Proceedings of the 2007 American Control Conference
Marriott Marquis Hotel at Times Square
New York City, USA, July 11-13, 2007

A Geometric Optimization Approach to Detecting and Intercepting Dynamic Targets

Silvia Ferrari, Chenghui Cai, Rafael Fierro, and Brent Perteet

***Abstract*— A methodology is developed to deploy a mobile sensor network for the purpose of detecting and capturing mobile targets in the plane. The sensing-pursuit problem considered in this paper is analogous to the Marco Polo game, in which the pursuer must capture multiple mobile targets that are sensed intermittently, and with very limited information. In this paper, the mobile sensor network consists of a set of robotic sensors that must track and capture mobile targets based on the information obtained through cooperative detections. Since the sensors are installed on robotic platforms and have limited range, the geometry of the platforms and of the sensors field-of-view play a key role in obstacle avoidance and target detection. Thus, a new cell decomposition approach is presented to formulate the probability of detection and the cost of operating the robots based on the geometric properties of the network. Numerical simulations verify the validity and flexibility of our methodology.**

I. Introduction

The proliferation of reliable, low-cost sensor networks and developments in autonomous vehicle technologies are producing advanced surveillance systems, where both the sensors and their platforms are characterized by a high degree of functionality and reconfigurability. These networks are expected to operate reliably in dynamic environments with little human intervention. However, coordinating such large heterogeneous sensor networks is extremely difficult and requires the development of novel methods of communication, motion control, computation, proactive estimation and sensing, and power management.

Cooperative pursuit strategies to detect, intercept and capture intelligent evaders in cluttered environments are described in [14]. However, the difficulty of solving pursuit-evasion games and obtaining closed-form solutions of the underlying optimization problem has motivated an intensive research aimed at developing algorithmic approaches.

A variety of optimization techniques [4] have been applied to the coordination of robotic networks engaged in distributed sensing tasks [7]. Optimal motion planning for multiple robots is considered in [3]. Distributed motion planning approaches are discussed in [10].

The problem of finding the configuration of a network with multiple sensors that optimizes the number of tracks intercepted is considered in [9]. A Bayesian network (BN) approach is used to develop an automated computer player for CLUE in [5]. In other related work [6] the authors develop a decentralized motion coordination algorithm for tracking tasks of dynamic targets. Strategies to search for moving targets in a two-dimensional plane are considered in [12]. Motion coordination strategies of multiple vehicles visiting targets generated by a stochastic process are proposed in [1].

In this paper, we develop a methodology that employs cell decomposition algorithms to obtain a graph representation of the robot configuration space that is void of obstacle and enables target detections and integrates some of the techniques reviewed. Specifically, the track-coverage definition in [9] and robot motion planning are combined such that multiple mission objectives are considered by the same formalism and optimized by the same policy. The basic problem of moving targets detection and interception is inspired by the game *Marco Polo* described in [13] and is motivated by many applications in surveillance. The goal of Marco Polo is to capture a group of intelligent evaders as quickly as possible using a team of autonomous pursuers that have an intermittent knowledge of the evaders' locations.

The remainder of the paper is organized as follows. We formulate the dynamic tracking problem in Section II. Section III describes the methodology that allows a group of pursuers to detect and intercept dynamic targets. The control strategies for capturing evaders are detailed in Section IV. Section V contains simulation results. Finally, we draw conclusions in Section VI.

II. Problem Statement

We consider a game that takes place in a simply connected, convex polygon $\mathcal{S} \subset \mathbb{R}^2$ with boundary $\partial\mathcal{S}$ and populated by convex obstacles $\mathcal{O}_j \subset \mathcal{S}$.

The pursuers or mobile sensor agents are nonholonomic vehicles that can be modeled using the unicycle model

$$\begin{aligned} \dot{x}_p^i &= v_p^i \cos\theta_p^i, \\ \dot{y}_p^i &= v_p^i \sin\theta_p^i, \\ \dot{\theta}_p^i &= \omega_p^i, \end{aligned} \tag{1}$$

where $q_i = (x_p^i,\ y_p^i,\ \theta_p^i) \in SE(2)$ and $p_i = [x_p^i \quad y_p^i]^T \in \mathbb{R}^2$. The input to pursuer i is $u_p^i = [v_p^i \quad \omega_p^i]^T$ and $u_p \in \mathcal{U} \subset \mathbb{R}^2$.

The model of the target agents or evaders is given by

$$\begin{aligned} \dot{x}_\tau^j &= c_{x_\tau}^j, \\ \dot{y}_\tau^j &= c_{y_\tau}^j, \end{aligned} \tag{2}$$

S. Ferrari and C. Cai are with the Laboratory for Intelligent Systems and Controls, Department of Mechanical Engineering & Materials Science, Duke University, Durham, NC 27708-0300, USA {sferrari, cc88}@duke.edu

R. Fierro and B. Perteet are with the MARHES Lab, School of Electrical & Computer Engineering, Oklahoma State University, Stillwater, OK 74078-5032, USA {rfierro, brent.perteet}@okstate.edu

where $\tau_j = [x^j_\tau \quad y^j_\tau]^T \in \mathbb{R}^2$ denotes the position of target j, and $c^j_{x_\tau}$ and $c^j_{y_\tau}$ are constants. In other words evaders move along straight lines

$$y^j_\tau(t) = \frac{c^j_{y_\tau}}{c^j_{x_\tau}} x^j_\tau(t) + y^j_\tau(0) - \frac{c^j_{y_\tau}}{c^j_{x_\tau}} x^j_\tau(0),$$

where $(x^j_\tau(0),\ y^j_\tau(0)) \in \partial\mathcal{S}$. The heading of a target is θ_j, thus $\theta_j := \arctan(c^j_{y_\tau}, c^j_{x_\tau})$. Let e_{ji} be the Euclidean distance from the j^{th} target position, τ_j, to the closest pursuer, i.e., $e_{ji} = \min d(\tau_j, p_i)$. Then the pursuer i is said to *capture* the target j when $e_{ji} < \varepsilon$. The threshold value ε is called the *capture threshold* for an interval Δ_c called the *capture timeframe*.

In the actual Marco Polo game, when requested by *Marco* (i.e., pursuer) all *Polos* (i.e., evaders) must communicate their positions to *Marco*. Let σ_j be a random variable representing the time period between communications for the j^{th} target. At the instant of communication from target j, its exact position within $\mathcal{S}$ is known by the pursuer. Following this, the target may continue to move but the pursuer receives no updated information until the next communication from target j. We consider here a modified version of the Marco Polo problem. First, no communication between targets and pursuers takes place. Second, pursuers have a limited detection disc $\mathcal{D}_i$ so that targets are detected if they enter $\mathcal{D}_i$. At any time t, the set of detections is denoted by Z^t_j, and symbolizes all measurements of the target positions τ_j obtained since the initial time $t = 0$, e.g., $Z^t_j = \{z_1(t_1), z_2(t_1), \ldots, z_i(t_\ell), \ldots, z_j(t)\}$. Based on the previous discussion, the problem can be stated as follows

Problem 2.1: Given a set $\mathcal{P}$ of N pursuers and a set $\mathcal{T}$ of M target agents within a specified game area $\mathcal{S}$, find a set of policies $u^i_p = c^i(p_i, Z^t_\tau) \in \mathcal{U}$ for all pursuers in $\mathcal{P}$ which maximizes the probability of intercepting partially-observed and unobserved tracks and minimizes the time t_c required to capture fully-observed targets in $\mathcal{T}$.

To solve the above problem, we assume that the autonomous pursuers each perform their own detection processing to determine the time and location of a detection event. A *track-before-detect* approach [15] is considered, in which the individual sensor detections are used to form a hypothetical track before declaring a positive detection. This approach is suited to systems comprised of sensors that are relatively simple and produce few observations for each moving target, have no prior knowledge of its track, and are subject to frequent false alarms. Once a track has been formed from at least k individual detections that have occurred at different moments in time, an upper-level controller determines the purssuer in $\mathcal{P}$ that is in the best position to pursue it.

In this approach, it is convenient to classify tracks based on the following definitions. An *unobserved track* refers to the state (e.g., position, velocity, and acceleration) of an evader (2) for which there have been no detections up to the present time, t. A *partially-observed track* refers to the track of an evader that is formed from $0 < l < k$ individual sensor detections obtained up to time t. An evader track is said to be *fully-observed* when it is formed from $l > k$ individual sensor detections obtained up to time t. Only after a track is fully-observed, the target is considered to be *positively detected*. Then, the formed track is used to compute a pursuit strategy on line, minimizing the individual capture time t_{c_j}. Assumptions on the agents dynamics are that the maximum velocity $V_{p,\tau_{\max}}$ of all agents (pursuers and targets) is known, the translational speed of target j is uniformly distributed in $[0, V_{\tau_{\max}}]$, and $V_{p_{\max}} > V_{\tau_{\max}}$.

Pursuers always operate in one of two modes: detection or pursuit. The objectives of an agent in detection mode can be summarized as follows: (i) Avoid n fixed obstacles $\{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$; (ii) maximize the probability of intercepting m partially-observed tracks $\{\mathcal{R}^1_\theta, \ldots, \mathcal{R}^m_\theta\}$; and maximize the probability of intercepting unobserved tracks. On the other hand, the objective of an agent in pursuit mode is to minimize the time t_{c_j} required to capture the target τ_j, based on its track $\mathcal{R}^j_\theta$, the pursuer's initial position $p_i(t_0)$, and any information sensed during the pursuit, which may be used to update the track (e.g., as shown in [2]).

The following sections describe a methodology for planning the motions of the pursuers, in order to meet all of the above objectives.

III. Methodology

The geometry of the pursuers is assumed to be a convex polygon denoted by $\mathcal{A}_i$, with a configuration q_i with respect to a fixed Cartesian frame $\mathcal{F}_\mathcal{S}$ such that $\mathcal{A}_i(q_i) \in \mathcal{S}$. The robot workspace $\mathcal{S}$ is populated with n fixed obstacles $\mathcal{O}_j$, $j = 1, 2, \ldots, n$, to be avoided, and by m targets or partially-observed tracks $\mathcal{R}^j_\theta$, $i = 1, 2, \ldots, m$, to be intercepted. Let a disk $\mathcal{D}_i$ represent the field of view of an omnidirectional sensor installed on $\mathcal{A}_i$, with radius r_i, and such that $\mathcal{D}_i(q_i) \in \mathcal{S}$. Henceforth, the sensor installed on $\mathcal{A}_i$ can detect the target on the track $\mathcal{R}^j_\theta$ when $\mathcal{D}_i(q_i) \cap \mathcal{R}^j_\theta \neq \varnothing$. We are interested in applications where the sensors field of view is much larger than the robot geometry, and hence a robot can sense a target without necessarily being close enough to capture it.

A. Robot Motion Planning in the Presence of Multiple Targets and Obstacles

The partially-observed tracks are viewed as targets from which it is desirable to collect additional measurements, before investing in the costly resources needed to capture them, such as deploying one or more robots to chase one of many possible targets. Then, the subsets of $\mathcal{S}$ where the sensor can collect measurements from the partially-observed tracks can be defined as

Definition 3.1 (C-Target): The target track $\mathcal{R}^j_\theta$ in $\mathcal{S}$ maps in the i^{th} pursuer configuration space $\mathcal{C}$ to the C-target region $\mathcal{CR}_j = \{q_i \in \mathcal{C} \mid \mathcal{D}_i(q_i) \cap \mathcal{R}_j \neq \varnothing\}$.

The boundary of a C-target is the curve followed by the origin of $\mathcal{F}_{\mathcal{A}_i}$ when $\mathcal{D}_i$ slides in contact with the boundary of $\mathcal{R}^j_\theta$. With the assumed robot and sensor geometries, the C-targets boundaries are obtained by growing $\mathcal{R}^j_\theta$ isotropically by the radius r_i within $\mathcal{S}$. C-obstacles are defined as in [11] and used together with the C-targets to obtain a roadmap representation of the robot configuration space, as explained in the following paragraphs.

Well known path planning techniques, such as cell decomposition [11] or roadmap methods, are modified to account for the presence of the targets, which play a role opposite to obstacles. In classical cell decomposition, the obstacle-free configuration space, defined as,

$$\mathcal{C}^i_{free} = \mathcal{C} \setminus \bigcup_{k=1}^{n} \mathcal{CO}_k = \{q_i \in \mathcal{C} \mid \mathcal{A}_i(q_i) \cap (\bigcup_{k=1}^{n} \mathcal{O}_k) = \varnothing\} \tag{3}$$

is decomposed into a finite set of *cells*, $\{\kappa_1, \ldots, \kappa_f\}$, within which a path free of obstacles can be easily generated. To account for the presence of targets, let an *observation cell*, $\bar{\kappa}_l$, be defined as a subset in configuration space with the property that every robot configuration q_i in it enables sensor measurements from at least one track, i.e., $q_i \in \bar{\kappa}_l \subset \mathcal{CR}_j$ implies both $q_i \in \mathcal{C}^i_{free}$ (or, equivalently, $q_i \notin \mathcal{CO}_j$, for $\forall j$) and $q_i \in \mathcal{CR}_j$.

The following definition, taken from [11], is useful when the C-targets are grown isotropically by a disk, $\mathcal{D}_i$:

Definition 3.2 (Generalized Polygon): A generalized polygon is a subset of $\mathbb{R}^2$ that is homeomorphic to the closed unit disc and whose boundary is a closed-loop sequence of straight line segments and circular arcs.

Alternatively, the pill-shape C-targets can be approximated by a convex polygon. Then, given the initial and final configurations, q_0 and q_f, a connectivity graph including observation cells can be obtained by the following algorithm:

(I) Decompose the configuration space that is void of any C-obstacles or C-targets, and is defined as:

$$\begin{aligned}\mathcal{C}^i_{void} &= \mathcal{C} \setminus \{\bigcup_{k=1}^{n} \mathcal{CO}_k \cup \bigcup_{j=1}^{r} \mathcal{CR}_j\} \\ &= \{q_i \in \mathcal{C} \mid \mathcal{A}_i(q_i) \cap (\bigcup_{k=1}^{n} \mathcal{O}_k) = \varnothing, \\ &\qquad \mathcal{D}_i(q_i) \cap (\bigcup_{j=1}^{r} \mathcal{R}_j) = \varnothing\}\end{aligned} \tag{4}$$

(II) Decompose each obstacle-free C-target,

$$\mathcal{CR}_j \setminus \bigcup_{k=1}^{n} \mathcal{CO}_k, \tag{5}$$

thereby obtaining the set of observation cells.

(III) Construct a connectivity graph $\mathcal{G}$ using both void (I) and observation cells (II).

A sweeping-line algorithm can be used to decompose a non-convex generalized polygon with ν vertices into $O(\nu)$ convex generalized polygons in $O(\nu log \nu)$ time (see Section 5.1 in [11]). An illustrative example of workspace and corresponding cell decomposition is shown in Fig. 1.

B. Search Area Coverage

The observation cells in the connectivity graph represent robot configurations that enable additional individual sensor

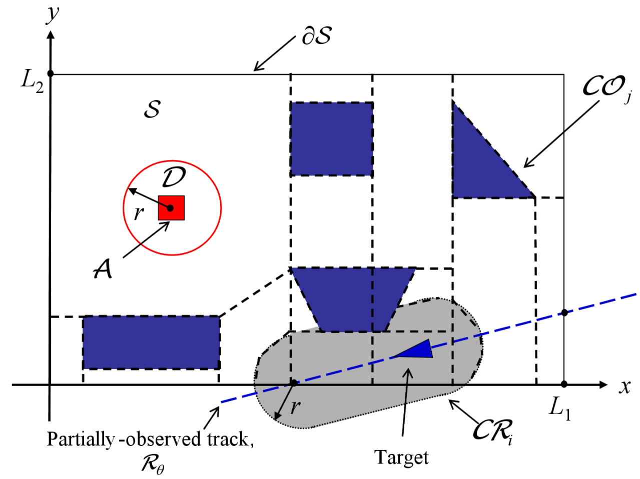

Fig. 1. Example of cell decomposition (dashed lines) for a workspace with four C-obstacles (black regions) and one C-target $\mathcal{CR}$ (patterned region) corresponding to $2 < k$ prior detections. The sensor with range r is installed on a robot with a square platform geometry $\mathcal{A}$.

detections of targets that may be moving along partially-observed tracks. Additionally, the team of pursuers must also detect new targets that may have just recently entered the search area $\mathcal{S}$ or that may have been missed up to the present time, t. Since the targets are always in motion, maximizing the area coverage of the sensors may not be the best strategy for obtaining *at least* k detections from each target. In fact these detections may take place at different moments in time, anywhere along a target track, and still lead to a positive detection provided they form a feasible track. The problem of intercepting a target moving along a straight line across $\mathcal{S}$ by means of k omnidirectional sensors in a network of size n is referred to as track-coverage [9]. It can be viewed within the context of geometric transversal theory, where:

Definition 3.3: A family of k convex sets in $\mathbb{R}^n$ is said to have a d*-transversal* if it is intersected by a common d-dimensional flat (or translate of a linear subspace).

When, $d = 1$ and the transversal is also known as a line-stabber of the family of convex sets. In the track coverage problem, $d = 1$, and a track intercepted by k sensors is said to be a stabber of the corresponding fields of view, e.g., $\{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$, in $\mathbb{R}^2$.

Then, it can be shown [9] that the family of stabbers of one disk $\mathcal{D}_i$, with radius r_i and centered at $p_i = [x_i,\ y_i]^T$ in $\mathbb{R}^2$, can be described by a two-dimensional cone defined by the unit vectors,

$$\hat{\mathrm{h}}_i = \begin{bmatrix} \cos\alpha_i & -\sin\alpha_i \\ \sin\alpha_i & \cos\alpha_i \end{bmatrix} \frac{\mathrm{v}_i}{\|\mathrm{v}_i\|} = \mathrm{Q}_i^+ \hat{\mathrm{v}}_i, \tag{6}$$

and,

$$\hat{\mathrm{l}}_i = \begin{bmatrix} \cos\alpha_i & \sin\alpha_i \\ -\sin\alpha_i & \cos\alpha_i \end{bmatrix} \frac{\mathrm{v}_i}{\|\mathrm{v}_i\|} = \mathrm{Q}_i^- \hat{\mathrm{v}}_i, \tag{7}$$

for the y-intercept b_y. Where, α_i denotes half of the opening angle of the coverage cone and its trigonometric functions are easily computed from the position p_i,

$$\sin\alpha_i = \frac{r_i}{\|\mathrm{v}_i\|} = \frac{r_i}{\sqrt{x_{1,i}^2 + (x_{2,1} - b)^2}} \tag{8}$$

and,

$$\cos\alpha_i = \frac{\sqrt{\|\mathrm{v}_i\| - (r_i)^2}}{\|\mathrm{v}_i\|} \tag{9}$$

with $\mathrm{v}_i \equiv p_i - [0 \quad b_y]^T$. Also, let all unit vectors of the same class be ordered according to their direction sine in $\mathbb{R}^2$. Then, for a fixed y-intercept b_y, the family of stabbers of a family of k disks with different radii and coordinates can be described by a cone, referred to as *k-coverage cone*, that is defined by two unit vectors:

$$\hat{l}^* \equiv \sup\{\hat{l}_1, \ldots, \hat{l}_k\}, \quad \hat{h}^* \equiv \inf\{\hat{l}_1, \ldots, \hat{l}_k\}$$

The opening angle of a k-coverage cone defined with respect to the y-axes,

$$\psi = \sin^{-1} ||\hat{l}^* \times \hat{h}^*|| \tag{10}$$

is a measure over the set of line stabbers for the family of k disks, and is used to derive the probability of detection for unobserved target tracks.

By summing the opening angles of the k-coverage cones corresponding to all possible intercepts, obtained by discretizing the border of $\mathcal{S}$, the following function is derived for the probability of detection of unobserved tracks:

$$\begin{aligned} P_{\mathcal{S}}^k(\mathcal{X}) \;\; = \;\; & \frac{\delta_2}{2\pi(L_2+\delta_2)} \sum_{b_y=0}^{L_2} [H(\psi)\psi + H(\varphi)\varphi] \quad (11) \\ & + \frac{\delta_1}{2\pi(L_1+\delta_1)} \sum_{b_x=0}^{L_1} [H(\zeta)\zeta + H(\rho)\rho] \end{aligned}$$

Where, the coverage function of a configuration of sensors $\mathcal{X} = \{p_1, \ldots, p_n\}$, given by,

$$\begin{aligned} 2 \cdot \mathcal{T}_{\mathcal{S}}^k(\mathcal{X}) \;\; = \;\; & \sum_{b_y=0}^{L_2} H(\psi)\psi + \sum_{b_x=0}^{L_1} H(\zeta)\zeta \quad (12) \\ & + \sum_{b_{y'}=0}^{L_2} H(\varphi)\varphi + \sum_{b_{x'}=0}^{L_1} H(\rho)\rho \end{aligned}$$

is derived in [9], and ζ, φ, and ρ denote the opening angles defined with respect to the axis x, y', and x', respectively. The Heaviside function $H(\cdot)$ guarantees that if $\hat{l}^* \succ \hat{h}^*$ the opening angle of the coverage cone is equal to zero. As an example, these angles are illustrated in Fig. 2, for a network with $n = 2$ sensors and $k = 2$ required detections. Also, it can be shown that total track-coverage is given by the constant,

$$\mathcal{T}_{\mathcal{S}}^{max} = \pi \left(\frac{\delta_2 + L_2}{\delta_2} + \frac{\delta_1 + L_1}{\delta_1} \right) \tag{13}$$

for any k and n, where δ_1 and δ_2 are the discretization intervals for (x, x') and (y, y'), respectively. Therefore, (11) can be used to measure the probability of detection of a sensor configuration $\mathcal{X}$, with n sensors, and k-required detections per track, given uniform probability distributions for their points of entry along $\partial\mathcal{S}$.

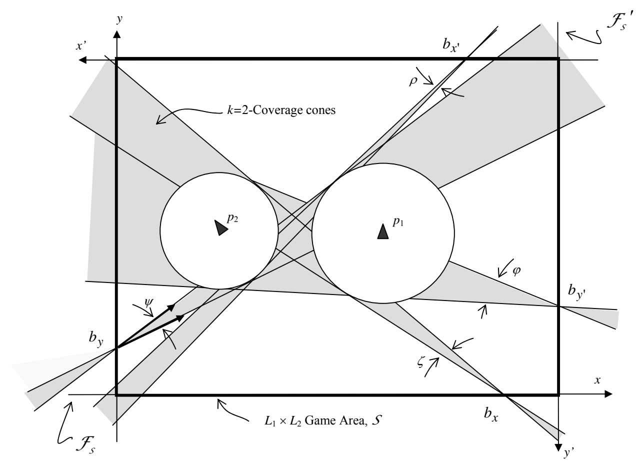

Fig. 2. Coverage cone definition illustrated for two sensors with fields of view centered at p_1 and p_2, and a rectangular area of interest $\mathcal{S}$, with perimeter $\partial\mathcal{S}$ shown in bold.

C. Information-driven Motion Planning

Information-driven motion planning is a new problem that arises when the robots navigate a workspace for the purpose of obtaining sensor information from multiple targets. In classical robot motion planning, the goal is to reach a final configuration q_f by traveling an obstacle-free path of minimum cost. By utilizing the free-space decomposition approach presented in Section III-A, the presence of the targets is accounted for, and the robot configurations that enable sensor measurements (i.e., lead to non-empty intersections of targets and sensor field of view) are represented by observation cells in the connectivity graph. While not sufficient to guarantee observation, visiting these cells increases the probability of observing the corresponding target.

The sensor planning objectives are expressed in terms of a reward function that is assigned to every arc in the connectivity graph

$$R(\kappa_l, \kappa_\imath) = B(\kappa_l, \kappa_\imath) - J(\kappa_l, \kappa_\imath) \tag{14}$$

representing the expected profit of the information that will be obtained by moving from a configuration $q_i \in \kappa_l$ to a configuration $q_i \in \kappa_\imath$, that is, the associated benefit minus its cost. Then, the optimal pursuer's path is the sequence of adjacent cells or *channel* that maximizes the total reward

$$\mu^* \equiv \{\kappa_0, \ldots, \kappa_f\}^* = \arg\max_{\mu} \sum_{(\kappa_l, \kappa_\imath) \in \mu} R(\kappa_l, \kappa_\imath), \tag{15}$$

and connects κ_0 to κ_f in the connectivity graph $\mathcal{G}$. Suppose κ_l is an observation cell that is obtained from the decomposition of the j^{th} C-target: $\bar{\kappa}_l \subset \mathcal{CR}_j$. Then, one benefit of visiting the l^{th} cell in $\mathcal{G}$ is the probability of detecting the target τ_j

$$P_{\mathcal{R}}(\bar{\kappa}_l \subset \mathcal{CR}_i) = \Pr\{D_{ji} = 1 \mid e_{ji} \leq r_i\}, \tag{16}$$

where D_{ji} represents the event that the i^{th} sensor reports a detection when the j^{th} target comes within its detection range. In this paper, it is assumed to be equal to one for simplicity, and when a cell is void $P_{\mathcal{R}} = 0$. In general, it can be estimated from knowledge of the measurement process, and can be made dependent on location and distance from the target.

Another benefit of moving to a configuration within a cell κ_l is the gain in probability of detection of unobserved tracks $\Delta\mathcal{P}_{\mathcal{S}}^k$ that when negative translates into a cost. The cost of traveling to any cell in $\mathcal{G}$ is comprised of distance and, possibly, of energy and computational power required by sensor measurements, if the cell is an observation cell. Since these quantities may vary slightly within each cell, they are computed in reference to the geometric centroid, $\bar{q}_i$, such that the distance-cost associated with moving between two nodes $\kappa_l \rightarrow \kappa_\imath$ in $\mathcal{G}$ is the Euclidean distance,

$$d(\kappa_l, \kappa_\imath) \equiv \max ||\mathcal{A}(\bar{q}_\imath) - \mathcal{A}(\bar{q}_l)||, \tag{17}$$

and the gain in probability of detection is

$$\Delta\mathcal{P}_{\mathcal{S}}^k(\kappa_l, \kappa_\imath) \equiv \mathcal{P}_{\mathcal{S}}^k(\mathcal{X}_\imath) - \mathcal{P}_{\mathcal{S}}^k(\mathcal{X}_l), \tag{18}$$

where $\mathcal{X}_l$ represents the set of pursuers' locations when the i^{th} pursuer is positioned such that the center of $\mathcal{D}_i$ coincides with the centroid of κ_l, i.e., $p_i \subset \bar{q}_l$. So, in summary, the reward function can be written as

$$R(\kappa_l, \kappa_\imath) = w_1 P_{\mathcal{R}}(\kappa_l) + w_2 \Delta\mathcal{P}_{\mathcal{S}}^k(\kappa_l, \kappa_\imath) - w_3 d(\kappa_l, \kappa_\imath). \tag{19}$$

In this paper, sensor power and energy considerations are neglected for simplicity. And, the weights w_1, w_2, and w_3 are chosen by the designer based on the desired tradeoff between these competing objectives.

IV. Control Strategy

Once the connectivity graph has been obtained, and the rewards computed for each arc in $\mathcal{G}$, the optimal channel μ^* is obtained using the graph searching algorithm A* [11]. Next, the output of the motion planner is mapped into a set of waypoints which in turn are used by a trajectory generator and trajectory tracking controller. In this section, we focus on a simple yet effective strategy to capture a full-observed target. Several control algorithms can be implemented such as a proportional controller and a more sophisticated leader-follower approach. In the latter case the target to capture is seen as the leader to follow.

We describe an effective approach that allows a pursuer to capture a target along a trajectory of minimal length [8]. We assume that the target is moving in a straight line with constant velocity and intercepting along that line. This strategy depicted in Fig. 3 is based upon the geometry of the problem and taking into account the kinematic constraints of the pursuer. The strategy attempts to intercept the target at a point δ. The interception point is calculated by determining the time for both the pursuer and target to reach that point. The initial states of the pursuer and target are $p_0 = (x_p, y_p, \theta_p)$ and $\tau_0 = (x_\tau, y_\tau, \theta_\tau)$, respectively. The interception point is defined as

$$\delta = \begin{bmatrix} x_t + t_c v_t \cos\theta_t \\ y_t + t_c v_t \sin\theta_t \end{bmatrix}$$

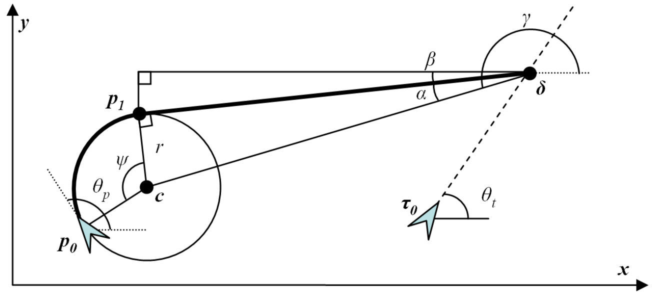

Fig. 3. Control strategy to capture a target.

and the time to interception becomes

$$t_c = \frac{r\psi + \|c - \delta\| \cos\alpha}{v_p}, \tag{20}$$

where the distance traveled by the pursuer is the distance along the arc $p_0 p_1$ plus the straight line distance between p_1 and δ. The arc radius is the turn radius of the pursuer and is defined as $r = \frac{v_p}{\omega_p}$. Using the geometric definition, the interception point δ in (20) which is passed to the controller may be solved numerically using an iterative algorithm or a Newton method.

V. Simulation Results

This simulation scenario considers a rectangular environment with multiple targets and multiple pursuit sensors using reward function weights $w_1 = 1$, $w_2 = 1$, and $w_3 = 1$. Before the simulation scenario begins, five sensors – one with sensing radius 1.5 m, one with sensing radius 1.25 m, and three with sensing radii of 1 m – with platforms measuring 0.25 m square are placed in the 10 m by 10 m environment to maximize the probability of detecting tracks with $k = 2$. Obstacle and coverage maps are generated for each sensor corresponding to placement in each cell. Initially, all sensors are in *detection* mode and each is a candidate to switch to the *pursuit* mode when target tracks become fully observed.

In this scenario, two targets enter the environment at different locations and headings and with different velocities. As they move along their trajectories, they are detected by the sensors . The sensors remain motionless since each target has been detected only once. After the second detection of a target, the network hypothesizes the target track based on previous detections and deploys the sensor which receives the highest reward (or lowest cost) as obtained by the A* graph searching algorithm to move to obtain an additional detection of the target (Fig. 4). When the second target becomes partially detected, the same track hypothesis and sensor deployment occurs. At the point that the first target's track becomes fully observed, the network again evaluates the reward (distance) and deploys the best sensor, denoted by its green color in Fig. 5, to pursue the target. The same pursuit is performed when the second target is fully observed.

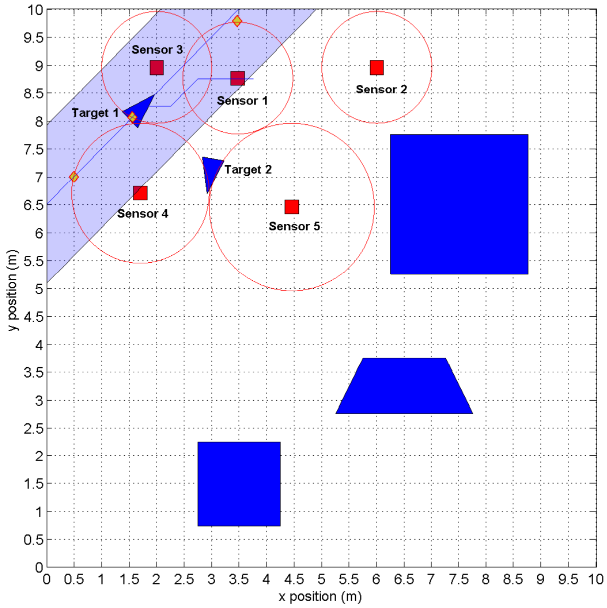

Fig. 4. Target 1 is partially observed with its hypothesized track, and Sensor 1 is deployed to obtain additional observations.

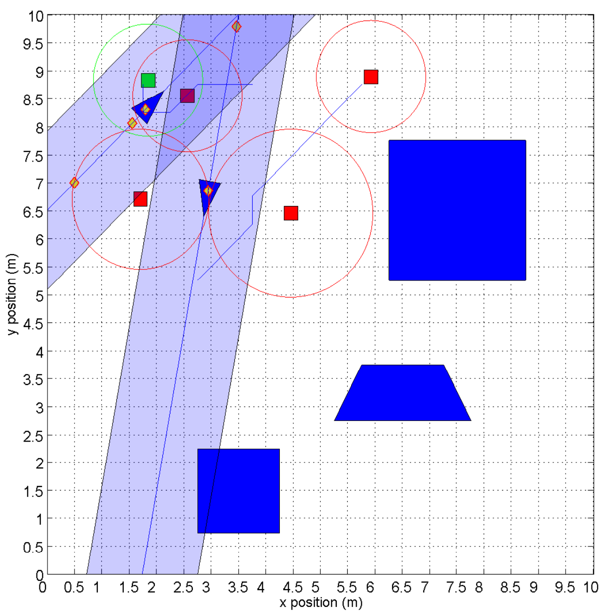

Fig. 5. Target 1 is fully observed and Sensor 3 is deployed to pursue it while Target 2 becomes partially observed, and Sensor 2 is deployed to obtain an additional observation.

VI. CONCLUSIONS

This paper presents a new framework for developing sensor control policies in systems involving multiple robotic platforms that seek to detect and intercept multiple mobile targets. Multiple objectives, including the probability of detection for unobserved tracks, for which little or no information is available *a priori*, obstacle-avoidance, and the profit of information associated with partially-observed targets are approached using a geometric approach. The A^* graph-searching algorithm obtains the optimal tradeoff path for the control strategy that accounts for the actual pursuers' dynamics. By adopting a track-before-detect approach, a target is declared positively detected once a satisfactory number of detections k are measured. Subsequently, a heuristic rule switches one of the mobile sensors from detection mode to pursuit mode, and the track is readily available to compute an optimal pursuit strategy. By maximizing the same reward function, the remaining sensors in detection mode are reconfigured such that the probability of detecting the remaining targets is again optimized.

ACKNOWLEDGMENTS

The work of S. Ferrari is supported in part by the Office of Naval Research Young Investigator Program (Code 321). The work of R. Fierro is supported by NSF grant CAREER #0348637 and by the U.S. Army Research Office under grant DAAD19-03-1-0142 (through the University of Oklahoma).

REFERENCES

[1] A. Arsie and E. Frazzoli, "Efficient routing of multiple vehicles with no communications," *Int. J. Robust and Nonlinear Control*, 2006, submitted.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: Algorithms and software for information extraction*. John Wiley and Sons, 2001.

[3] C. Belta and V. Kumar, "Optimal motion generation for groups of robots: a geometric approach," *ASME Journal of Mechanical Design*, vol. 126, pp. 63–70, 2004.

[4] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, March 2004.

[5] C. Cai and S. Ferrari, "On the development of an intelligent computer player for CLUE: A case study on preposterior decision analysis," in *Proceedings of the American Control Conference*, Minneapolis, MN, June 2006, pp. 4350–4355.

[6] T. H. Chung, J. W. Burdick, and R. M. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *IEEE Int. Conf. on Robotics and Automation*, Orlando, Florida, May 2006, pp. 2416–2422.

[7] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, April 2004.

[8] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[9] S. Ferrari, "Track coverage in sensor networks," in *Proceedings of the American Control Conference*, Minneapolis, MN, June 14-16 2006, pp. 2053–2059.

[10] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, April 2000, pp. 95–101.

[11] J. C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[12] T. G. McGee and J. K. Hedrick, "Guaranteed strategies to search for mobile evaders in the plane," in *Proceedings of the American Control Conference*, Minneapolis, MN, June 14-16 2006, pp. 2819–2824.

[13] B. Perteet, J. McClintock, and R. Fierro, "A multi-vehicle framework for the development of robotic games: The Marco Polo case," in *IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, April 2007, (To Appear).

[14] R. Vidal, O. Shakernia, J. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, October 2002.

[15] T. A. Wettergren, R. L. Streit, and J. R. Short, "Tracking with distributed sets of proximity sensors using geometric invariants." *IEEE Trans. Aerospace and Electronic Systems*, vol. 40, no. 4, pp. 1366–1374, 2004.