note that the issue of noisy sensors plays a role in helping to ensure that the robot behavior corresponding to the vector summations is appropriate. The stochastic nature of a robot's sensory measurements prevents undesirable singularities from occurring. For example, if a robot were

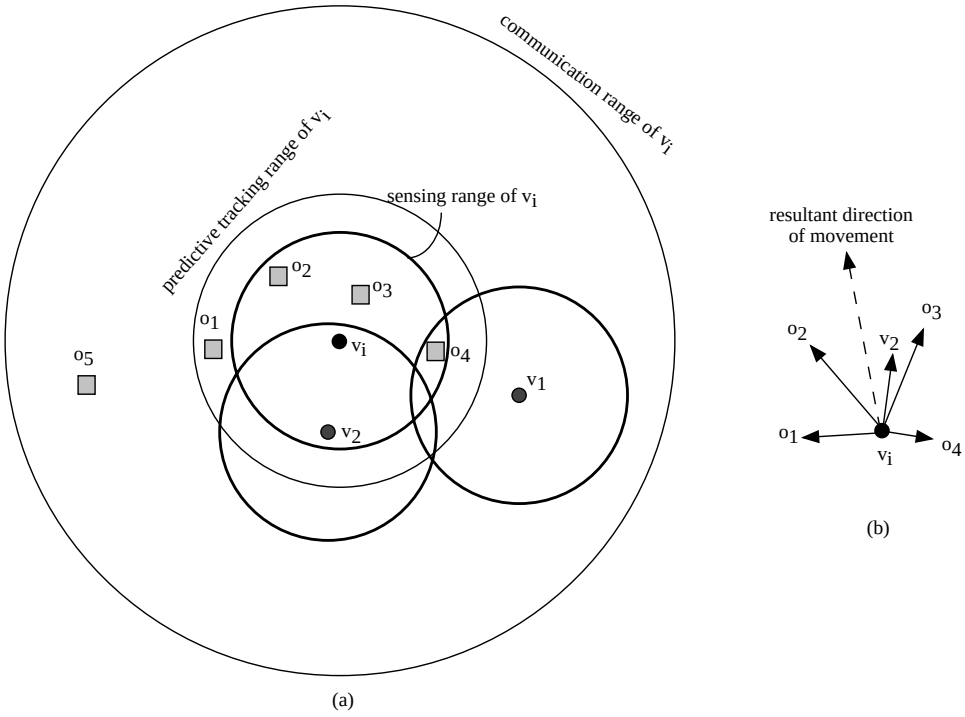


Fig. 4. Example of forces acting on robot v_i. In this figure, robots are represented by small black dots and targets are represented by small grey squares. (a) Robot v_i has four targets within its predictive tracking range and two other robots within its communication range. One other target, o_5, is outside v_i's predictive tracking range. (b) Forces acting on robot v_i are attractive forces to targets o_1, o_2, o_3. and o_4, and a repulsive force from v_2. Note that the attraction to o_1 is less than that to o_2 or o_3, since o_1 is outside the sensing range, but within the predictive tracking range. The attraction to o_4 is less than that to o_2 and o_3 because o_4 is within the sensing range of another robot, v_1. There is no attraction to target o_5, since o_5 is outside v_i's predictive tracking range. There is also no repulsion acting on v_i due to v_1 because v_1 is already sufficiently far from v_i.

located equidistant between two targets that move away from the robot at the same rate, sensory noise can prevent the force vectors from cancelling the attraction to zero. Instead, slight deviations in sensing can cause the robot to begin to be attracted more towards one of the targets, leading the robot to follow that single target, rather than losing both targets.

Figure 4 gives an example of the generation of the weighted local force vectors for robot v_i at a given point in time. In part (a) of this example, there are five targets, o_1, o_2, o_3, o_4, and o_5, and two other robots, v_1 and v_2, within v_i's communication range. The magnitude of the force vectors attracting robot v_i to targets o_2 and o_3 is equivalent to the maximum value, since those targets are within v_i's sensing range but not within any other robot's sensing range. The attraction of v_i to target o_1 is less than that for o_2 and o_3, because o_1 is outside the sensing range of v_i (although still within predictive tracking range). The force vector to target o_4 is weighted less due to o_4's presence within v_1's sensing range. Finally, there is no attraction to target o_5, because it is outside v_i's predictive tracking range. Robot v_i also experiences a repulsive force due to robot v_2, because of the close proximity of the robots. There is no repulsion due to v_1, since v_1 is sufficiently distant. Part (b) of Figure 4 shows the resultant direction of movement of v_i due to the weighted force vectors.

4.6. Seeking out targets

Of course, for any cooperative observation technique to be of use, the robots must first find targets to observe. All techniques are trivially equivalent if no targets are ever in view. Thus, the robots must have some means of searching for targets if none are currently detected. We have not yet implemented an approach for seeking out targets, but here we discuss a few ideas as to how this might be done.

In the *A-CMOMMT* and *Local* approaches, when a robot does not detect any target nearby, the weighted sum of the force vectors will cause each robot to move away from its robot neighbors and then idle in one location. While this may be acceptable in some applications, in general, we would like to have the robots actively and intelligently seek out potential targets in the area. Suzuki and Yamashita [37] address this problem through the development of search schedules for "∞-searchers". An "∞-searcher" is a mobile searcher that has a 360^o infinite field of view. A search schedule for an ∞-searcher is a path through a simple polygonal area that allows the searcher (or robot) to detect a mobile "intruder" (or target), regardless of the movements of the target. While clearly related to the *CMOMMT* problem, this earlier work makes a number of assumptions that do not hold in the *CMOMMT* problem: infinite range of searcher visibility, only a single searcher, only a single target, and an enclosed polygonal area which does not allow any targets to enter or exit the area. Nevertheless, this approach could potentially be extended and adapted for use in the automated observation problem.

Another approach would have robots influence their search to concentrate their movements in areas previously known to have a higher likelihood of target appearance. For instance, since new targets are more likely to appear near entrances, a robot could spend a higher percentage of its time near entrances. A challenge would be developing an efficient mechanism for dealing with the more global nature of the problem when dealing with knowledge of previous target movements and target concentrations.

5. Experimental results and discussion

To evaluate the effectiveness of the *A-CMOMMT* algorithm in addressing the *CMOMMT* problem, we conducted experiments both in simulation and on a team of 4 Nomad mobile robots. The simulation studies allowed us to test larger numbers of robots and targets, while the physical robot experiments allowed us to validate the results discovered in simulation in the real world. The results that we present below summarize the outcomes of nearly 1,000,000 simulation test runs, and over 750 physical robot experimental runs.

In both sets of studies, we compared four possible cooperative observation policies: (1) *A-CMOMMT* (weighted force vector control), (2) *Local* (nonweighted force vector control), (3) *Random* (robots move random/linearly), and (4) *Fixed* (robots remain in fixed positions). The *Local* algorithm computes the motion of the robots by calculating the same local force vectors of *A-CMOMMT*, but without the force vector weights. This approach was studied to determine the effectiveness of the weights on the force vectors in the *A-CMOMMT* algorithm. The last two approaches are control cases for the purposes of comparison: the *Random* approach causes robots to move according to a "random/linear" motion (defined below), while the *Fixed* approach distributes the robots uniformly over the area $\mathcal{S}$, where they maintain fixed positions. In these last two approaches, robot movements are not dependent upon target locations or movements. While we recognize that these control approaches are "straw men" approaches in the sense that any number of other control approaches could be imagined, they seem to be obvious choices that are important to evaluate. The objective of our ongoing learning research (briefly introduced in Section 6) is to develop a principled approach to search the domain of potential solutions to this challenging problem.

The set of experiments conducted in simulation varied somewhat from the experimentations on the physical robots. In simulation, we studied two types of target movements – (1) random/linear and (2) evasive – whereas in the physical robot experiments, we only studied random/linear target movements. In addition, the simulation experiments were conducted using a circular region $\mathcal{S}$ with no obstacles (other than the boundary), whereas on the physical robots, the work area $\mathcal{S}$ was a rectangular area. Furthermore,

in one collection of experiments on the physical robots, we also populated $\mathcal{S}$ with varying densities of rectangular obstacles of size approximately equal to the robot diameter. Of course, the numbers of robots and targets we could experiment with was limited on the physical robot team; in simulation, we tested up to 10 robots and 20 targets, whereas on the physical robot team, we tested up to 3 robots and 3 targets (but with a limitation of only a total of 4 robots and targets at a time).

The next subsections describe the experimental setup and results for the simulation studies, followed by the the physical robot studies.

5.1. Simulation experimental setup

We performed two sets of experiments in simulation. In the first set, targets move according to a "random/linear" movement, which causes each target to move in a straight line, with a 5% chance at each step (i.e., every $\delta t = 1$ second) to have the direction of movement altered randomly between $-90°$ and $+90°$. When the boundary of $\mathcal{S}$ is reached, the targets reflect off the boundary. Targets are randomly assigned a fixed speed at the beginning of each experiment between the values of 0 and 150 units/second, while robots are assigned a fixed speed of 200 units/second. At the beginning of each experiment, robots and targets are randomly positioned and oriented in the center region of $\mathcal{S}$ (except for the robots in the *Fixed* approach).

In the second set of simulation experiments, targets move evasively, in which they try to avoid detection by nearby robots. The calculations for the evasive movements were the same as the local force vectors between neighboring robots, except that targets were given the added benefit of a larger range of view than that of the robots, in order to magnify the behavioral differences in evasive versus random/linear target motions. Thus, in these experiments, the targets had a sensing range of 1.5 times that of the robots. If a target did not see a robot within its field of view, it moved linearly along its current direction of movement, with boundary reflection. All other experimental setups were the same as for the experiments with the random/linear target movements.

We compared these four approaches in both sets of experiments by measuring the A metric (see section 2 for the definition of A) during the execution of the algorithm. Since the algorithm performance is expected to be a function of the number of robots m, the number of targets n, the robot *sensing range*, and the relative size of the area $\mathcal{S}$; we collected data for a wide range of values of these variables. To simplify the analysis of our simulation results, we defined the area $\mathcal{S}$ as the area within a circle of radius R, fixed the range of robot sensing at 2,600 units of distance, treated robots and targets as points, and included no obstacles within $\mathcal{S}$ (other than the robots and targets themselves, and the boundary of $\mathcal{S}$). For the force vector functions defined in Figures 2 and 3, we used the following parameter settings:

$$\begin{aligned} do_1 &= 400 \\ do_2 &= 800 \\ do_3 &= 2600 \\ predictive\_tracking\_range &= 3000 \\ dr_1 &= 1250 \\ dr_2 &= 2000 \end{aligned}$$

We varied m from 1 to 10, n from 1 to 20, and R from 1,000 to 50,000 units. For each instantiation of variables n, m, and R, we evaluated the A metric for runs of length 2 minutes. We then repeated this process for 250 runs for each instantiation to derive an average A value for the given values of n, m, and R.

Since the optimum value of the A metric for a given experiment equals the number of targets, n, we normalized the experiments by computing A/n, which is the average percentage of targets that are within some robot's view at a given instant of time. This allows us to compare the results of experiments that vary in the number of targets.

5.2. Statistical significance

To verify the statistical significance of our results, we used the Student's t distribution, comparing the policies two at a time for all six possible pairings. In these computations, we used the null hypothesis: $H_0 : \mu_1 = \mu_2$, *and there is essentially no difference between the two policies.* Under hypothesis H_0:

$$T = \frac{\overline{X}_1 - \overline{X}_2}{\sigma\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}} \quad where \quad \sigma = \sqrt{\frac{n_1 S_1^2 + n_2 S_2^2}{n_1+n_2-2}}$$

Then, on the basis of a two-tailed test at a 0.01 level of significance, we would reject H_0 if T were outside the range $-t_{.995}$ to $t_{.995}$, which for $n_1 + n_2 - 2 = 250 + 250 - 2 = 498$ degrees of freedom, is the range -2.58 to 2.58. For the data given in the figures in this section, we found that we could reject H_0 at a 0.01 level of significance for all pairing of policies that show a visible difference in performance in these figures. Thus, we can conclude that the variation in performance of the policies illustrated by the following results is significant.

5.3. Qualitative comparison

Our results can be discussed both from a qualitative perspective of variation in behavior, and from a quantitative perspective, by evaluating and comparing metrics of performance of each approach. In this subsection, we present and discuss the behavioral differences among the four approaches. The next subsection discusses the quantitative comparisons of the approaches.

Figure 5 shows snapshots of the beginning of the experiments for each of the four approaches for three robots and six targets for random/linear target movements; Figure 6 shows similar snapshots for five robots and twenty targets. All of the snapshot figures show results for a work space radius, R, of 10,000. In these figures (and similar ones to follow), the small black circles give the positions of the robots, while the small gray squares give the positions of the targets. The gray circles around each robot represent the sensing range of that robot. The boundary of the area $\mathcal{S}$ is the large black circle. Traces of the robot and target paths are shown; the spacing of the dots along each trace indicates the relative speed of the corresponding robot or target.

Both of these Figures 5 and 6 illustrate the "harder" observation problem (but typical of the problems we are interested in), which occurs when there are more targets than robots. Thus, an assignment of one robot per target would not be a straightforward solution to the problem. In the *Local* approach, the robots tend to cluster near the center of mass of the targets, with some separation due to the repulsive forces between the robots. This leads to several targets being under observation by multiple robots, while other targets escape observation. In the *A-CMOMMT* approach, the robots exhibit more distribution due to the weights on the force vectors that cause them to be less attracted to targets that are already under observation by a nearby robot. Thus, more targets remain under observation.

Figures 7 and 8 show similar results for evasive target movements. The *Fixed* and *Random* approaches do much worse here, because they do not get observation "for free" from targets wandering through their sensing ranges. Instead, the targets avoid them, and very little time elapses during which targets are under observation. The *Local* and *A-CMOMMT* approaches perform similarly to before, except that now they suffer a stronger penalty for losing a target because, once lost, targets tend to remain out of the sensing range of the robots.

5.4. Quantitative comparison

We can more effectively evaluate the approaches by quantitatively comparing the approaches based upon the A metric described earlier. Figure 9 shows the results of our first set of experiments where targets move

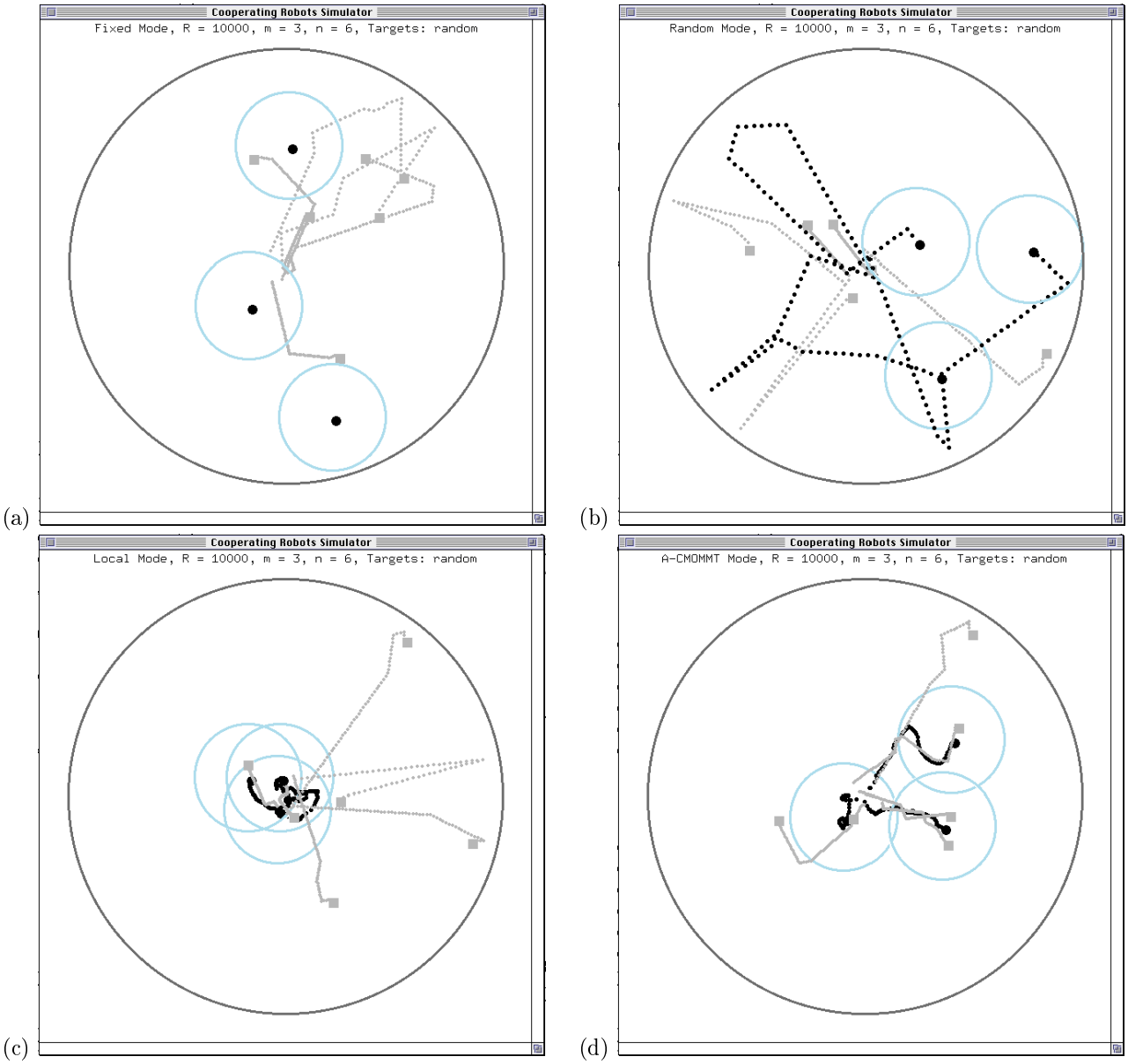


Fig. 5. Simulation results of 3 robots and 6 targets, with targets moving random/linearly, for four algorithms: (a) *Fixed*, (b) *Random*, (c) *Local*, and (d) *A-CMOMMT*. In this and similar figures that follow, the robots are represented by black circles, and the targets are represented by gray squares. The gray circles surrounding each robot indicate the robot sensing range. Traces of robot and target movements are also shown.

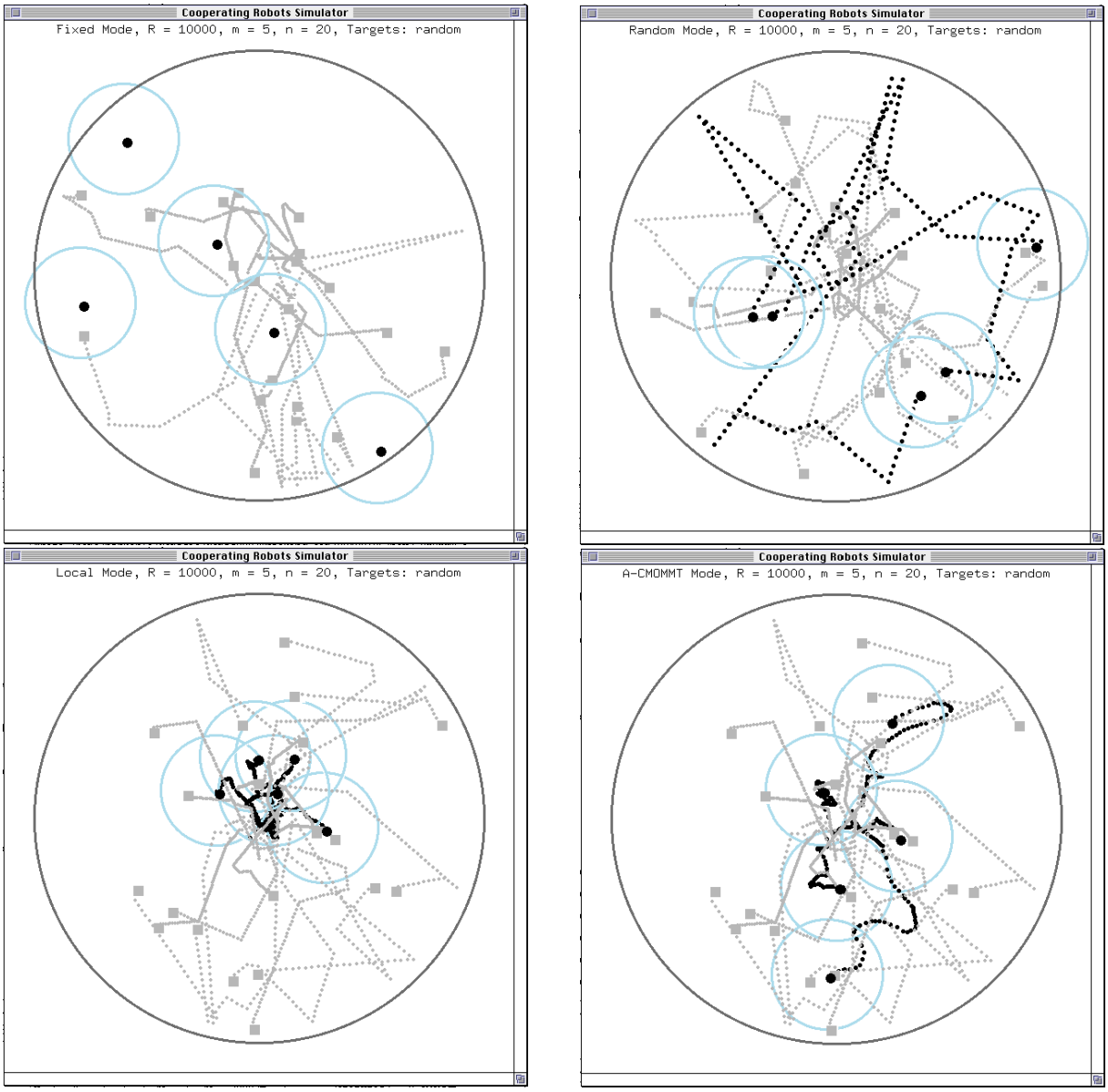


Fig. 6. Simulation results of 5 robots and 20 targets, with targets moving random/linearly.

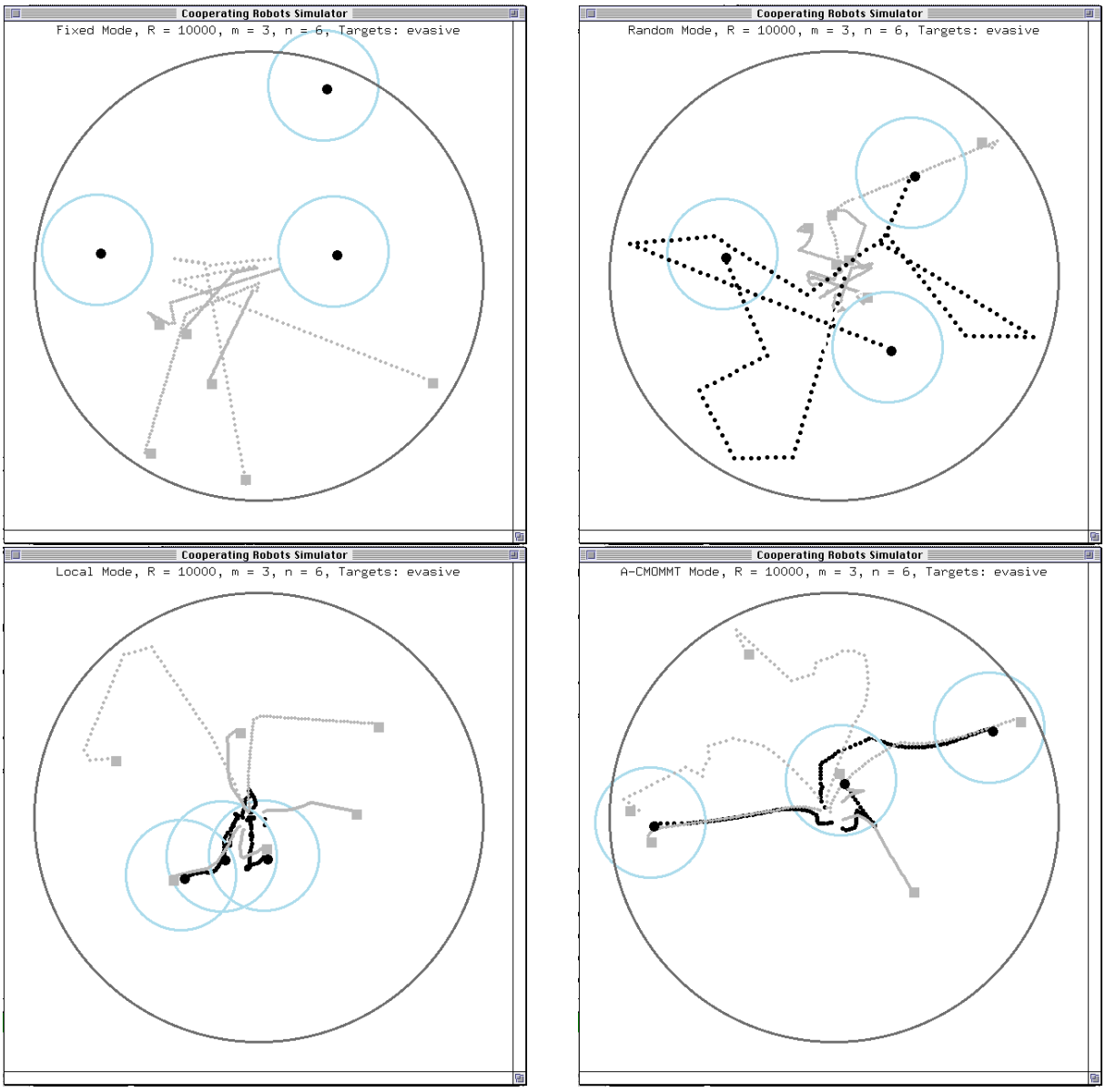


Fig. 7. Simulation results of 3 robots and 6 targets, with targets moving evasively.

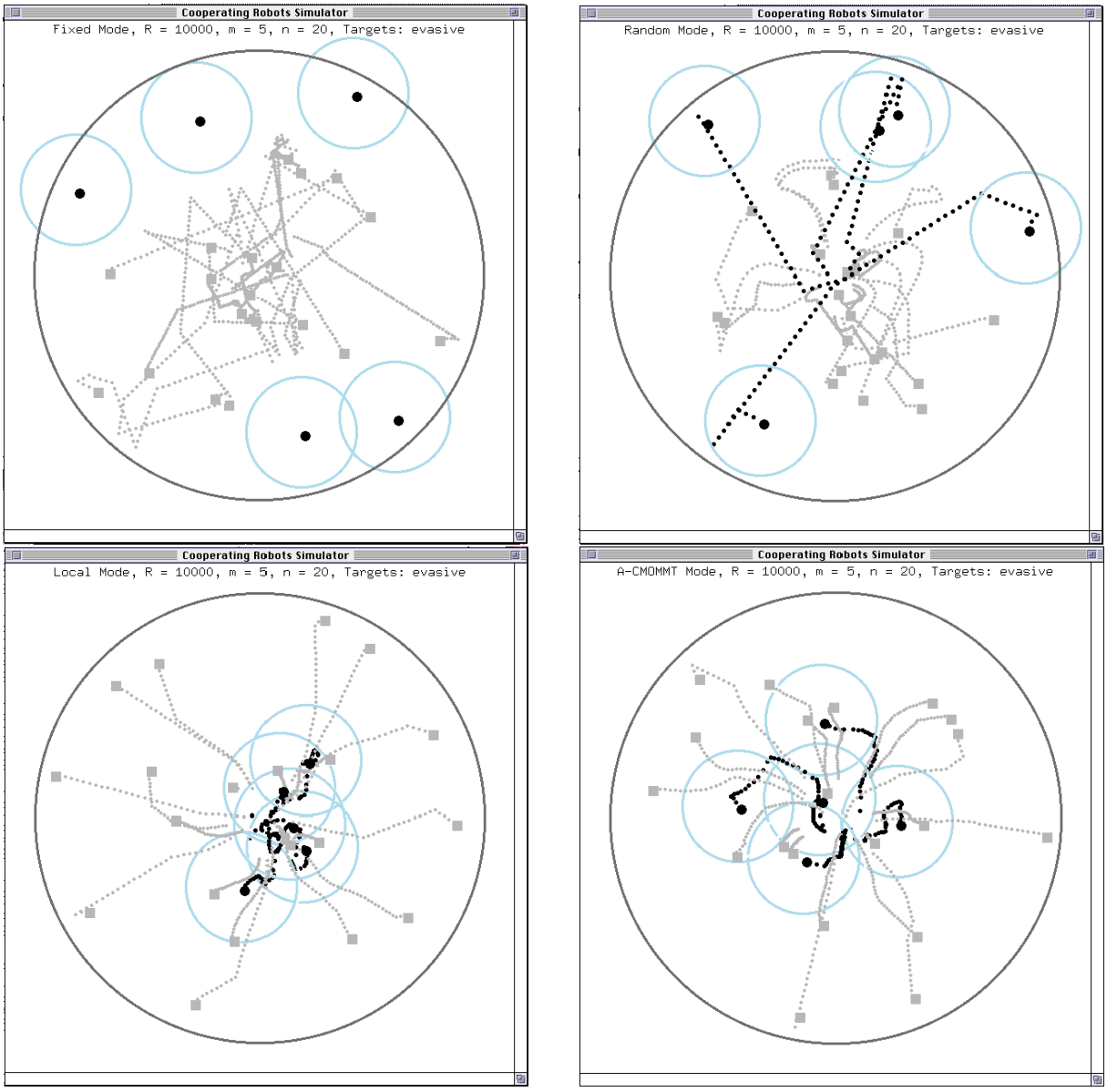


Fig. 8. Simulation results of 4 robots and 20 targets, with targets moving evasively.

Percentage improvement in simulation of *A-CMOMMT* and *Local* over baseline, for R > 10,000, r = 2600, and *random* target movements

$\frac{n}{m}$	$\frac{1}{5}$	$\frac{1}{2}$	1	4	10
A-CMOMMT	654%	550%	600%	570%	511%
Local	708%	550%	525%	450%	389%

Table 1. Percentage improvement in simulation of *A-CMOMMT* and *Local* algorithms over the baseline (which we define to be the *Random* approach). These results are for radius of work area, R, not less than 10,000, radius of robot sensing range, r, of 2600, and targets moving random/linearly. The values are given in terms of the ratio of the number of targets (n) to the number of robots (m).

Percentage improvement in simulation of *A-CMOMMT* over *Local*, for R > 10,000, r = 2600, and *random* target movements

$\frac{n}{m}$	$\frac{1}{5}$	$\frac{1}{2}$	1	4	10
A-CMOMMT	-8%	0%	14%	27%	31%

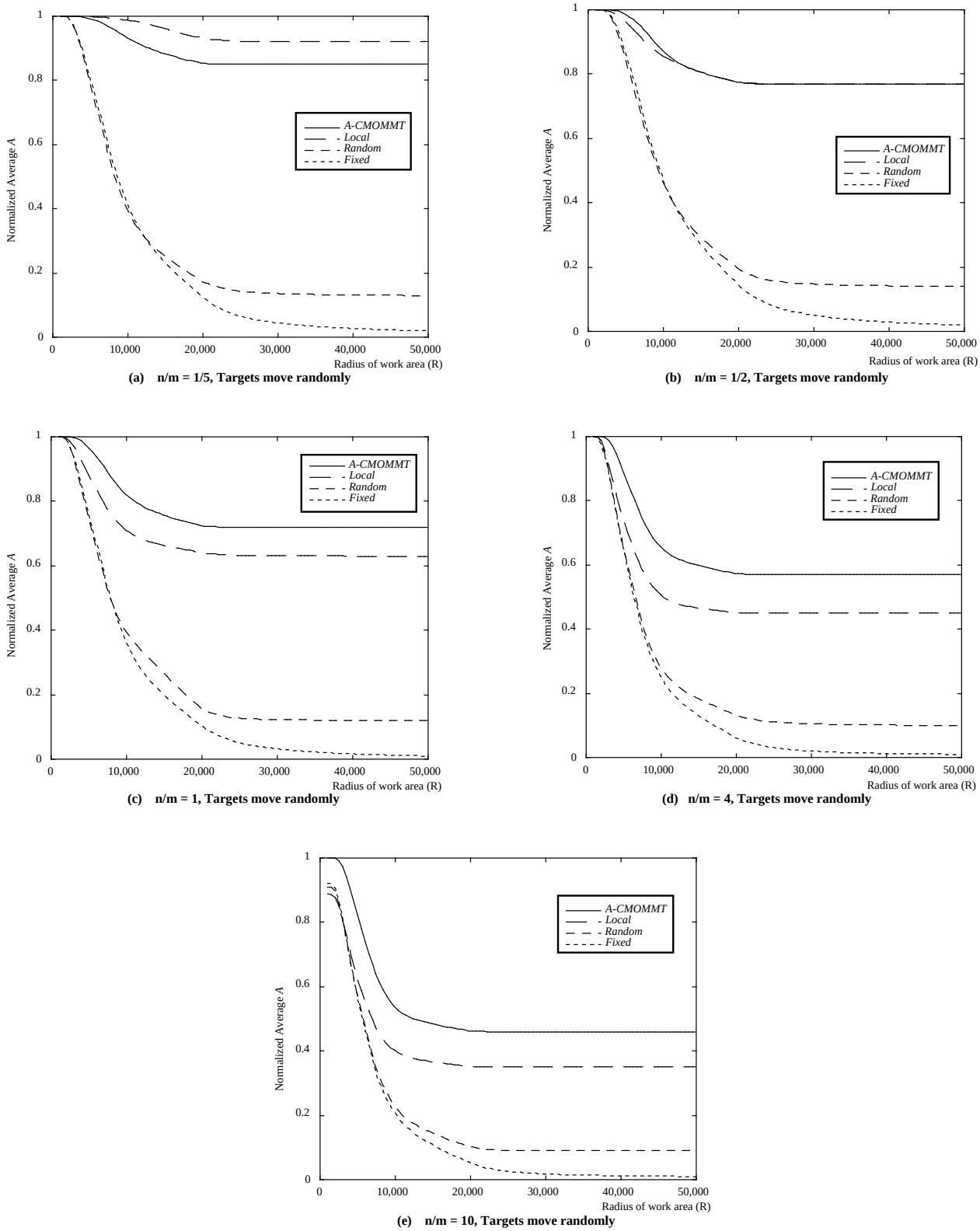
Table 2. Percentage improvement in simulation of *A-CMOMMT* over *Local*, for radius of work area, R, not less than 10,000, radius of robot sensing range, r, of 2600, and targets moving random/linearly. The values are given in terms of the ratio of the number of targets (n) to the number of robots (m).

random/linearly. Table 1 gives the percentage improvement of *A-CMOMMT* and *Local* over the *Random* approach for the results shown in Figure 9. Table 2 shows this improvement in terms of *A-CMOMMT* over *Local*.

The results are given as a function of the radius of the work area, R, for five different ratios of the number of targets to the number of robots (n/m). As expected, all of the approaches degrade in performance as the size of the area under observation increases. The more naive approaches — *Fixed* and *Random* — degrade very quickly as R increases, which is expected since these approaches use no information about target positions. In the limit, the *Random* approach performs better than *Fixed*, due to the proximity of the initial starting locations of the robots and targets in the *Random* approach. Thus, for real-world applications in which targets move along more predictable paths rather than randomly, this suggests that a significant benefit can be gained by having robots learn areas of the environment $\mathcal{S}$ where targets are more likely to be found, and for robots to concentrate their observations in those locations.

More interesting is the comparative performance of *A-CMOMMT* versus *Local*. As Table 2 shows, *A-CMOMMT* performs from 8% worse than *Local* for $n/m = 1/5$, to 31% better for $n/m = 10$. Thus, for $n/m, < 1/2$ (i.e., less than half as many targets as robots), *Local* performs better than *A-CMOMMT*, while for $n/m > 1/2$, *A-CMOMMT* is the superior performer. Since the "harder" version of the *CMOMMT* problem is when there are many more targets than robots, these results show that the higher-level weighting is beneficial for solving the harder problems — up to a 31% improvement.

However, when there are only a few targets, the weight factors work to the disadvantage of the team, resulting in up to an 8% degradation. This is because the weights cause the robot team to run the risk of losing targets if their attraction to those targets is influenced by other robots being nearby. This is a not a problem if there are a sufficient number of targets such that most robots have other target observation choices; but it is a problem for cases of relatively few targets. Figure 10 gives an example of this situation, in which there are five robots and one target. Averaged over 250 runs of this situation, the *A-CMOMMT* approach yields a normalized value of A of 0.82, while the *Local* approach yields a normalized value of 0.99. Thus, the *A-CMOMMT* approach usually allows the team to track the target, as shown in Figure 10



(a) n/m = 1/5, Targets move randomly

(b) n/m = 1/2, Targets move randomly

(c) n/m = 1, Targets move randomly

(d) n/m = 4, Targets move randomly

(e) n/m = 10, Targets move randomly

Fig. 9. Simulation results of 4 distributed approaches to cooperative observation, for random/linear target movements, for various ratios of number of targets (n) to number of robots (m).

Percentage improvement in simulation of *A-CMOMMT* and *Local* over baseline, for R > 10,000, r = 2600, and *evasive* target movements

$\frac{n}{m}$	$\frac{1}{5}$	$\frac{1}{2}$	1	4	10
A-CMOMMT	770%	618%	700%	525%	429%
Local	870%	655%	656%	438%	343%

Table 3. Percentage improvement of *A-CMOMMT* and *Local* algorithms over the baseline (which we define to be the *Random* approach). These results are for radius of work area, R, not less than 10,000, radius of robot sensing range, r, of 2600, and targets moving evasively. The values are given in terms of the ratio of the number of targets (n) to the number of robots (m).

Percentage improvement in simulation of *A-CMOMMT* over *Local*, for R > 10,000, r = 2600, and *evasive* target movements

$\frac{n}{m}$	$\frac{1}{5}$	$\frac{1}{2}$	1	4	10
A-CMOMMT	-11%	0%	7%	20%	25%

Table 4. Percentage improvement of *A-CMOMMT* over the *Local* approach, for radius of work area, R, not less than 10,000, radius of robot sensing range, r, of 2600, and targets moving evasively. The values are given in terms of the ratio of the number of targets (n) to the number of robots (m).

(a). However, in several occurrences, the team loses the target, as shown in Figure 10 (b). The *Local* approach, on the other hand, nearly always allows the team to maintain the target in view, as shown in Figure 10 (c). Thus, a fielded version of the *A-CMOMMT* approach should take into account the relative numbers of targets to revise the weighting influence dynamically to better deal with small numbers of targets.

Figure 11 shows the results of our second set of experiments where the targets move evasively. Relative to the random/linear target movement results from Figure 9, the performance of all of the approaches degrades, as expected, with the *Fixed* and *Random* approaches degrading quicker. Otherwise, the relative performances remain the same. Table 3 gives the percentage improvement of *A-CMOMMT* and *Local* over the *Random* approach for the results shown in Figure 11. Table 4 gives the percentage improvement of *A-CMOMMT* over *Local* for these results, showing that *A-CMOMMT* performs from 11% worse than *Local* for $n/m = 1/5$, to 25% better for $n/m = 10$.

5.5. Robot implementation

We implemented the *A-CMOMMT* algorithm on a team of four Nomad 200 robots and performed extensive experimentation on the physical robots. The purposes of these experiments were: (1) to illustrate the feasibility of our approach for physical robot teams, (2) to compare the results with the results from simulation, and (3) to determine the impact of random scattered obstacles on the effectiveness of the proposed approach. We conducted several hundred experimental runs on the physical robots to compile statistically significant data on the performance with different numbers of trackers and targets, both with and without scattered obstacles. The following subsections describe the design and results of these physical robot experiments.

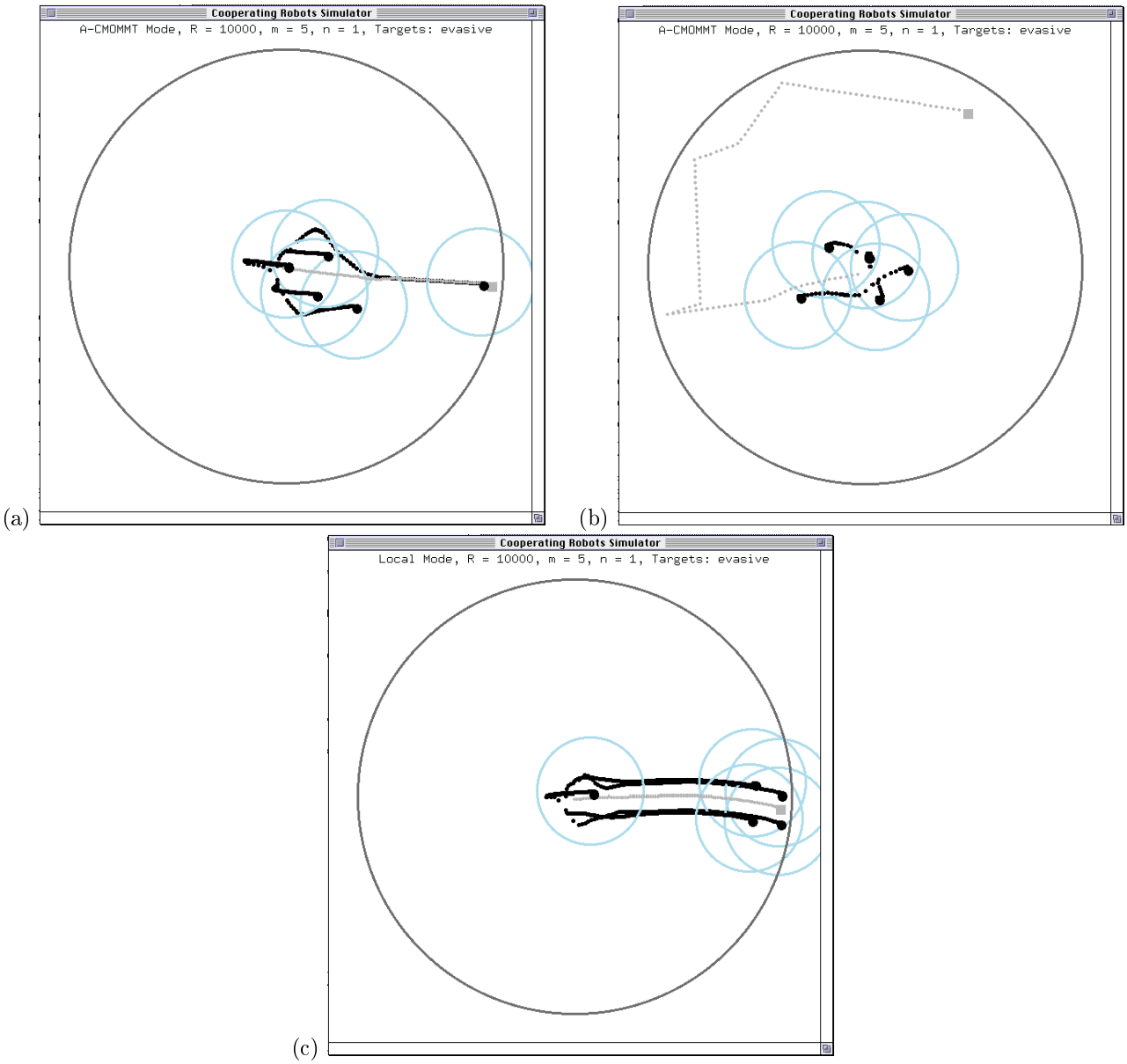
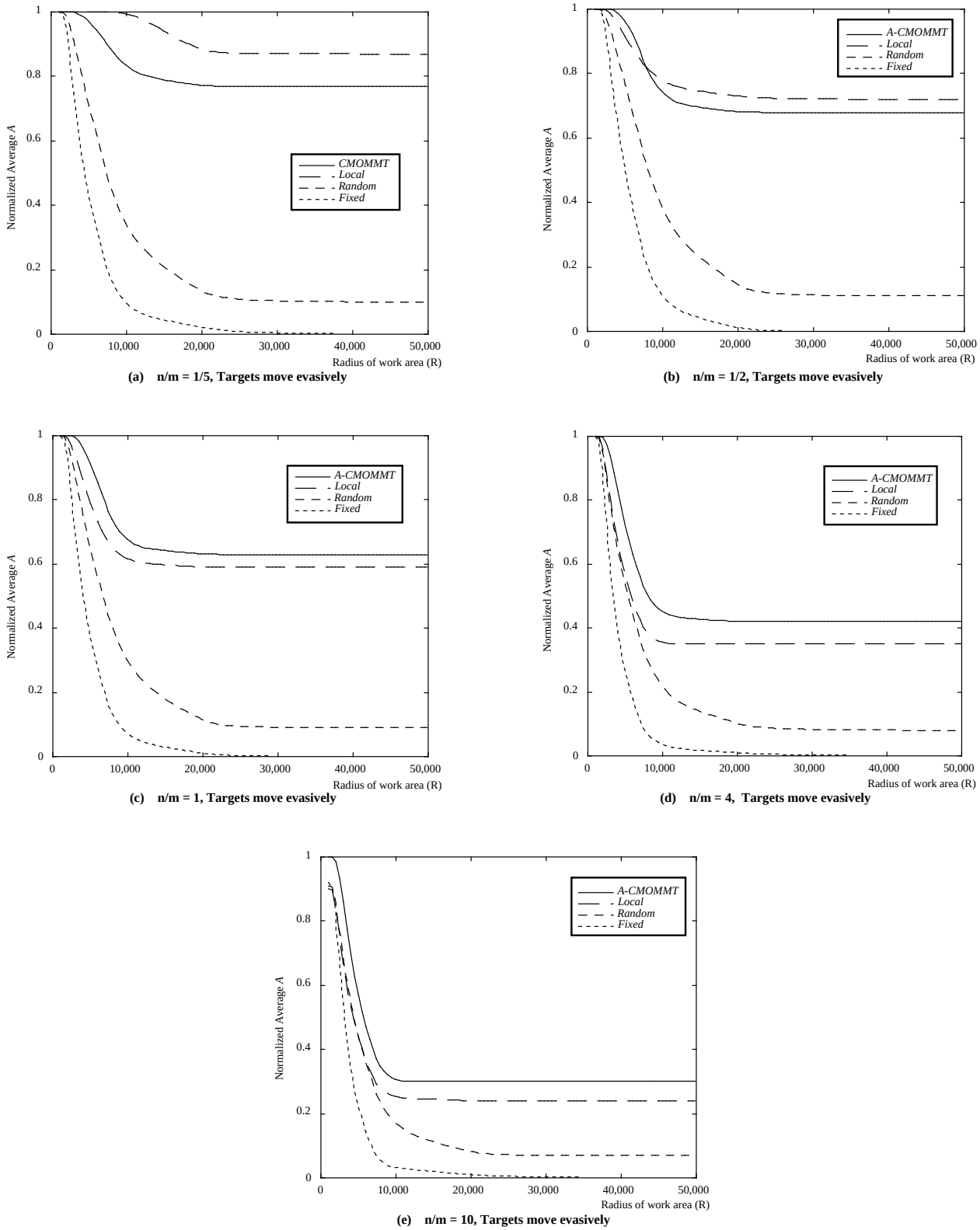


Fig. 10. Simulation results showing instance when *A-CMOMMT* performs inferior to *Local*, for five robots and one target, with targets moving evasively. The first snapshot in (a) shows the most common *A-CMOMMT* behavior, which is desirable, in which one robot ends up tracking the one target. The second snapshot in (b) shows the occasional, undesirable behavior of *A-CMOMMT*, in which all of the robots lose the target, due to the high-level weights causing robots to resist tracking the target. The third snapshot in (c) shows the behavior of the Local approach, in which nearly all robots follow the target, preventing the target from escaping observation.



(a) n/m = 1/5, Targets move evasively

(b) n/m = 1/2, Targets move evasively

(c) n/m = 1, Targets move evasively

(d) n/m = 4, Targets move evasively

(e) n/m = 10, Targets move evasively

Fig. 11. Simulation results of 4 distributed approaches to cooperative observation, for *evasive* target movements, for various ratios of number of targets (n) to number of robots (m).

Experimental design The Nomad 200 robots are wheeled vehicles with tactile, infrared, ultrasonic, 2D laser range, and indoor global positioning systems. The indoor global positioning system we use is a custom-designed 2D Conac Positioning System by MTI Research, Inc. This system uses two laser beacons, with a sensor onboard each robot that allows the robot to use triangulation to determine its own x, y coordinate location to within about ± 10 centimeters[3]. In addition, the robots are equipped with radio ethernet for inter-robot communication. In the current phase of this research, which concentrates on the cooperative control issues of distributed tracking, we utilize an indoor global positioning system as a substitute for vision- or range-sensor-based tracking. Under this implementation, each target to be tracked is equipped with an indoor global position sensor, and broadcasts its current (x, y) position via radio to the robots within communication range. Each robot team member is also equipped with a positioning sensor, and can use the targets' broadcast information to determine the relative location of nearby targets.

Figure 12 shows an example of the physical robot experiments. In these experiments, we typically designated certain robots to be targets, and other robots as observers (or searchers). We programmed the robots acting as targets to move in one of two ways: movements based on human joystick commands, or simple wandering through the area of interest. For the data collected and presented in this section, all the targets moved using autonomous wandering. No evasive motions were implemented. Since we were using our robots as both observers and targets, we studied various combinations of observers and targets, up to a total of four combined vehicles. Table 5 shows the sets of experiments we conducted. For each of the algorithms shown, ten experimental runs of 10 minutes each were conducted. The parameters of these experiments were as follows, for a work space of 12.8m $\times$ 6.1m (42ft $\times$ 20ft) and robot diameter of 0.61m (2ft):

$$\begin{aligned} do_1 &= 1m(40in) \\ do_2 &= 2m(80in) \\ do_3 &= 6.6m(260in) \\ predictive\_tracking\_range &= 10.16m(400in) \\ dr_1 &= 1m(40in) \\ dr_2 &= 15.2m(60in) \end{aligned}$$

Recognizing that the *A-CMOMMT* approach does not explicitly deal with obstacles in the environment, we conducted physical robot experiments both with and without scattered, simple, random obstacles, to determine the impact of simple obstacles on the control approaches. We added a simple obstacle avoidance behavior to the control approaches to enable robots to move around encountered obstacles. Obstacles averaged a footprint of $.6m^2$, which was grown by the obstacle avoiding behavior to a footprint of size $1.3m^2$. We tested random obstacle coverage percentages of 7%, 13%, and 20%. These obstacles were placed randomly throughout the environment, and did not frequently result in "box canyons", or other difficult geometrical configurations. In adding these obstacles, we assumed only that the robot navigation capabilities were restricted by the obstacles. We did not consider restrictions in the visibility of targets caused by obstacles blocking the robots' views. Under this scenario, robots were able to see over and/or around the obstacles.

Physical Robot Experimental Results To illustrate the typical behavior of the physical robots, we first present an experiment using human joysticking. In Figure 12, the targets are indicated by the triangular flags. The first frame in Figure 12 shows the arrangement of the observers and targets at the very beginning of experiment. The second frame (top right) shows how the two observers move away from each other once the experiment is begun, due to the repulsive forces between the observers. In the third frame, a human joysticks one of the targets away from the other target and the observers. As the target is moved, the two observers also move in the same direction, due to the attractive forces of the target that is moving away. However, if the target exits the area of interest, $\mathcal{S}$, as illustrated in the

# observers \ # targets	1	2	3
1	*Fixed, Local*	*Fixed, Local*	*Fixed, Local*
2	*Fixed, Local, A-CMOMMT*	*Fixed, Local, A-CMOMMT* *	
3	*Fixed, Local, A-CMOMMT*		

Table 5. Tabulation of physical robot experiments conducted. All of the cases indicated were averaged over ten runs of ten minutes each without obstacles. In addition, the case marked '*' is the situation for which we also ran experiments with three different percentages of scattered random obstacles.

Fig. 12. Results of robot team performing task using summation of force vectors. The robots with the triangular flags are acting as targets, while the robots without the flags are performing the distributed observation.

fourth frame, then the observers are no longer influenced by the moved target, and again draw nearer to the stationary target, due to its attractive forces. Note that throughout the example, the observers keep away from each other, due to the repulsive forces.

Figures 13 and 14 show typical scenarios of the experimentations both without and with scattered random obstacles. The results of the experimental runs of the scenarios listed in Table 5 (without obstacles) are shown in Figures 15 and 16. In Figure 15, the results show the relative performances of the *Fixed*, *Local*, and *A-CMOMMT* approaches, measured in terms of the normalized average value of the A metric, as a function of the n/m ratio (i.e., numbers of targets to numbers of robots). To be consistent with our simulation results, we varied the effective sensing range of the robots (through software) within a fixed experimental area, so as to provide results as a function of the amount of experimental area the robots can observe. Figure 15 shows results for three sensing ranges – 2 meters, 4 meters, and 6 meters. As these results show, the *A-CMOMMT* approach enabled robots to track all targets for all these experimental scenarios. The *Local* approach also worked well except for the short sensing range (2 meters) with more targets than robots. The *Fixed* approach was effective only when the sensing range was extended to 6 meters.

Fig. 13. Robot team executing *A-CMOMMT* approach in area with no obstacles.

Fig. 14. Robot team executing *A-CMOMMT* approach in area with obstacles in the work area.

Figure 16 presents these experimental results in terms of the average distance between a target and the closest robot, rather than in terms of the normalized average A metric value. These results show the same relative performance of the three approaches, but show the actual distance data as well as the standard deviations in the robot performance for each experimental scenario. These data are consistent with the simulation results, showing the near equivalence of the *A-CMOMMT* and *Local* approaches for small ratios of n/m, but the improved performance of *A-CMOMMT* approach for larger n/m ratios. While we cannot prove that the simulation results would hold for larger numbers of physical robot and target experiments, it is expected that results similar to the simulation findings would continue to hold true for larger numbers of physical robots and targets.

Figure 17 presents the results of our experiments that included randomly scattered simple obstacles that covered up to 20% of the experimental area. As these results show, because the obstacles impeded both the robots and the targets, and because robots were enabled to see over and/or around obstacles, none the control approaches were significantly affected by the addition of scattered obstacles.

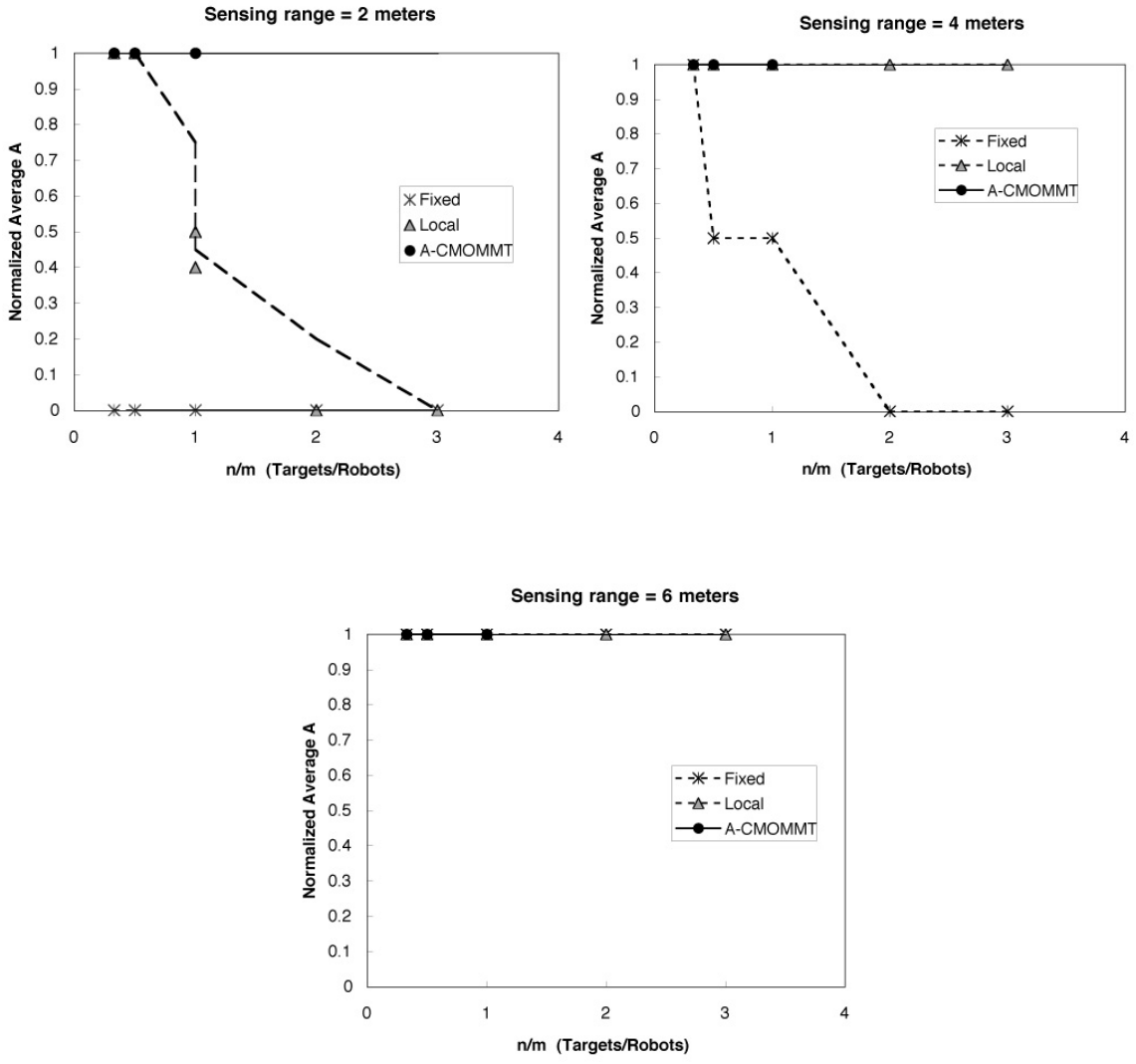


Fig. 15. Results of physical robot experiments with no obstacles. The three graphs show the results of the *Fixed*, *Local*, and *A-CMOMMT* approaches for different sensing ranges within a fixed-size experimental area, as a function of n/m.

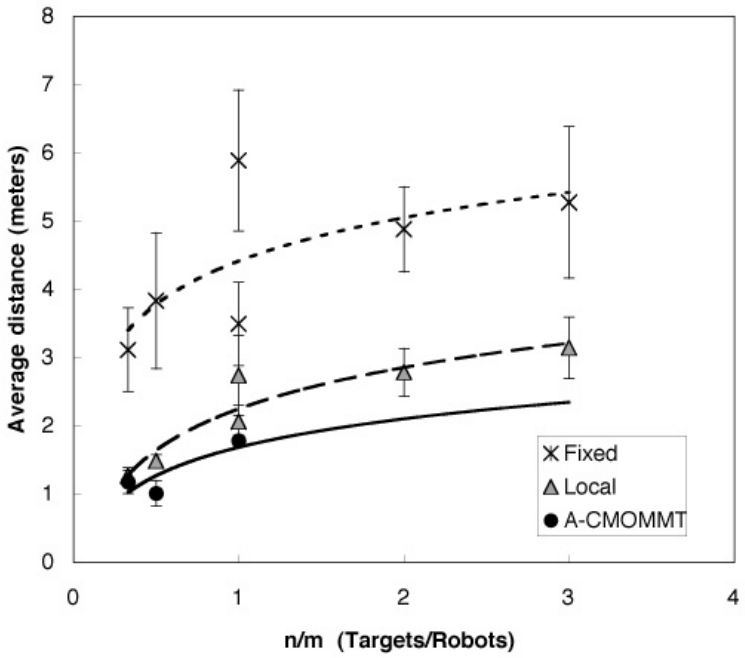


Fig. 16. Results of physical robot experiments with no obstacles. Each data point represents the average of 10 experimental runs of ten minutes each for a given ratio of targets to robots. Standard deviations for each 10-run average are shown. The two data points for each of *Fixed* and *Local* at $n/m = 1$ represent the two cases of 1) one robot and one target, and 2) two robots and two targets.

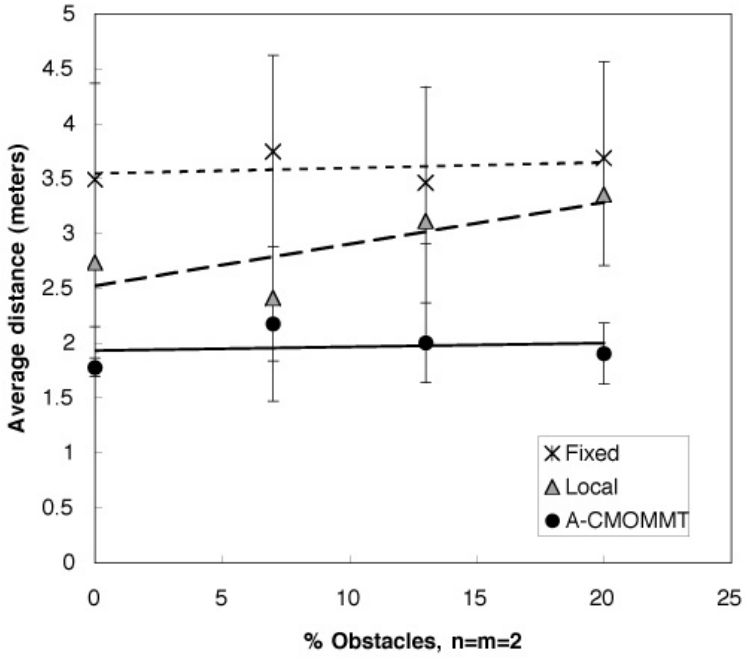


Fig. 17. Results of physical robot experiments with scattered random obstacles. Each data point represents the average of 10 experimental runs of ten minutes each for two robots and two targets (thus n/m in this case equals 1).

6. Multi-Robot Learning of CMOMMT Solutions

We believe that the CMOMT application is an excellent domain for multi-robot learning because it is an inherently cooperative task. in these types of tasks, the utility of the action of one robot is dependent upon the current actions of the other team members. Inherently cooperative tasks cannot be decomposed into independent subtasks to be solved by a distributed robot team. Instead, the success of the team throughout its execution is measured by the combined actions of the robot team, rather than the individual robot actions. This type of task is a particular challenge in multi-robot learning, due to the difficulty of assigning credit for the individual actions of the robot team members.

Researchers have recognized that an approach with more potential for the development of cooperative control mechanisms is autonomous learning. Hence, much current work is ongoing in the field of multi-agent learning (e.g., [38]). The types of learning applications that are typically studied for multi-agent and multi-robot systems vary considerably in their characteristics. Some of the learning application domains include air fleet control [34], predator/prey [4], [17], [14], box pushing [22], foraging [24], and multi-robot soccer [35], [23], [16].

Of the previous application domains that have been studied in the context of multi-robot learning, only the multi-robot soccer domain addresses inherently cooperative tasks with more than two robots while also addressing the real-world complexities of embodied robotics, such as noisy and inaccurate sensors and effectors in a dynamic environment that is poorly modeled. One aspect of CMOMMT that is different from multi-robot soccer is that it raises the issue of scalability much more than in multi-robot soccer. The issue of dealing with larger and larger numbers of robots and targets must be taken into account in the design of the cooperative control approach[4]. Thus, we feel that these characteristics make the CMOMMT domain worthy of study for multi-robot cooperation, learning, and adaptation.

We are currently developing multi-robot learning approaches to this problem. In [33], we report on an approach that addresses the CMOMMT problem without the assumption of an *a priori* model. This approach combines reinforcement learning, lazy learning, and a pessimistic algorithm able to compute for each team member a lower bound on the utility of executing an action in a given situation. Lazy learning [1] is used to enable robot team members to build a memory of situation action pairs through random exploration of the CMOMMT problem. A reinforcement function gives the utility of a given situation, providing positive feedback if new targets are acquired and negative feedback if targets are lost. The pessimistic algorithm for each robot then uses the utility values to select the action that maximizes the lower bound on utility. This selection is performed by finding the lower bounds of the utility value associated with the various potential actions that may be conducted in the current situation, and then choosing the action with the greatest utility. The resulting Pessimistic Lazy Q-Learning algorithm is able to perform considerably better than the *Random* action policy, although it is still significantly inferior to the human-engineered *A-CMOMMT* algorithm described in this paper. This reduced performance likely due to the fact that this algorithm only takes into account nearby targets, ignoring nearby robots. Our ongoing research is aimed at improving this learning approach to include information on nearby robots in the reinforcement function.

We have also explored a second approach to learning [11] that is also based on reinforcement learning. This approach applies Q-Learning along with the VQQL [12] technique and the Generalized Lloyd Algorithm [21], [20] to address the generalization issue in reinforcement learning. This approach allows a dramatic reduction in the number of states that would be needed if a uniform representation of the environment of the robots were used, thus allowing a better use of the experience obtained from the environment. Thus, useful information can be stored in the state space representation without excessively increasing the number of states needed. This approach was shown to be highly successful in reaching performance results almost as good as the *A-CMOMMT* approach described in this paper.

Our ongoing research is aimed at exploring other approaches to multi-robot learning in this domain. Ideally, we would like to develop a learning approach that can discover solutions that are better than the human-engineered *A-CMOMMT* approach described in this paper. Our ultimate objective in this

research is to develop general techniques for multi-robot learning that apply not only to the CMOMMT domain, but also to a wide variety of challenging application domains.

7. Conclusions

Many real-world applications in security, surveillance, and reconnaissance tasks require multiple targets to be monitored using mobile sensors. We have examined a version of this problem, called *Cooperative Multi-Robot Observation of Multiple Moving Targets (CMOMMT)*, and have presented distributed heuristic approach, called *A-CMOMMT*, to this problem. This approach is based upon local force vectors that cause robots to be attracted to nearby targets, and to be repelled from nearby robots. The force vectors are weighted to cause robots to be less attracted to targets that are also under observation by other nearby robots.

We have compared the *A-CMOMMT* approach to three other possible policies: *Fixed*, in which robots remain at uniformly distributed locations in the area, *Random*, in which robots move according to a random/linear motion, and *Local*, which uses non-weighted local force vectors. We performed extensive studies using both simulated and physical robots and presented results that took into account the number of robots, the number of targets, the sensing range of the robots, and the size of the observation area $\mathcal{S}$. We also examined two types of target movements — random/linear motion and evasive motion. Using the physical robots, we also explored the impact of randomly scattered simple convex obstacles.

Our results show that the *A-CMOMMT* approach is significantly superior to the *Local* approach for the difficult (but typical of the problems we are interested in) situations in which there are more targets than robots. Thus, *A-CMOMMT* is successful in achieving its goal of maximizing the observation of targets in the more challenging instantiations of the *CMOMMT* problem. On the contrary, for experiments in which there are many more robots than targets, the weights on the local force vectors used in the *A-CMOMMT* approach cause robots to occasionally lose targets, and thus perform worse than in the *Local* approach. Therefore, practical use of the *A-CMOMMT* approach will require additional information on the relative numbers of targets versus robots to dynamically change the weights of the force vectors.

We also introduced our ideas in using CMOMMT as a multi-robot learning domain and our research in developing learning approaches to solve this problem. We are continuing this research, to explore the possibility of automatically generating solutions to challenging multi-robot control problems such as *CMOMMT*.

Acknowledgements

The author expresses her thanks to visiting students Brad Emmons, Paul Eremenko, and Pamela Murray, who helped in the implementation and/or data collection of the physical robot experimentations. Many thanks also to Claude Touzet for his discussions and suggestions during the development of this research. The author is also very appreciative of the three anonymous reviewers who provided many helpful suggestions for improving an earlier draft of this article. This research is sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, U. S. Department of Energy. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U. S. Government purposes. Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the U.S. Dept. of Energy under contract DE-AC05-00OR22725.

Notes

1. Using the 1.6 Mbps Proxim radio ethernet system we have in our laboratory, and assuming messages of length 10 bytes per robot per target are transmitted every 2 seconds, we find that nm (for n targets and m robots) must be less than 4×10^4 to avoid saturation of the communication bandwidth.

2. We implement this in our laboratory using the 2D Conac Positioning System by MTI Research, Inc.
3. With precise calibration, laser positioning, and tuning, the MTI Conac System is capable of providing higher accuracy positioning. However, the 10 centimeter accuracy was sufficient for this application.
4. We assume that the game of soccer does not involve arbitrarily large numbers of players.

References

1. D. Aha, editor. *Lazy Learning*. Kluwer Academic Publishers, 1997.
2. Y. Bar-Shalom. Tracking methods in a multitarget environment. *IEEE Transactions on Automatic Control*, AC-23(4):618–626, 1978.
3. Y. Bar-Shalom. *Multitarget Multisensor Tracking: Advanced Applications*. Artech House, 1990.
4. M. Benda, V. Jagannathan, and R. Dodhiawalla. On optimal cooperation of knowledge sources. Technical Report BCS-G2010-28, Boeing AI Center, August 1985.
5. P. Bernhard. Computation of equilibrium points of delayed partial information pursuit evasion games. In *Proceedings of the 26th Conference on Decision and Control*, December 1987.
6. P. Bernhard and A.-L. Colomb. Saddle point conditions for a class of stochastic dynamical games with imperfect information. *IEEE Transactions on Automatic Control*, 33(1):98–101, 1988.
7. S. S. Blackman. *Multitarget Tracking with Radar Applications*. Artech House, 1986.
8. A. J. Briggs. *Efficient Geometric Algorithms for Robot Sensing and Control*. PhD thesis, Cornell University, 1995.
9. E. H. Durfee, V. R. Lesser, and D. D. Corkill. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, C-36:1275–1291, 1987.
10. H. R. Everett, G. A. Gilbreath, T. A. Heath-Pastore, and R. T. Laird. Coordinated control of multiple security robots. In *Proceedings of SPIE Mobile Robots VIII*, pages 292–305, 1993.
11. F. Fernandez and L. E. Parker. Learning in large cooperative multi-robot domains. *to appear in International Journal of Robotics and Automation*, 2001.
12. Fernando Fernndez and Daniel Borrajo. VQQL. Applying vector quantization to reinforcement learning. In *RoboCup-99: Robot Soccer World Cup III*. Springer Verlag, 2000.
13. G.C. Fox, R.D. Williams, and P.C. Messina. *Parallel Computing Works*. Morgan Kaufmann, 1994.
14. Thomas Haynes and Sandip Sen. Evolving behavioral strategies in predators and prey. In Gerard Weiss and Sandip Sen, editors, *Adaptation and Learning in Multi-Agent Systems*, pages 113–126. Springer, 1986.
15. MTI Research Inc. Conac 3-D tracking system. Operating manual, Chelmsford, MA, 1995.
16. H. Kitano, Y. Kuniyoshi, I. Noda, M. Asada, H. Matsubara, and E. Osawa. Robocup: A challenge problem fir AI. *AI Magazine*, 18(1):73–85, Spring 1997.
17. R. Korf. A simple solution to pursuit games. In *Working Papers of the 11th International Workshop on Distributed Artificial Intelligence*, pages 183–194, 1992.
18. S. M. LaValle, H. H. Gonzalez-Banos, C. Becker, and J-C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Proceedings of the 1997 International Conference on Robots and Automation*, pages 731–736, 1997.
19. S. M. LaValle, D. Lin, L. J. Guibas, J-C. Latombe, and R. Motwani. Finding an unpredictable target in a workspace with obstacles. In *Proceedings of the 1997 International Conference on Robots and Automation*, pages 737–742, 1997.
20. Yoseph Linde, André Buzo, and Robet M. Gray. An algorithm for vector quantizer design. In *IEEE Transactions on Communications, Vol1. Com-28, N 1*, pages 84–95, 1980.
21. S. P. Lloyd. Least squares quantization in pcm. In *IEEE Transactions on Information Theory*, number 28 in IT, pages 127–135, March 1982.
22. S. Mahadevan and J. Connell. Automatic programming of behavior-based robots using reinforcement learning. In *Proceedings of AAAI-91*, pages 8–14, 1991.
23. S. Marsella, J. Adibi, Y. Al-Onaizan, G. Kaminka, I. Muslea, and M. Tambe. On being a teammate: Experiences acquired in the design of RoboCup teams. In O. Etzioni, J. Muller, and J. Bradshaw, editors, *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 221–227, 1999.
24. Maja Mataric. *Interaction and Intelligent Behavior*. PhD thesis, Massachusetts Institute of Technology, 1994.
25. J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.
26. L. E. Parker. ALLIANCE: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In *Proc. of the 1994 IEEE/RSJ/GI Int'l Conf. on Intelligent Robots and Systems (IROS '94)*, pages 776–783, Munich, Germany, Sept. 1994.
27. L. E. Parker. An experiment in mobile robotic cooperation. In *Proceedings of the ASCE Specialty Conference on Robotics for Challenging Environments*, Albuquerque, NM, February 1994.
28. L. E. Parker. Multi-robot team design for real-world applications. In *Distributed Autonomous Robotic Systems 2*, pages 91–102. Springer-Verlag Tokyo, 1996.
29. L. E. Parker. On the design of behavior-based multi-robot teams. *Journal of Advanced Robotics*, 1996.
30. L. E. Parker. ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

31. L. E. Parker. Distributed control of multi-robot teams: Cooperative baton-passing task. In *Proceedings of the 4th International Conference on Information Systems Analysis and Synthesis (ISAS '98)*, volume 3, pages 89–94, 1998.
32. L. E. Parker. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing, special issue on Robotics Research at Oak Ridge National Laboratory*, 5(1):5–19, 1999.
33. L. E. Parker and C. Touzet. Multi-robot learning in a cooperative observation task. In *Distributed Autonomous Robotic Systems 4*, pages 391–401. Springer, 2000.
34. Randall Steeb, Stephanie Cammarata, Frederick Hayes-Roth, Perry Thorndyke, and Robert Wesson. Distributed intelligence for air fleet control. Technical Report R-2728-AFPA, Rand Corp., 1981.
35. P. Stone and M. Veloso. A layered approach to learning client behaviors in the robocup soccer server. *Applied Artificial Intelligence*, 12:165–188, 1998.
36. K. Sugihara, I. Suzuki, and M. Yamashita. The searchlight scheduling problem. *SIAM Journal of Computing*, 19(6):1024–1040, 1990.
37. Ichiro Suzuki and Masafumi Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal of Computing*, 21(5):863–888, 1992.
38. Gerhard Weiss and Sandip Sen, editors. *Adaption and Learning in Multi-Agent Systems*. Springer, 1996.
39. R. B. Wesson, F. A. Hayes-Roth, J. W. Burge, C. Stasz, and C. A. Sunshine. Network structures for distributed situation assessment. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1):5–23, 1981.