Cooperative Robotics for Multi-Target Observation

Lynne E. Parker
Center for Engineering Systems Advanced Research (CESAR)
Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6355

Abstract

An important issue that arises in the automation of many security, surveillance, and reconnaissance tasks is that of observing (or monitoring) 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 article, we investigate the use of a cooperative team of autonomous sensor-based robots for the observation of multiple moving targets (a problem that we term *CMOMMT*). We focus primarily on developing the distributed control strategies that allow the robot team to attempt to maximize the collective time during which each target is being observed by at least one robot team member in the area of interest. Our initial efforts on this problem address the aspects of distributed control in robot teams with equivalent movement capabilities working in an uncluttered, bounded area. This article first formalizes the problem and discusses related work. We then present a distributed approximate approach to solving this problem (called *A-CMOMMT*) that combines low-level multi-robot control with higher-level control. The low-level control is described in terms of force fields emanating from the targets and the robots. The higher level control is presented in our ALLIANCE formalism [16, 17], which provides mechanisms for fault tolerant cooperative control, and allows robot team members to adjust their low-level actions based upon the actions of their teammates. We then present the results of the ongoing implementation of our approach, both in simulation and on physical robots. To our knowledge, this is the first article addressing this research problem that has been implemented on physical robot teams.

Keywords: multi-robot cooperation, multi-target tracking, ALLIANCE, behavior-based.

1 Introduction

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 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 monitored, 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 fixed-location (static) 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.

In this article, we investigate the use of a cooperative team of autonomous sensor-based robots for applications in this domain. We focus primarily on developing the distributed control strategies that allow the team to attempt to maximize the collective time during which each target is being observed by at least one robot team member in the area of interest.

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 2D) 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, IR, 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 homogeneous robot teams with equivalent sensing and movement capabilities working in an uncluttered, bounded area.

We also note that although the cooperative multi-robot target observation application is interesting in its own right, this application domain can also serve as a testbed for developing generalized approaches for the control of cooperative teams. The cooperative monitoring (or observation) problem is attractive for this purpose for at least two reasons. First, it requires a *strongly cooperative* solution [7] to achieve the goal, meaning intuitively that the robots must act in concert to achieve the goal, and that the task is not trivially serializable. This makes the cooperative control problem much more challenging than a *weakly cooperative* approach. And, second, it allows us to explore the extension of our ALLIANCE cooperative control architecture [17, 18] that we previously developed for the domain of loosely-coupled, independent tasks, to the domain of strongly cooperative applications.

In this article, we describe a mechanism for achieving distributed cooperative control in the defined application domain. Section 2 defines the multi-target observation problem of interest in this article, and is followed by a discussion of related work in section 3. Section 4 describes our approach, discussing each of the subcomponents of the system. Section 5 describes the implementation of our approach on both a simulated and a physical robot team. Finally, we offer concluding remarks in section 6, as well as directions of continuing and future research.

2 Problem Description: *CMOMMT*

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

$\mathcal{S}$:	a two-dimensional, bounded, enclosed spatial region, with entrances/exits
$\mathcal{R}$:	a team of m robots with 360^0 field of view observation sensors, that are noisy and of limited range
$In(o_j(t), \mathcal{S})$:	a binary variable defined to be *true* when target $o_j(t)$ is located within region $\mathcal{S}$ at time t
$\mathcal{O}(t)$:	a set of n targets, $o_j(t)$, $j = 1, 2, ..., n$, such that $In(o_j(t), \mathcal{S})$ is true

Define an $m \times n$ matrix $A(t)$, where

$$a_{ij}(t) = \begin{cases} 1 & \text{if robot } r_i \text{ is } \textit{observing} \text{ target } o_j(t) \text{ in } \mathcal{S} \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

We further define the *logical OR* operator over a vector H of k elements as:

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

We say that a robot is *observing* a target when the target is within that robot's *sensing range* (which is defined explicitly in section 4.1.1). Then, the goal is to develop an algorithm, which we will call *A-CMOMMT*, that maximizes the following:

$$\sum_{t=0}^{T} \sum_{j=1}^{n} \bigvee_{i=1}^{m} a_{ij}(t)$$

over time steps Δt under the assumptions listed below. In other words, the goal of the robots is to maximize the collective time during which each target in $\mathcal{S}$ is being observed by at least one robot during the mission from $t = 0$ to $t = T$. Note that we do not assume that the membership of $\mathcal{O}(t)$ is known in advance (i.e., the movements of the targets are unknown in advance).

In addressing this problem, we assume the following: Define $sensor\_coverage(r_i)$ as the area visible to robot r_i's observation sensors, for $r_i \in \mathcal{R}$. (Note that the *sensor_coverage* of a sensor is dependent upon both its range (defined as *sensing range* in section 4.1.1) and its field of view (i.e. the angle subtended by the sensor).) Then we assume that, in general,

$$\bigcup_{r_i \in \mathcal{R}} sensor\_coverage(r_i) \ll \mathcal{S}.$$

That is, the maximum area covered by the observation sensors of the robot team is much less than the total area to be monitored. 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 observational contact with them and to maximize the coverage of the area $\mathcal{S}$.

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. 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 $r_i \in \mathcal{R}$ and for all $o_j(t) \in \mathcal{O}(t)$, $max\_v(r_i) > max\_v(o_j(t))$, where $max\_v(a)$ returns the maximum possible velocity of entity a, for $a \in \mathcal{R} \cup \mathcal{O}(t)$.

- Targets in $\mathcal{O}$ can enter and exit region $\mathcal{S}$ through distinct entrances/exits on the boundary of $\mathcal{S}$.

- The robot team members share a known global coordinate system.

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 of the case of an omni-directional 2D sensory system (such as a ring of cameras or sonars), in which the robot sensory system is of limited range, but is available for the entire 360^o around the robot, as depicted in figure 1.

[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 must be less than 4×10^4 bps to avoid saturation of the communication bandwidth. Thus, the upper limit of the total allowable number of robots and targets is about 400.

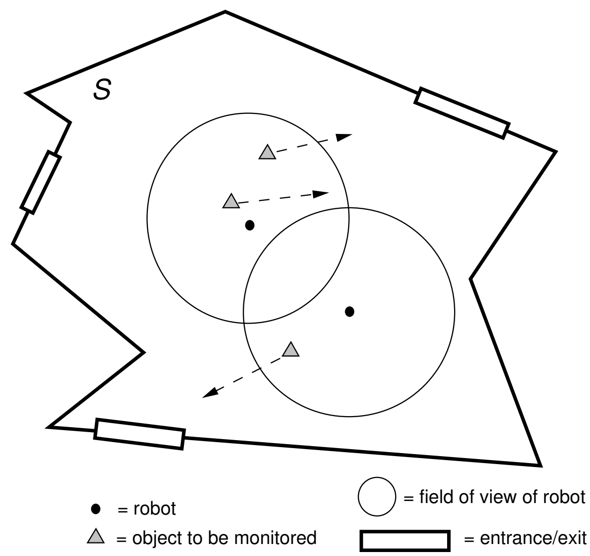

Figure 1: The problem depicted in terms of omni-directional 2D robot sensors.

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. While a complete review of these fields is not within the scope of this article, we will briefly outline the most relevant previous work in these areas.

The work most closely related to the *CMOMMT* problem falls into the category of the *art gallery* and related problems [15], 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. Most research in this area typically utilizes centralized approaches to the placement of sensors, uses ideal sensors (noise-free and infinite range), and assumes the availability of sufficient numbers of sensors to cover the entire area of interest. Several authors have looked at the static placement of sensors for target tracking in known polygonal environments. For example, Briggs [6] uses art gallery theorems in the development of algorithms for planning the set of placements from which a sensor can monitor a region within a task environment. Her approach uses *weak visibility* as a model for detectability, in which all points in the area to be monitored are visible from at least one point in the sensor placement region. 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.* [21] 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. Their objective is to determine whether a search schedule exists, given a polygon and the locations of the searchlights. In this context, a search schedule is a mapping from an interval of time to a direction in which the searchlight should aim. They develop certain necessary and sufficient conditions for the existence of a search schedule in certain situations. This work, however, assumes that there is only one target, that the target cannot enter or exit the polygon after the start of the problem, and that the searchers maintain fixed positions. It also does not give a prescriptive algorithm for determining the appropriate search schedule for any given simple polygon, although algorithms for special cases are provided.

Suzuki and Yamashita [22] address the *polygon search* problem, which deals with searching for a mobile target in a simple polygon by a single 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 that the searcher has an infinite sensory range, that the target cannot enter or exit the polygon after the start of the problem, and that only one searcher is available. It also does not give a prescriptive algorithm for determining the appropriate search schedule for the single searcher for any given simple polygon, although algorithms for special cases are provided.

LaValle *et al.* [13] introduces 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. In LaValle *et al.* [12], they address the related question of maintaining the visibility of a moving target in a cluttered workspace by a single robot. They are also able to optimize the path along additional criteria, such as the total distance traveled. The problems they address in these articles are closely related to the problem of interest here. The primary difference is that their work does not deal with multiple robots maintaining visibility of multiple targets, nor a domain in which targets may enter and exit the area of interest.

Another large area of related research has addressed the problem of multi-target tracking (e.g. Bar-Shalom [1, 2], Blackman [5], Fox *et al.* [10]). 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 [3, 4], which is the problem of determining where to shoot to minimize the survival probability of the rabbit. Our task in this article differs from these works in that our goal is not to calculate the trajectories of the targets, but rather to find dynamic sensor placements that maximize the collective time that each target is being observed by at least one of the mobile sensors.

In the area of multi-robot surveillance, Everett *et al.* [9] 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.* [23] 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.* [8] 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 in areas in which targets may enter and exit.

4 Approach

Figure 2 shows the overall design of the control system within each robot team member. This design is based upon our ALLIANCE architecture [17, 18], which facilitates the fault tolerant cooperative control of multiple robot teams. We now provide a brief overview of ALLIANCE, and then describe how we use this approach to develop the overall control system for robots performing the *CMOMMT* application. The following subsections describe the subsystems in more detail.

The ALLIANCE software architecture is a behavior-based, fully distributed architecture that utilizes adaptive action selection to achieve fault tolerant cooperative control. Robots under this architecture possess a variety of high-level functions (modeled as *behavior sets*) that they can per-

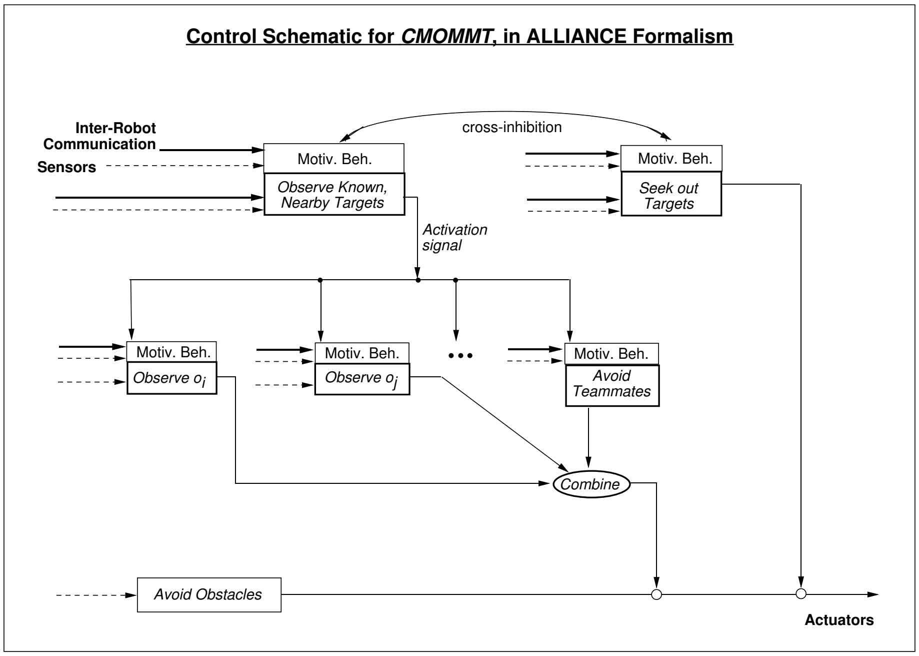

Figure 2: Control within an individual robot for the *CMOMMT* mission, in our ALLIANCE formalism.

form during a mission, and must at all times select an appropriate action based on the requirements of the mission, the activities of other robots, the current environmental conditions, and their own internal states. Since cooperative robotic teams often work in dynamic and unpredictable environments, this software architecture allows the team members to respond robustly and reliably to the learning of new skills and to unexpected environmental changes and modifications in the robot team that may occur due to mechanical failure or the addition or removal of robots from the team by human intervention. This is achieved through the interaction of mathematically modeled motivations of behavior, such as impatience and acquiescence, within each individual robot. These motivations allow a robot to take over a task from any other team member if that team member does not demonstrate its ability — through its effect on the world — to accomplish its task. Similarly, it allows a robot to give up its own current task if its sensory feedback indicates that adequate progress is not being made to accomplish that task. The primary mechanism for achieving adaptive action selection in this architecture is the *motivational behavior*. The output of a motivational behavior is typically the activation level or importance weighting of its corresponding behavior set, represented as a non-negative number. When the current level of activation of a behavior set crosses a threshold, that behavior set becomes active, and all other behavior sets are inhibited from activation. This results in the robot performing no more than one high-level function at a time. Thus, ALLIANCE is superior to a simple subsumption approach for those applications that require higher-level reasoning to determine which behavior to activate.

In the *CMOMMT* problem shown in figure 2, each robot has two high-level behavior sets: *Observe Known, Nearby Targets* and *Seek Out Targets*. The *Observe Known, Nearby Targets* behavior set in turn controls a number of additional behavior sets (called *Observe* o_k) for the observation of individual targets. In figure 2, the motivational behaviors are indicated by the small rectangle attached at the top of the behavior sets. The following subsections describe these behaviors in more detail.

4.1 Observe Known, Nearby Targets

The *Observe Known, Nearby Targets* behavior set is responsible for controlling robot r_i's movements in relationship to other nearby targets and nearby robots. This part of the control scheme is modeled by a collection of lower-level behavior sets and motivational behaviors (as shown in figure 2), each of which is spawned automatically when a robot has become aware of a target nearby. The motivational behaviors in this subsystem are responsible for determining the weight, or importance, of robot r_i's continued observation of target o_i. If any target o_j leaves robot r_i's *predictive tracking range* (defined in the next subsection), the corresponding behavior set is terminated by its respective motivational behavior. The generated weights are then factored into the output of the *Observe Known, Nearby Targets* behavior set (described below) to calculate the desired direction of motion of robot r_i. This combination of information is modeled in figure 2 as the *combine* module.

The following subsections describe how the local control information based upon robot and target locations is derived, how the motivational behaviors derive the weights corresponding to each target, and how the lower-level and higher-level information is combined.

4.1.1 Target and Robot Detection

Ideally, the robots would be able to passively observe 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 the robots use a global positioning system (such as the satellite-based GPS for outdoors, or the laser-based MTI indoor positioning system [11] that is in use at our CESAR laboratory) to determine their own position, and communicate this information to other robot team members. In our approach, robots do not store position information for robots that are not relatively close (made explicit below).

In addition to robot position information, team members need to determine the positions and velocities of the targets within their own field of view. Since previous work [14, 19] has shown that communication and awareness of robot team member actions can significantly improve the quality of a distributed solution for certain task domains, we supplement a robot's knowledge of target movements gained from direct sensing (e.g. from its cameras or sonar) with position and derived velocity information on target sightings that is communicated by other robot team members within a given communication range. Thus, targets can be one of two types: directly sensed or "virtually" sensed through predictive tracking. However, a team member does not store position information for targets that are not within its own vicinity. Note that this approach requires the available communication bandwidth to be $O(mn)$, for m robots and n targets (see earlier footnote for the impact of this bandwidth requirement on the size of the problem).

To clarify this idea, figure 3 depicts three ranges that are defined with respect to each robot r_i. The innermost range is the *sensing range* of r_i, within which the robot can use a sensor-based tracking algorithm to maintain targets within its field of view. The middle range is the *predictive tracking range* of the robot r_i, which defines the range in which targets localized by other robots $r_k \neq r_i$ can affect r_i's movements. The outermost range is the *communication range* of the robot, which defines the extent of the robot's communicated messages. (To ground this idea, in our experimentation, the sensing range is on the order of three meters, the predictive tracking range is about twice the sensing range, and the communication range is about five times the sensing range.)

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

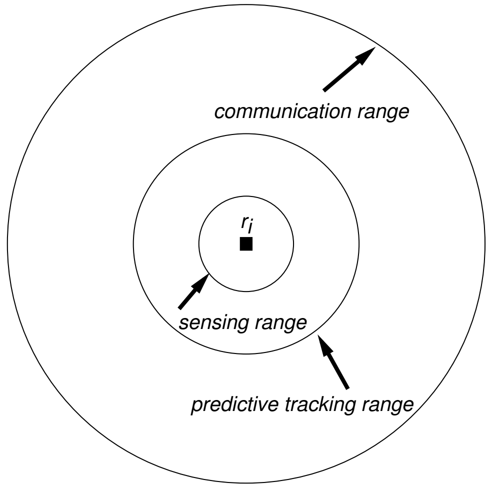

Figure 3: Definition of the *sensing range*, *predictive tracking range*, and *communication range* of a robot. Although the exact range values may change, we assume that the relative ordering of range distances remains the same.

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. (Refer to section 4.1.3 for methods of differentially weighting targets based upon whether they are directly sensed, or are estimated to be present through predictive tracking.)

We assume that if the targets are dense enough that their position estimations do not supply enough information to disambiguate distinct targets, then existing tracking approaches (e.g. Bar-Shalom [2]) should be used to uniquely identify each target based upon likely trajectories.

4.1.2 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. the robots') more sophisticated tracking devices (such as cameras) while remaining dispersed from each other to cover more terrain. The local control of a robot team member is thus based upon a summation of force vectors which are attractive for nearby targets and repulsive for nearby robots. Figure 4 defines the 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 distance between do_2 and do_3 defines the preferred tracking range of a robot from a target[2]. In practice, this range will be set experimentally 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. The robot *sensing range*, defined in figure 3, will lie somewhere between do_3 and the predictive tracking range. The attraction to the target falls off linearly as the distance to the target increases from do_3. The attraction goes to 0 beyond the predicted tracking range, indicating that this target is too far to have an effect on the robot's movements.

Figure 5 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

[2]This force between a robot and a target is slightly different from that reported in our earlier work in [20]. We now define a range of preferred distance values rather than one unique preferred distance, resulting in a better performance.

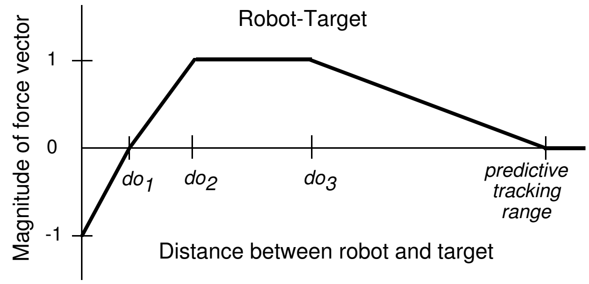

Figure 4: Function defining the magnitude of the force vector of nearby targets.

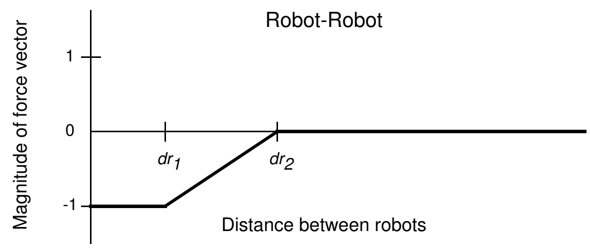

Figure 5: Function defining the magnitude of the force vector of nearby robots.

linearly between these values.

4.1.3 High-Level Control via ALLIANCE

Using only local force vectors for this problem neglects higher-level information that could be used to improve performance. Thus, we now enhance the control approach by adding higher-level control via motivational behaviors to differentially weight the contributions of each target's force field on the total computed field. This higher-level knowledge is expressed in the form of two types of probabilities: the probability that a given target actually exists, and the probability that no other robot is already observing a given target. Combining these two probabilities helps intelligently reduce the overlap of robot sensory areas toward the goal of minimizing the likelihood of a target escaping detection. Figure 6 illustrates the relationships between these probabilities and the sensing and predictive tracking ranges, as well as the general settings of the probabilities in various regions around a robot r_l.

The probability that a target o_k exists according to robot r_l (termed $Pr(exists_{kl})$) is modeled as a decay function based upon when the target was most recently seen, and by whom. In general, a robot will trust its own recent measurements within its sensing range more than it will trust (1) the predictions of target locations within its predictive tracking range but outside its sensing range,(2) the target location measurements made by other robots, or (3) its own older measurements. The probability will also be dependent upon the characteristics of the sensors used to detect the target; in general, this probability will decrease inversely with distance from the sensor, under the assumption that sensor uncertainty increases with distance from the sensor. Beyond the predictive tracking range of the robot, the probability becomes zero.

The probability that no robot other than r_l is already observing a nearby target o_k (termed $Pr(NT_{kl})$) is based upon target o_k's position and the location of nearby robots. If robot r_l knows that another robot r_j is nearby, and is likely within sensing range[3] of target o_k, then $Pr(NT_{kl})$

[3]Note that we make no assumption that the sensing ranges of all robots are the same, nor that robots are aware of their teammate's sensing ranges.

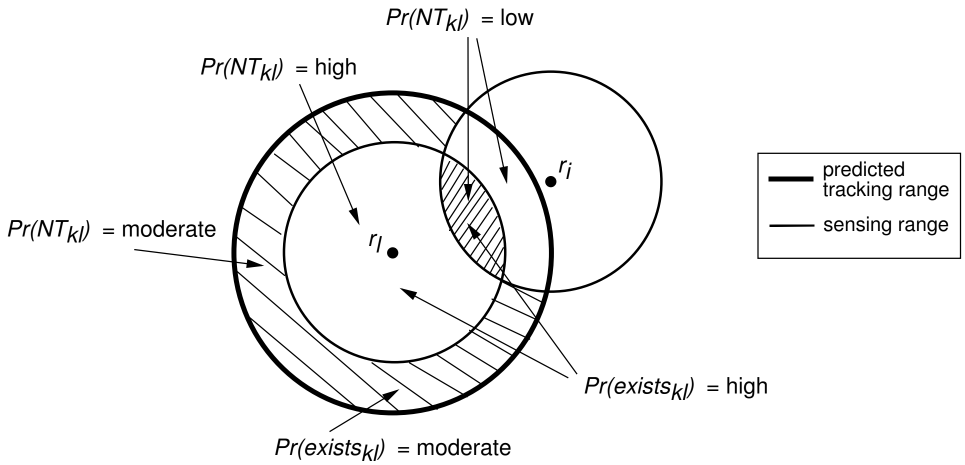

Figure 6: Two probabilities are used to add high-level control to improve performance over local control alone.

should usually be low. In the simplest case, since we define (in section 2) a robot r_l to be *observing* a target o_k when it is within r_l's sensing range, we could assign $Pr(NT_{kl})$ to be zero whenever another robot is within sensing range of o_k. However, we do not want a robot r_l to completely ignore any nearby target, since r_l will be unaware of targets on the far side of robot r_j that may also influence r_j's motion. Thus, we set $Pr(NT_{kl})$ to some non-zero value.

The proper setting of $Pr(NT_{kl})$ 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. We have not yet conducted an extensive exploration of the proper computation of these probabilities based upon these issues. This will be the basis of future work.

The output of the motivational behavior corresponding to a given target is the product of the probability that the target exists and the probability that no other robot is currently observing that target. These probabilities have the effect of causing a robot to prefer the observation of certain targets over others.

4.1.4 Combination of Local and Higher-Level Information

The local force vectors are combined with the higher-level information, resulting in the commanded direction of robot movement. The direction of movement for robot r_l is given by:

$$\sum_{k=1}^{n}(FVO_{kl} \times Pr(exists_{kl}) \times Pr(NT_{kl})) + \sum_{i=1, i\neq l}^{m} FVR_{il}$$

where FVO_{kl} is the force vector attributed to target o_k by robot r_l and FVR_{il} is the force vector attributed to robot r_i by robot r_l. The summation of the weighted force vectors yields an x, y coordinate indicating the desired location of the robot at that point in time. 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, as shown in figure 2, the velocity and steering command can be overwritten by the *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 ensuring 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 located equidistant between two targets that move away from the robot at the same rate, sensory noise will prevent the force vectors from cancelling the attraction to zero. Instead, slight deviations in sensing will 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.

4.2 Seek 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.

In the algorithm as described so far, 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 [22] 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.

In our future work, we intend to develop an automated process that allows the robots to generate the appropriate search schedule for a given area, perhaps based upon this earlier work of Suzuki and Yamashita. Our current approach, however, simplifies the task by supplying the robot team members with a human-derived search path through the area S. In practice, the derivation of an adequate ∞-search schedule by hand through the polygonal areas that define the interiors of most buildings appears to be fairly straightforward under the assumptions of Suzuki and Yamashita [22]. More challenging is dealing with multiple targets, multiple robots, and entrances/exits in the polygonal area. We leave this task to future work.

Thus, when no targets are detected by a given robot, that robot moves along the search path looking for targets. Since new targets are more likely to appear near entrances, ideally, the robot would spend a higher percentage of its time near entrances. To prevent the robot's path from being predictable to a knowledgeable target, the robot could randomly select a direction to traverse at each intersection in the search path. If two robots encounter each other moving in the opposite direction along the search path, they reverse directions. Future work will improve this approach to deal with the possibility of oscillations. As soon as targets are detected along the search route, the highest level motivational behaviors switch the robot from seek mode to observe mode, and the weighted force vector algorithm becomes active. At this point, the searching technique no longer influences the robot behavior.

5 Experiments

Our approach to the cooperative multi-robot observation problem has been implemented both in simulation and on a team of four Nomadic Technologies robots. These robots are wheeled vehicles with tactile, infrared, ultrasonic, 2D laser range, and indoor global positioning systems. In addition, the robots are equipped with a voice synthesizer and radio ethernet for inter-robot communication. Nomadic's multi-robot simulator allows us to test and debug our algorithms (written in C) in simulation prior to executing them on the actual robots. The simulator uses sensory error models that incorporate noise into the sensor readings to increase the realism of the experiments. The code generated during the simulation can then be ported directly to the robots for experimentation in the "real world" with relatively minor changes.

In the initial phase of research in this problem, 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 approach, 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.

Figures 7 and 8 illustrate two examples of portions of our approach that have been implemented on the simulated robots — namely, the local force-field control. In these figures, the black points represent targets, and the gray points represent robots.

Figure 7 shows a case where two targets are being tracked by two robots. The first frame begins with the two targets heading towards each other, and each of the robots "following" one of the targets. In the second frame, the targets have passed each other, and the robots meet in the middle. At this point the repulsive force between the two robots takes precedence and pushes them away from each other, causing them to swap targets. In the final two frames, the robots continue to follow the new targets.

Figure 8 shows a case where the targets stay relatively distributed throughout the simulation. The robots tend to hover around the center of the mass of targets; they keep their distance from one another throughout the simulation, due to the repulsive forces.

The local control subsystems have also been ported to, and successfully demonstrated on, our team of 4 mobile robots. Figure 9 shows an example of the robot implementation. In these experiments, we typically designated certain robots to be targets, and other robots as observers. Since we are not dealing with the issues of visual tracking of targets in our current work, using some robots as targets allowed us to take advantage of the global positioning system on the robots to perform "virtual" tracking. Thus, the robots acting as targets were programmed to broadcast their current location to the robot team; this information could then be used by the observers to calculate their desired movements. 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. In figure 9, the robot targets are indicated by the triangular flags.

The first frame in figure 9 shows the arrangement of the observers and targets at the very beginning of the experiment, where both targets lie within the sensing range of each observing robot. The second frame 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 robot 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 fourth frame, then the observers are no longer influenced by the moved target, and again draw

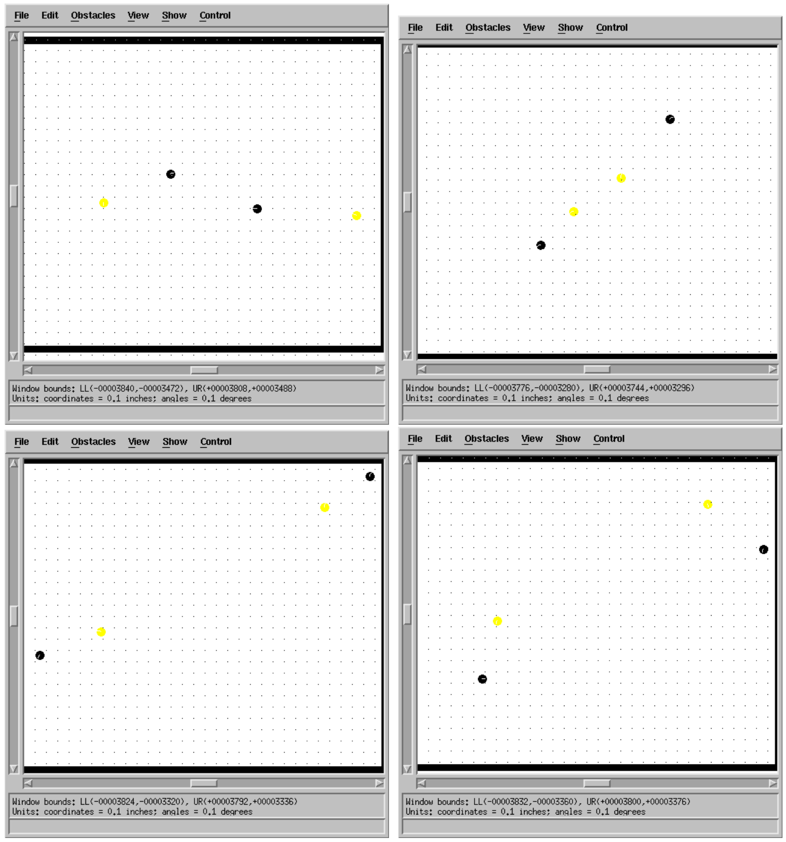

Figure 7: Two targets tracked by two robots performing a swap. The black points represent targets while the gray points represent robots.

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.

To quantitatively evaluate the usefulness of the proposed approach, we have begun comparison of our weighted force vector algorithm (which we will call *A-CMOMMT*) with a random wander approach, in which each robot wanders in arbitrary directions until an obstacle is encountered, at which time it alters its course to avoid the obstacle. For each algorithm, we computed the average distance between each target $o_i(t)$ and the closest robot during the course of the experiment. The environment used in these experiments was a circular room of radius 4000 units, with no additional obstacles other than the walls of the room. The robots were of diameter approximately 200 units. The values of do_1, do_2, do_3, and the predictive tracking range were set to 400, 800, 2600, and 3000 respectively. The robots and the targets were given initial random starting locations in the center of the room, within approximately 300 units of each other. The value of $Pr(exists_{kl})$ was set to 1 within robot r_l's sensing range, and to 0.75 within r_l's predictive tracking range (but outside the sensing range). The value of $Pr(NT_{kl})$ was set to 1 when o_k was within the sensing range of another robot r_i, $i \neq l$. Data was collected for multiple runs, each of which ran for 2 minutes, with discrete distance measurements computed every 2 seconds.

To date, we have collected data for two situations: (a) 1 robot and 1 target, and (b) 3 robots and 3 targets. Figure 10 shows the results of 10 runs of the *A-CMOMMT* algorithm and random wander for 1 robot and 1 target, and the results of 20 runs of the *A-CMOMMT* algorithm and random wander for 3 robots and 3 targets. (These numbers of runs were sufficient to provide statistically

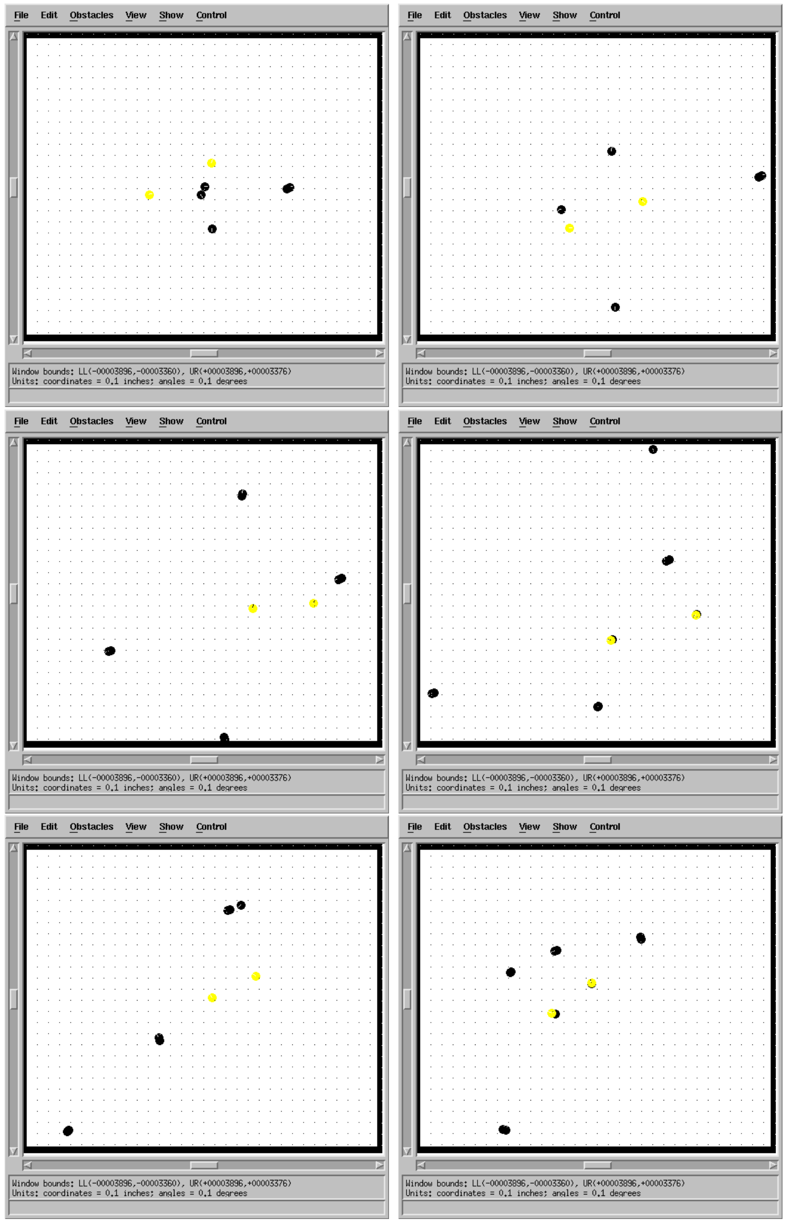

Figure 8: When targets stay relatively distributed, the robots hover around the centroid. (The black points represent targets while the gray points represent robots.)

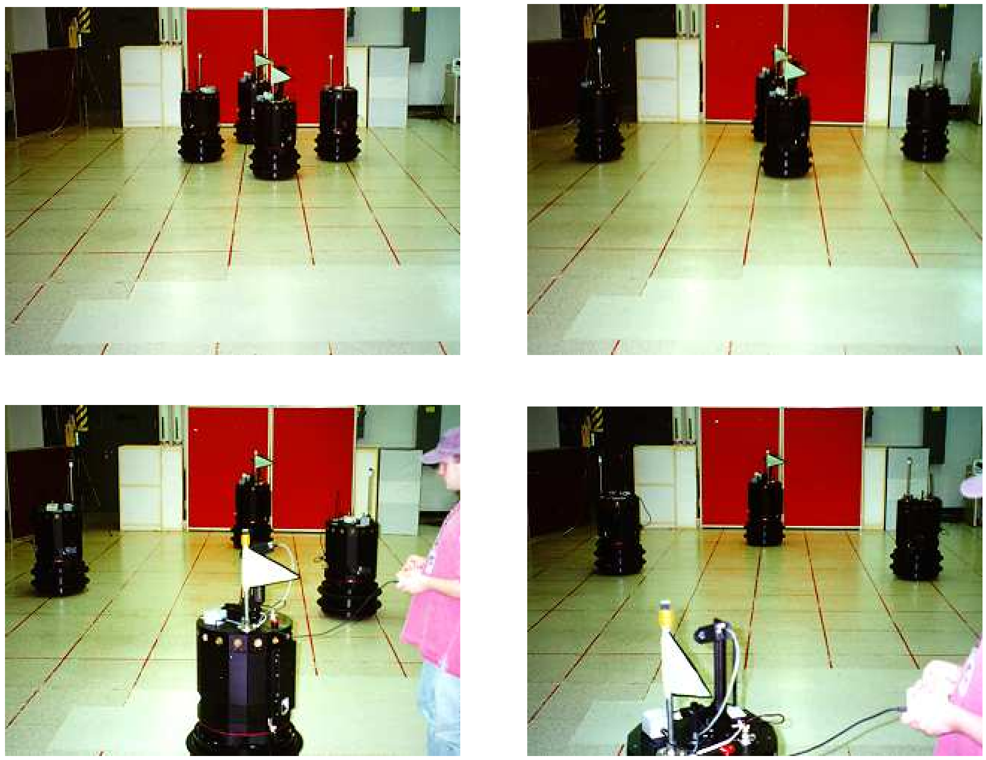

Figure 9: Results of robot team performing task using summation of force vectors. The robots with the triangular flags are acting as targets, while the robots without the flags are performing the distributed observation.

significant results.) These figures show the mean values for runs, as well as one standard deviation.

As expected, this data shows a clear advantage of the *A-CMOMMT* algorithm over the random wander algorithm. In continuing work, we are collecting and analyzing data for a wider range of robot and target numbers, and are implementing the entire control schematic for *A-CMOMMT* on both simulated and physical robot teams.

6 Conclusions and Future Work

Many real-world applications in security, surveillance, and reconnaissance tasks require multiple targets to be observed using mobile sensors. We have defined a problem, called *CMOMMT*, that requires a team of robots to cooperate to maintain observation of multiple targets moving through an area of interest. We have presented a distributed solution, called *A-CMOMMT*, that is based upon high-level control provided through our ALLIANCE formalism, combined with lower-level attractive and repulsive force fields, and a target seeking system. This approach enables the execution of tasks in strongly cooperative application domains. Empirical investigations of portions of our cooperative control approach have been presented on both the simulated robots and the physical robot team, and quantitative data collected indicates the usefulness of our approach. To our knowledge, no previous work related to the *CMOMMT* problem has been implemented on actual robots.

Continuing and future work includes completing the implementation on both the simulated and physical robot teams and the development of an automatic generation of search schedules for times when the observers do not perceive any targets. We also are continuing the collection and analysis of data showing the quantitative usefulness of the *A-CMOMMT* algorithm. Additional related

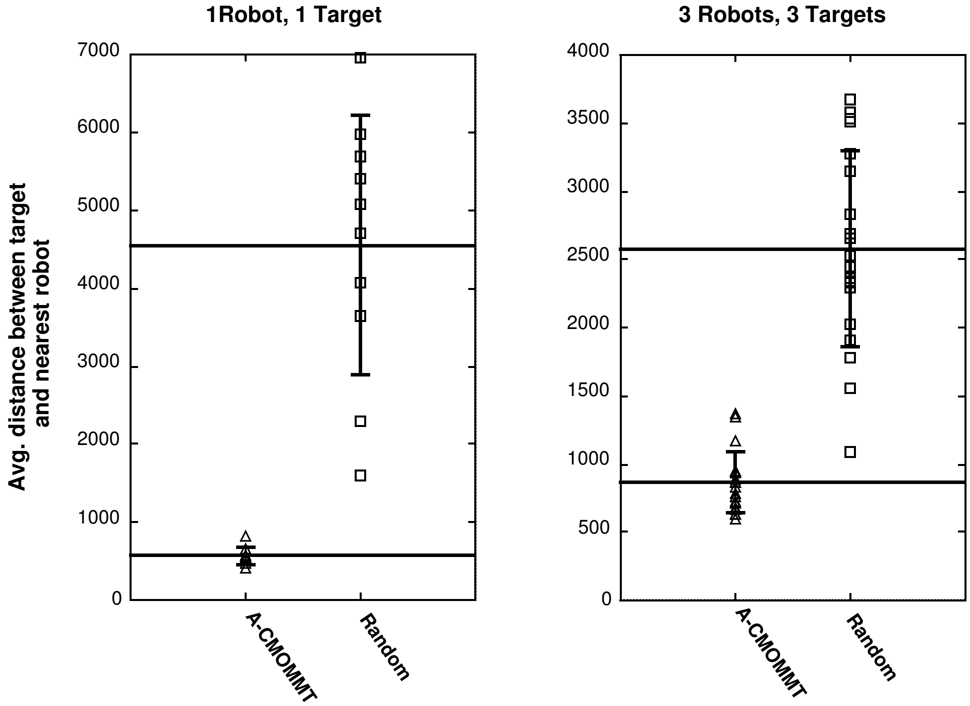

Figure 10: Comparison of *A-CMOMMT* and random wander for two cases.

research includes extending the work to apply to more complex environments, to robots that differ in their sensing and movement capabilities, and to address the subproblems of the physical tracking of targets (e.g. using vision, sonar, IR, or laserrange) and the prediction of target movements.

Acknowledgements Thanks are extended to Brad Emmons, of Albion College, for implementing the local control methodology on the simulated and the physical robot team. Thanks also to the anonymous reviewers for their helpful suggestions and comments. This research was funded by the Office of Engineering Research, Basic Energy Sciences (directed by Dr. Oscar Manley), of the U.S. Department of Energy, under contract No. DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation.

References

[1] Yaakov Bar-Shalom. Tracking methods in a multitarget environment. *IEEE Transactions on Automatic Control*, AC-23(4):618–626, 1978.

[2] Yaakov Bar-Shalom. *Multitarget Multisensor Tracking: Advanced Applications.* Artech House, 1990.

[3] 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.

[4] Pierre Bernhard and Annie-Laure Colomb. Saddle point conditions for a class of stochastic dynamical games with imperfect information. *IEEE Transactions on Automatic Control*, 33(1):98–101, 1988.

[5] S. S. Blackman. *Multitarget Tracking with Radar Applications.* Artech House, 1986.

[6] Amy J. Briggs. *Efficient Geometric Algorithms for Robot Sensing and Control.* PhD thesis, Cornell University, 1995.

[7] Russell G. Brown and James S. Jennings. A pusher/steerer model for strongly cooperative mobile robot manipulation. In *Proceedings of 1995 IEEE International Conference on Intelligent Robots and Systems (IROS '95)*, volume 3, pages 562–568, 1995.

[8] 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.

[9] 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.

[10] G.C. Fox, R.D. Williams, and P.C. Messina. *Parallel Computing Works*. Morgan Kaufmann, 1994.

[11] MTI Research Inc. Conac 3-D tracking system. Operating manual, Chelmsford, MA, 1995.

[12] 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.

[13] 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.

[14] Bruce MacLennan. Synthetic ethology: An approach to the study of communication. In *Proceedings of the 2nd interdisciplinary workshop on synthesis and simulation of living systems*, pages 631–658, 1991.

[15] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[16] L. E. Parker. On the design of behavior-based multi-robot teams. *Journal of Advanced Robotics*, 1996.

[17] Lynne 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.

[18] Lynne E. Parker. *Heterogeneous Multi-Robot Cooperation*. PhD thesis, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, MA, February 1994. MIT-AI-TR 1465 (1994).

[19] Lynne E. Parker. The effect of action recognition and robot awareness in cooperative robotic teams. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '95)*, Pittsburgh, PA, August 1995.

[20] Lynne E. Parker and Brad Emmons. Cooperative multi-robot observation of multiple moving targets. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 2082–2089, Albuquerque, New Mexico, April 1997.

[21] Kzuo Sugihara, Ichiro Suzuki, and Masafumi Yamashita. The searchlight scheduling problem. *SIAM Journal of Computing*, 19(6):1024–1040, 1990.

[22] Ichiro Suzuki and Masafumi Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal of Computing*, 21(5):863–888, 1992.

[23] 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.