A Geometric Optimization Approach to Tracking Maneuvering Targets Using a Heterogeneous Mobile Sensor Network

Silvia Ferrari*, Rafael Fierro† and Domagoj Tolic†
*Laboratory for Intelligent Systems and Control (LISC), Department of Mechanical Engineering, Duke University
Durham, NC 27708-0005, sferrari@duke.edu
†MARHES Lab, Department of Electrical and Computer Engineering, University of New Mexico
Albuquerque, NM 87131–0001, {rfierro, dtolic}@ece.unm.edu

***Abstract*—A methodology is developed to deploy a mobile sensor network for the purpose of detecting and capturing mobile targets in the plane. The mobile sensor network consists of a set of heterogeneous robotic sensors modeled as hybrid systems with individual processing capabilities. The targets are modeled by a Markov motion process that is commonly used in target tracking applications. Since the sensors are installed on mobile robots and have limited range, the geometry of their platforms and fields-of-view play a critical role in motion planning and obstacle avoidance. The methodology presented in this paper uses line transversals and cell decomposition in order to compute sensing and pursuit strategies that maximize the probability of detection, while minimizing energy consumption. The approach is demonstrated through progressive simulation scenarios involving multiple sensors installed on UGVs and UAVs, that are characterized by different sensing and motion capabilities, but are deployed to cooperatively detect, track, and pursue the same set of maneuvering targets.**

I. INTRODUCTION

Modern sensing and communication technologies are producing advanced surveillance systems in which both the sensors and their platforms are characterized by a high degree of functionality and reconfigurability. Mobile sensor networks employed for monitoring urban environments, or unauthorized intruders are expected to operate cooperatively and reliably in cluttered dynamic environments with little human intervention. Coordinating such large heterogeneous sensor networks is challenging, and requires the development of novel methods of communication, motion control and planning, computation, proactive estimation and sensing, and power management [1]. As a result, in recent years, a variety methodologies for coordination of robotic networks and sensor planning have been proposed based on distributed control and stochastic hybrid systems [2]–[4]. A comprehensive review of distributed control of synchronous robotic networks with an emphasis on communication protocols and geometric notions relevant in motion coordination is provided in [5]. One line of research has investigated the extension of motion planning techniques to the problem of sensor placement for achieving coverage of unstructured environments [6], [7], or of a desired visibility space [8], [9]. Obstacle-avoidance motion planners have been effectively modified in [10], [11] to plan the path of mobile sensors for the detection and classification of stationary targets in an obstacle-populated environment. Probabilistic pursuit-evasion strategies to detect and capture intelligent evaders in obstacle-populated environments are described in [12].

The sensor planning problem treated in this paper is motivated by heterogeneous mobile sensor networks that are deployed to cooperatively detect, track, and pursue multiple unauthorized targets. This problem is analogous to the Marco Polo game in which a pursuer *Marco* must capture multiple mobile targets that are sensed intermittently, and with very limited information in [13]. When the pursuers are equipped with wireless sensors, this problem combines pursuit-evasion games' objectives, with sensing objectives that represent the network's ability to perform cooperative multi-target tracking and distributed estimation, in order to determine the target tracks. A geometric optimization approach based on cell decomposition and geometric transversals is used to obtain a graph representation of the robot configuration space that is void of obstacle and enables target detections. Cell decomposition algorithms have previously been employed to represent the obstacle-free configuration of a robot for the purpose of obstacle avoidance [14]. In this paper, the algorithm developed in [11] is used to obtain a decomposition in which observation cells are used to represent configurations intersecting the targets. By this approach, the sensor's platform avoids intersections with the obstacles, while the on-board sensor's field-of-view intersects the targets to enable detections. The decomposition takes into account the sensors' geometric characteristics, as well as the geometry of the workspace and that of the tracks formed from prior detections.

When track estimates are available, they are used in the decomposition and determine the probability of obtaining additional detections in the observation cells. At any given time, the pursuers must also detect new targets, based on little or no prior information, by maximizing their track coverage [15]. Track coverage refers to the probability that multiple sensors will obtain multiple detections, possibly at different moments in time, along the target track [15]. In this paper, we extend previous work on track coverage [15] and C-targets [16] to the case of maneuvering targets that can change their speed and heading at random moments in time and, thus, can be modeled by a Markov motion process [17]. Another

extension considered in this paper is that of a heterogeneous sensor network, such as one comprised of both aerial and ground sensors that are used to detect and capture mobile targets in the plane. Ground sensors have smaller fields-of-view and are slower than aerial sensors, but can capture the targets. Aerial sensors can fly over obstacles, but can only be used in detection mode. The approach presented in this paper computes a set of control policies that optimize a tradeoff between sensing and motion objectives by searching the robot connectivity graph, and by performing inner-loop trajectory generation and tracking. As a result, the sensors' paths maximize the overall probability of detection, and minimize the distance traveled to avoid obstacles, and to detect or capture the targets.

The remainder of the paper is organized as follows. The problem formulation and assumptions are stated in Section II. The modeling and solution method used to compute the control policies for the sensor network are presented in Section III. Simulations and numerical results are provided in Section IV.

II. Problem Formulation and Assumptions

This paper considers a network of N cooperative heterogeneous sensors installed on mobile platforms that are deployed in a square area-of-interest (AOI) for the purpose of detecting, tracking, and pursuing M moving targets with minimum energy, during a time interval $[t_0,\ t_f]$. Where, the end time t_f is not fixed, and is defined as the time at which all targets have been captured. The AOI, $\mathcal{S} = [0,\ L] \times [0,\ L] \subset \mathbb{R}^2$, has a boundary $\partial\mathcal{S}$ and is populated by n fixed and convex obstacles $\{\mathcal{O}_1, \ldots, \mathcal{O}_n\} \subset \mathcal{S}$. The geometry of the i^{th} pursuer is assumed to be a convex polygon denoted by $\mathcal{A}_i$, with a configuration q_i that specifies its position and orientation with respect to a fixed Cartesian frame $\mathcal{F}_\mathcal{S}$. A subset of the sensors, e.g., unmanned ground vehicles (UGVs), must avoid the obstacles in $\mathcal{S}$, while the rest of the sensors, e.g., unmanned aerial vehicles (UAVs), can fly over obstacles to minimize the distance traveled.

The dynamics of the sensors are approximated by the nonholonomic unicycle model,

$$\begin{aligned} \dot{x}^i_\mathcal{P} &= v^i_\mathcal{P} \cos\theta^i_\mathcal{P}, \\ \dot{y}^i_\mathcal{P} &= v^i_\mathcal{P} \sin\theta^i_\mathcal{P}, \\ \dot{\theta}^i_\mathcal{P} &= \omega^i_\mathcal{P}, \end{aligned} \tag{1}$$

where $q^i_\mathcal{P} = (x^i_\mathcal{P},\ y^i_\mathcal{P},\ \theta^i_\mathcal{P}) \in SE(2)$, and $p^i_\mathcal{P} = [x^i_\mathcal{P} \quad y^i_\mathcal{P}]^T \in \mathbb{R}^2$ is the position vector of pursuer i (referred to its centroid). The input to pursuer i is $u^i_\mathcal{P} = [v^i_\mathcal{P} \quad \omega^i_\mathcal{P}]^T$, and $u_\mathcal{P} \in \mathcal{U} \subset \mathbb{R}^2$. The set of all pursuers is denoted by $\mathcal{P}$, and $I_\mathcal{P}$ is the index set of $\mathcal{P}$. Each pursuer is equipped with an isotropic or omnidirectional sensor with a field-of-view (FOV) represented by a disk $\mathcal{D}_i(t) = \mathcal{D}[p_i(t), r_i] \in \mathcal{S}$ that has a constant radius r_i, and is centered at p_i at time t.

The set of all targets in $\mathcal{S}$ is denoted by $\mathcal{T}$, where $I_\mathcal{T}$ is the index set of $\mathcal{T}$. This paper extends the approach presented in [18], which assumed targets move along straight lines, to the case of targets that exhibit a piece-wise linear motion and can be modeled by a Markov motion process [17]. In this model, the motion of the target is modeled as a continuous-time Markov process. From [19] we have the following definitions:

Definition 1: A continuous-time random process is a family of random variables x_t where t ranges over a specified interval of time.

Definition 2: We say that x_t is a continuous-time Markov process if for $0 \le t_0 < \cdots < t_{k-1} < t_k < t$ we have $\Pr\{x_t \in B | x_k = s_k, x_{k-1} = s_{k-1}, \cdots, x_0 = s_0\} = \Pr\{x_t \in B | x_k = s_k\}$ *where* Pr *denotes the probability function, and $s_1, \ldots, s_k \in \mathcal{X}$ are realizations of the state space $\mathcal{X}$.*

Let the random variables $\theta^j_\mathcal{T}$ and $v^j_\mathcal{T}$ represent the target's heading and velocity, respectively. Assuming the target's heading and velocity are constant during every interval $\Delta t_\jmath = (t_{\jmath+1} - t_\jmath)$, $\jmath = 1, 2, \ldots$, where $\Delta t_\jmath$ is not necessarily constant, the target motion can be modeled as a continuous-time Markov process with a family of random variables $\{q^j_\mathcal{T}, v^j_\mathcal{T}\}$, where $q^j_\mathcal{T} = (x^j_\mathcal{T},\ y^j_\mathcal{T},\ \theta^j_\mathcal{T}) \in SE(2)$, and $p^j_\mathcal{T} = [x^j_\mathcal{T} \quad y^j_\mathcal{T}]^T \in \mathbb{R}^2$.

Under the aforementioned assumptions, a three-dimensional real-valued vector function maps the family of random variables $\{\theta^j_\mathcal{T}(t), v^j_\mathcal{T}(t)\}$ into the random vector $p^j_\mathcal{T}(t)$ at every time $t \in [t_0,\ t_f]$, such that the value of the target motion process is given by

$$\begin{aligned} \dot{x}^j_\mathcal{T}(t) &= v^j_\mathcal{T}(t) \cos\theta^j_\mathcal{T}(t), \\ \dot{y}^j_\mathcal{T}(t) &= v^j_\mathcal{T}(t) \sin\theta^j_\mathcal{T}(t), \end{aligned} \tag{2}$$

and, therefore, the motion of target j is a Markov process. The third component of the vector function is the identity function. Thus, $\theta^j_\mathcal{T}$ and $v^j_\mathcal{T}$ are piece-wise constant, while $x^j_\mathcal{T}$ and $y^j_\mathcal{T}$ have discontinuities at the time instants $t_\jmath$, $\jmath = 1, \ldots, \rho^j$, when target j changes its heading and velocity. Let $\{p^j_{\mathcal{T}_\jmath}\}_{\jmath=1,\ldots,\rho^j}$ denote the set of target positions at which these discontinuities occur. Then, by integrating the linear differential equation (2) over every interval $\Delta t_\jmath$ with initial condition $p^j_{\mathcal{T}_\jmath}$, the position of the j^{th} target at any time t can be obtained as a function of the sequence of random variables $\{p^j_{\mathcal{T}_\jmath}, v^j_{\mathcal{T}_\jmath}, \theta^j_{\mathcal{T}_\jmath}\}_{\jmath=1,\ldots,\rho^j} \equiv \{\mathcal{M}_\jmath\}_{\jmath=1,\ldots,\rho^j}$ also known as Markov motion parameters,

$$p^j_\mathcal{T}(t) = p^j_{\mathcal{T}_\jmath} + v^j_{\mathcal{T}_\jmath}(t - t_\jmath)[\cos\theta^j_{\mathcal{T}_\jmath} \ \sin\theta^j_{\mathcal{T}_\jmath}]^T,\ t_\jmath \le t < t_{\jmath+1} \tag{3}$$

where, the Markov motion parameter values $\mathcal{M}_\jmath$ only depend on the values of the previous time step $\mathcal{M}_{\jmath-1}$, and remain constant during the time interval $\Delta t_\jmath$. It follows that the target motion is properly represented by the joint probability density function on $\mathcal{M}_\jmath$, which for simplicity can be assumed to be independent of the time interval $\Delta t_\jmath$.

It is assumed that the sensors operate in one of two modes, *detection* or *pursuit*, depending on whether their primary objective is to detect targets or to capture them. The i^{th} sensor has a non-zero probability to *detect* the j^{th} target at time t if and only if its FOV intersects the target's center of mass, i.e., $\mathcal{D}_i(t) \cap p^j_\mathcal{T}(t) \neq \varnothing$. The measurements obtained from each detection provide an estimate of the target position and are subject to errors and false alarms. Let the set of detections associated with the target j from t_0 up to time t be denoted by $Z^t_j = \{z_j(t_1), z_j(t_2), \ldots, z_j(t_l)\}$, and $Z^t = \{Z^t_j \mid j \in I_\mathcal{T}\}$.

A tracking algorithm is used to fuse these measurements and provide an estimate of the target motion parameters from Z_j^t, which is updated in real time via Kalman filtering. Once a target track has been estimated from at least k sensor detections, an upper-level controller selects and deploys a sensor in pursuit mode to capture it. The parameter k is chosen by the user based on the reliability of the sensors detections and on the cost associated with deploying a pursuer to capture the target [20]. Let e_{ji} be the Euclidean distance from the j^{th} target position to the closest sensor i. Then sensor i is said to *capture* target j when $e_{ji} < \varepsilon$, where ε is the *capture threshold* for an interval Δ_c called the *capture timeframe*. In order to ensure that the game can terminate, it is assumed that every sensor can move with any velocity in $[0, v_{\mathcal{P}_{\max}}]$, and that the targets can move with a maximum velocity that is known, and obeys $v_{\mathcal{T}_{max}} < v_{\mathcal{P}_{\max}}$. Once a target is captured, it becomes inactive and is removed from the set $\mathcal{T}$, thus the game terminates when $\mathcal{T} = \emptyset$. Since $r_i >> \varepsilon$, a sensor may obtain measurements from a target without necessarily being close enough to capture it.

As shown in Section III, the pursuit strategy of the sensor i selected to pursue and intercept target j takes into account the estimated target track and the sensor's dynamics (1) in order to minimize the *capture time* $t_{c_i^j}$, which is defined as the time that elapses between when sensor i becomes a pursuer, and when it captures target j. The objectives of the sensors in detection mode are to maximize the probability of cooperatively detecting unobserved tracks, and that of detecting partially-observed tracks, where target tracks are classified as follows:

Definition 3: An unobserved track is the path of a target j for which there are no detections at time, t, thus $Z_j^t = \emptyset$.

Definition 4: A partially-observed track is the path of a target that is estimated from $1 < l < k$ individual sensor detections obtained up to time t, i.e., $Z_j^t = \{z_j(t_1), \ldots, z_j(t_l)\}$.

Definition 5: A fully-observed track is the path of a target that is estimated from $m > k$ individual sensor detections obtained up to time t, i.e., $Z_j^t = \{z_j(t_1), \ldots, z_j(t_m)\}$, where $t_m \geq t_k$.

Sensors in any mode must also avoid obstacles, and collisions with other sensors. Thus, the sensing-pursuit problem can be stated as follows:

Problem 2.1: Given a set $\mathcal{P}$ of N mobile sensors with the dynamics in (1) and FOVs $\{\mathcal{D}_1(t), \ldots, \mathcal{D}_N(t)\}$ that are deployed to detect, track, and capture a set $\mathcal{T}$ of M targets with the dynamics in (3), find a set of policies $u_{\mathcal{P}}^i = c^i[q_{\mathcal{P}}^i(t), Z^t] \in \mathcal{U}$, for all $i \in I_{\mathcal{P}}$, which maximizes the total sensing reward, and minimizes the total capture time for $\mathcal{T}$, while avoiding obstacles in $\mathcal{S}$.

The total sensing reward and the set of policies of the sensor network are derived in the next section, and demonstrated through numerical simulations in Section IV.

III. Methodology

The methodology described in this section computes policies for a network of sensors that must meet multiple sensing and pursuit objectives in a dynamic, obstacle-populated environment. A game round is initiated every time a target track estimate is updated from the latest measurements, and a sensor is deployed either in detection or pursuit mode. The sensors' motion is planned using a connectivity graph obtained via cell decomposition [14]. Let $\mathcal{C}_{free}$ denote the robotic sensors' configuration space that is free of obstacles. A cell is defined as a closed and bounded subset of $\mathcal{C}_{free}$ within which a robotic sensor path can be easily generated, and is classified based on the following properties:

Definition 6: A void cell is a convex polygon $\kappa \subset \mathcal{C}_{free}$ with the property that for every configuration $q_i \in \kappa$ the sensor i has zero probability of detecting a partially-observed target.

In order to account for the sensing abilities of the robotic sensor, we also adopt the following definition from [11]:

Definition 7: An observation cell is a convex polygon $\underline{\kappa} \subset \mathcal{C}_{free}$ with the property that for every configuration $q_i \in \kappa$ the sensor i has a non-zero probability of detecting a partially-observed target.

Void and observation cells are determined based on the geometry of C-obstacles and C-targets using the approximate-and-decompose methodology presented in [11]. All cells are computed such that they do not overlap, and such that an obstacle-free pursuer path can be easily computed between any two configurations inside the same cell. By this methodology, adjacency relationships between cells are also established by determining whether any two cells in the decomposition share a common boundary and, therefore, whether the sensor can move between them without colliding with the obstacles. Thus, the approximate-and-decompose methodology can be used to obtain a one-dimensional representation of the workspace [11]:

Definition 8: A connectivity graph, $\mathcal{G}$, is an undirected graph where a node represents either an observation cell or a void cell, and two nodes in $\mathcal{G}$ are connected by an arc if and only if the corresponding cells are adjacent.

In this paper, the value of moving from a configuration in a cell κ_l to a configuration in κ_i in $\mathcal{G}$ is given by a sensing reward function that represents the difference between the average sensing gain and the average distance traveled, i.e.,

$$R(\kappa_l, \kappa_i) = w_{\mathcal{S}} \Delta P_{\mathcal{S}}(\kappa_l, \kappa_i) + w_{\mathcal{R}} P_{\mathcal{R}}(\kappa_i) - w_d d(\kappa_l, \kappa_i), \quad (4)$$

where, $\Delta P_{\mathcal{S}}$ denotes the gain in the probability of detecting unobserved tracks, $P_{\mathcal{R}}$ denotes the probability of detecting partially-observed tracks, and the Euclidean distance d between cells is defined as

$$d(\kappa_l, \kappa_i) \equiv \max ||\mathcal{A}(\bar{q}_i) - \mathcal{A}(\bar{q}_l)||. \quad (5)$$

The connectivity graph and the probability density functions $\Delta P_{\mathcal{S}}$ and $P_{\mathcal{R}}$, derived in the next subsections, are updated at every t, based on Z^t. From (4), the value of $R(\kappa_l, \kappa_i)$ is computed for and attached to every arc $(\kappa_l, \kappa_i) \in \mathcal{G}$, and used to compute the optimal policies, as shown in Section III-C.

A. Probability of Detection of Unobserved Tracks

The probability of obtaining multiple k detections by a cooperative, omnidirectional sensor network, referred to as

track coverage, was recently obtained for unauthorized targets that are assumed to travel along straight tracks in [15]. In this paper, a novel probability function representing the track coverage for unauthorized targets with a Markov motion process (Section II) is derived using convex theory and geometric transversals (see [21] for a comprehensive review):

Definition 9: A family of k convex sets in $\mathbb{R}^c$ 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 $c = 2$, the transversal is said to be a *line-stabber* of the family of convex sets. For convenience, we refer to the target track obtained from a Markov motion process as *Markov track*. As explained in Section II, the j^{th} Markov track consists of a set of ρ^j straight-line segments randomly generated according to the probability distributions of the parameters $\{\mathcal{M}_j\}_{j=1,\ldots,\rho^j}$. When a Markov track is detected by k sensors, one or more of these segments are the stabbers of $\{\mathcal{D}_1(t),\ldots,\mathcal{D}_N(t)\}$ in $\mathbb{R}^2$, possibly at different moments in time during the interval $[t_0,\ t_f]$. The temporal nature of the detections can be treated by noting that a spatio-temporal Markov track also consists of ρ^j three-dimensional straight-line segments in the space $\mathcal{S}\times[t_0,\ t_f]\subset\mathbb{R}^3$ (where, we can let $t_0 = 0$ for simplicity). By definition, sensor i has a non-zero probability to detect target j at time t if and only if $||p^i_{\mathcal{P}}(t) - p^j_{\mathcal{T}}(t)|| \leq r_i$. Thus, the family of segments that are stabbers of the disk $\mathcal{D}_i(t)$ in $\mathbb{R}^3_+$, at t, can be represented by a three-dimensional generalized cone parameterized by,

$$\begin{bmatrix} x \\ y \\ t \end{bmatrix} = p^j_{\mathcal{T}_j} + t_k \begin{bmatrix} r_i\cos\theta^j_{\mathcal{T}_j} \\ r_i\sin\theta^j_{\mathcal{T}_j} \\ 1 \end{bmatrix} \tag{6}$$

and with a random vertex $p^j_{\mathcal{T}_j}$. The cone in (6), denoted by $K[\mathcal{D}_i(t), p^j_{\mathcal{T}_j}]$, contains all combinations of target headings and velocities that would cause a detection at t and, thus, it is referred to as *coverage cone*.

An efficient representation for $K[\mathcal{D}_i(t), p^j_{\mathcal{T}_j}]$ consists of its two-dimensional projection onto $\mathcal{S}$, and of its intersection with the so-called velocity plane. The projection of $K[\mathcal{D}_i(t), p^j_{\mathcal{T}_j}]$ onto $\mathcal{S}$ is a two-dimensional cone $K_\theta[\mathcal{D}_i(t), p^j_{\mathcal{T}_j}]$ with an opening angle $\psi_i = 2\alpha_i$, where,

$$\alpha_i(t) = \sin^{-1}\left[\frac{r_i}{\|p^i_{\mathcal{P}}(t) - p^j_{\mathcal{T}_j}\|}\right]. \tag{7}$$

$K_\theta[\mathcal{D}_i(t), p^j_{\mathcal{T}_j}]$, abbreviated by K_θ for simplicity, is generated by the unit vectors,

$$\hat{h}_i(t) = \begin{bmatrix} \cos\alpha_i(t) & -\sin\alpha_i(t) \\ \sin\alpha_i(t) & \cos\alpha_i(t) \end{bmatrix} \frac{p_i(t)}{\|p_i(t)\|} \tag{8}$$

and,

$$\hat{l}_i(t) = \begin{bmatrix} \cos\alpha_i(t) & \sin\alpha_i(t) \\ -\sin\alpha_i(t) & \cos\alpha_i(t) \end{bmatrix} \frac{p_i(t)}{\|p_i(t)\|}, \tag{9}$$

where $p_i(t) \equiv p^i_{\mathcal{P}}(t) - p^j_{\mathcal{T}_j}$. K_θ is referred to as *heading cone* because it contains all target heading angles that would cause a detection at t by the sensor i. The velocity plane is defined as,

$$(\sin\theta^j_{\mathcal{T}_j})x + (\cos\theta^j_{\mathcal{T}_j})y = 0 \tag{10}$$

and, thus, it contains the t-axis and is perpendicular to $\mathcal{S}$. The intersection of $K[\mathcal{D}_i(t), p^j_{\mathcal{T}_j}]$ with the velocity plane (10) is a two-dimensional cone denoted by $K_v[\mathcal{D}_i(t), p^j_{\mathcal{T}_j}, \theta^j_{\mathcal{T}_j}]$, and abbreviated by K_v, for all $\theta^j_{\mathcal{T}_j} \in K_\theta$. Where, $\theta^j_{\mathcal{T}_j} \in K_\theta$ is a shorthand notation for target headings that satisfy the inequality

$$[\tan^{-1}(y^i_{\mathcal{P}}/x^i_{\mathcal{P}}) - \alpha_i] \leq \theta^j_{\mathcal{T}_j} \leq [\tan^{-1}(y^i_{\mathcal{P}}/x^i_{\mathcal{P}}) + \alpha_i] \tag{11}$$

Then, K_v is generated by the unit vectors,

$$\hat{\xi}_i(t) = \begin{bmatrix} \cos\theta^j_{\mathcal{T}_j}\sin[\pi/2 - \eta_i(t)] \\ \sin\theta^j_{\mathcal{T}_j}\sin[\pi/2 - \eta_i(t)] \\ \cos[\pi/2 - \eta_i(t)] \end{bmatrix} \tag{12}$$

and,

$$\hat{\omega}_i(t) = \begin{bmatrix} \cos\theta^j_{\mathcal{T}_j}\sin[\pi/2 - \mu_i(t)] \\ \sin\theta^j_{\mathcal{T}_j}\sin[\pi/2 - \mu_i(t)] \\ \cos[\pi/2 - \mu_i(t)] \end{bmatrix}, \tag{13}$$

where

$$\eta_i, \mu_i = \tan^{-1}\left[\frac{1}{v^i_{\mathcal{T}_{min}}, v^i_{\mathcal{T}_{max}}}\right] \tag{14}$$

$$\begin{aligned} v^i_{\mathcal{T}_{min}}, v^i_{\mathcal{T}_{max}} \;=\; & \frac{1}{t}\Big[x^i_{\mathcal{P}}\cos\theta^j_{\mathcal{T}_j} + y^i_{\mathcal{P}}\sin\theta^j_{\mathcal{T}_j} \\ & \mp\sqrt{r_i^2 - (x^i_{\mathcal{P}}\sin\theta^j_{\mathcal{T}_j} + y^i_{\mathcal{P}}\cos\theta^j_{\mathcal{T}_j})^2}\Big]. \end{aligned} \tag{15}$$

$v^i_{\mathcal{T}_{min}}$ and $v^i_{\mathcal{T}_{max}}$ represent the minimum and maximum target velocities that would cause a detection at t by sensor i as a function of $\theta^j_{\mathcal{T}_j} \in K_\theta$. If $\theta^j_{\mathcal{T}_j} \notin K_\theta$, then there are no target velocities that can cause such a detection. Thus, K_v is a *velocity cone* containing all target velocities that would cause a detection by the i^{th} sensor at t, expressed as the arc-tangent of $(1/v^j_{\mathcal{T}_j})$, as a function of heading angle. The cone K, its two-dimensional representations K_θ and K_v, and the generating unit vectors are computed and plotted in Fig. 1 for a sample time t_k, and for $p^j_{\mathcal{T}_j} = 0$.

Since K_θ and K_v represent all possible Markov tracks that are detected by sensor i at time t, the sum of their opening angles,

$$\begin{aligned} \psi_i(t) &= \sin^{-1}||\hat{l}_i(t)\times\hat{h}_i(t)|| \\ &= H(\det[\hat{l}_i(t)\quad \hat{h}_i(t)]^T)\sin^{-1}(\det[\hat{l}_i(t)\quad \hat{h}_i(t)]^T) \end{aligned} \tag{16}$$

and,

$$\begin{aligned} \zeta_i(t) &= \sin^{-1}||\hat{\omega}_i(t)\times\hat{\xi}_i(t)|| \\ &= H(\det[\hat{\omega}_i(t)\quad \hat{\xi}_i(t)]^T)\sin^{-1}(\det[\hat{\omega}_i(t)\quad \hat{\xi}_i(t)]^T) \end{aligned} \tag{17}$$

is a Lebesgue measure over the set of stabbers of $\mathcal{D}_i(t)$, and can be used to obtain the probability of detection of unobserved tracks. We order all coplanar unit vectors based on the orientation of an inertial reference frame such that two

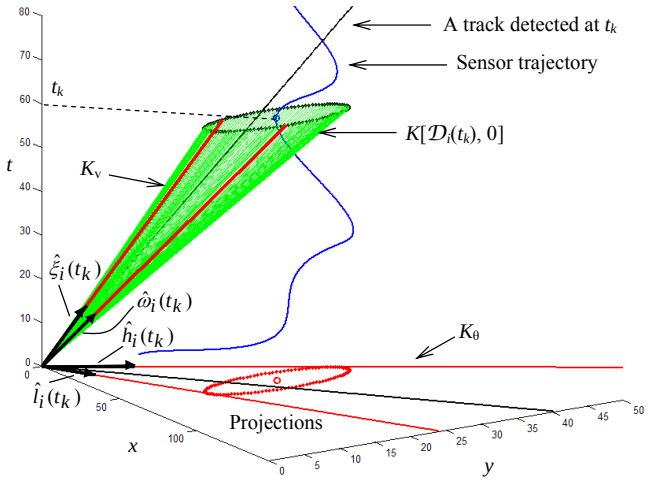

Fig. 1. Coverage cone K (green) and two-dimensional representations K_θ and K_v (red), with generating unit vectors, at time t_k.

coplanar vectors $u_i \prec u_j$ if when these vectors are translated such that their origins coincide, and u_i is rotated through the smallest possible angle to meet u_j, this orientation is in the same direction as the orientation of the reference frame. Thus, the Heaviside function $H(\cdot)$ in (16) and (17) guarantees that if $\hat{l}_i(t) \succ \hat{h}_i(t)$, or $\hat{\omega}_i(t) \succ \hat{\xi}_i(t)$, the corresponding opening angles are equal to zero.

From (3),(7)-(17), it can be seen that if r_i is a constant, the opening angles of the heading and velocity cones can be written as explicit functions of the sensor position, and of the target position at the discontinuities, i.e., $\psi_i = \psi_i[p^i_{\mathcal{P}}(t), p^j_{\mathcal{T}_j}]$ and $\zeta_i = \zeta_i[p^i_{\mathcal{P}}(t), p^j_{\mathcal{T}_j}, \theta^j_{\mathcal{T}_j}]$. For unobserved tracks, the Markov motion parameters $\mathcal{M}_j$ can be assumed independent and uniformly distributed. Thus, the joint probability mass function (PMF) between two consecutive discontinuities (or maneuvering time instants) $f_t(t_\jmath, t_{\jmath+1})$ is uniformly distributed over discrete time intervals $\Delta t_\jmath$, $\jmath = 1, \ldots, \rho^j$, with values defined by the user. During every time interval, the target heading has a probability density function (PDF) $f_\theta(\theta^j_{\mathcal{T}_j})$ that is uniform over the interval $\mathcal{I}_{\theta_{\jmath-1}} \equiv [\theta^j_{\mathcal{T}_{\jmath-1}} - \pi/2,\ \theta^j_{\mathcal{T}_{\jmath-1}} + \pi/2]$, due to a maximum turning radius of $\pi/2$. Similarly, the velocity of every target has a PDF $f_v(v^j_{\mathcal{T}_j})$ that is uniform over the interval $[v_{\mathcal{T}_{\min}},\ v_{\mathcal{T}_{\max}}]$, with $v_{\mathcal{T}_{min}} > 0$. It follows that the probability that a target j is detected by the network during the time interval Δt_j is given by,

$$P_{\mathcal{S}_j} = \sum_{i=1}^{N} \int_{\mathcal{S}} f_p(p^j_{\mathcal{T}_j}) \int_{t_j}^{t_{\jmath+1}} \{\psi_i[p^i_{\mathcal{P}}(t), p^j_{\mathcal{T}_j}] + \sum_{\theta^j_{\mathcal{T}_j} \in \{K_\theta \cap \mathcal{I}_\theta\}} \zeta_i[p^i_{\mathcal{P}}(t), p^j_{\mathcal{T}_j}, \theta^j_{\mathcal{T}_j}]\}\ dt\ dp^j_{\mathcal{T}_j}, \quad (18)$$

where, $f_p(p^j_{\mathcal{T}_j})$ is the PDF of the spatial discontinuities (or maneuvering positions) in $\mathcal{S}$, which can be computed based on the position of the obstacles and other prior information about the AOI. By marginalizing (18) over all maneuvering times, the total probability of detection for a target j with an unobserved Markov track in $\mathcal{S}$ can be obtained and written as

$$P_{\mathcal{S}} = \sum_{j=1}^{\rho^j} f_t(t_\jmath, t_{\jmath+1}) \int_{\mathcal{S}} f_p(p^j_{\mathcal{T}_j}) \int_{t_\jmath}^{t_{\jmath+1}} \sum_{i=1}^{N} \{\psi_i[p^i_{\mathcal{P}}(t), p^j_{\mathcal{T}_j}] + \sum_{\theta^j_{\mathcal{T}_j} \in \{K_\theta \cap \mathcal{I}_\theta\}} \zeta_i[p^i_{\mathcal{P}}(t), p^j_{\mathcal{T}_j}, \theta^j_{\mathcal{T}_j}]\}\ dt\ dp^j_{\mathcal{T}_j}, \quad (19)$$

When a sensor moves to a cell κ_l, the new network configuration is approximated by $\mathcal{X}_l = \{p^1_{\mathcal{T}}, \ldots, p^i_{\mathcal{T}} \subset \bar{q}^l_{\mathcal{T}}, \ldots, p^N_{\mathcal{T}}\}$, letting the center of the sensors' field-of-view, $p^i_{\mathcal{T}}$, coincide with the centroid $\bar{q}^l_{\mathcal{T}}$ of κ_l. Thus, the gain in probability of detection for unobserved tracks that is associated with moving between two nodes $\kappa_l \rightarrow \kappa_i$ in $\mathcal{G}$ is

$$\Delta P_{\mathcal{S}}(\kappa_l, \kappa_i) \equiv P_{\mathcal{S}}(\mathcal{X}_i) - P_{\mathcal{S}}(\mathcal{X}_l), \quad (20)$$

The gain $\Delta P_{\mathcal{S}}$ is negative when the above change in configuration leads to a decreased probability of detection of unobserved tracks. However, since sensors in detection mode are moved according to (23) and both $P_{\mathcal{R}}$ and $P^k_{\mathcal{S}}$ pertain the same set of targets $\mathcal{T}$, the overall probability of detection (4) increases at every round of the game.

B. Probability of detection of partially-observed tracks

The partially-observed tracks are viewed as an opportunity for obtaining additional measurements before investing in the costly resources needed to capture a target. Cell decomposition can be used to account for the geometry of the sensors' FOVs, platform, and for the estimated target tracks [11]. While the sensor's platform must avoid intersecting obstacles to prevent collisions, the sensor's FOV must intersect the targets in order to enable additional measurements. Let $\mathcal{F}_{\mathcal{A}_i}$ denote a moving Cartesian frame embedded in $\mathcal{A}_i$. The configuration $q^i_{\mathcal{P}}$ specifies the position and orientation of $\mathcal{F}_{\mathcal{A}_i}$ with respect to the inertial frame $\mathcal{F}_{\mathcal{S}}$. If we assume that $\mathcal{D}_i$ and $\mathcal{A}_i$ are both rigid, then $q^i_{\mathcal{P}}$ also specifies the position of every point in $\mathcal{D}_i$ (or $\mathcal{A}_i$) relative to $\mathcal{F}_{\mathcal{S}}$. Then, using the latest estimate of the partially-observed tracks, it is possible to identify the subset of $\mathcal{S}$ where the sensors may obtain additional target measurements. Let a C-target be defined as a bounded subset of the sensor's configuration space $\mathcal{C}$ that contains only values of $q^i_{\mathcal{P}}$ for which the probability of obtaining additional target detections is above a user-defined threshold $\epsilon > 0$. Let $\mathcal{C}_i$ denote the configuration space of the i^{th} sensor, and let $\mathcal{R}^t_j$ denote the Markov track of a target estimated from $Z^t_j \neq \emptyset$. Then, the j^{th} C-target can be defined as follows:

Definition 10: The target track $\mathcal{R}^\tau_j \in \mathcal{S}$ maps into the i^{th} sensor configuration space $\mathcal{C}_i$ to the C-target region $\mathcal{CR}^\tau_j = \{q^i_{\mathcal{P}}(t) \in \mathcal{C}_i \mid \Pr\{\mathcal{D}_i(t) \cap \mathcal{R}^\tau_j\} > \epsilon, \forall t \geq \tau, i \in \mathcal{I}_{\mathcal{P}}, j \in \mathcal{I}_{\mathcal{T}}\}$.
Moreover, for a Markov motion process, the C-targets have the following property:

Proposition 1: The C-target of a Markov target with bounded heading angle can be approximated by a bounded subset of a two-dimensional cone.
The proof, is provided in [16].

The cell decomposition approach presented in [11] is implemented in order to discretize a configuration that contains both C-obstacles and C-targets, and obtain the connectivity graph defined in Definition 8. Uniform cell decomposition is used because of its implementation tractability and optimal dispersion [22]. Optimal dispersion (in $\mathcal{L}_2$ norm) is important since the sensing regions are circles with finite area. In order to treat heterogeneous networks, every sensor has its own connectivity graph, which is updated as the game progresses and is denoted by $\mathcal{G}_i^t$. The probability density function $P_{\mathcal{R}}$, used to compute the reward (4), is obtained as follows. Suppose $\underline{\kappa}_l$ is one of the c_j observation cells $\mathcal{G}_i^t$ and is obtained from the decomposition of the j^{th} C-target at time t: $\underline{\kappa}_l \subset \mathcal{CR}_j^t$. Then, the sensing benefit of visiting $\underline{\kappa}_l$ is the probability of detecting target j,

$$P_{\mathcal{R}}(\underline{\kappa}_l \subset \mathcal{CR}_j^t) = \Pr\{D_{ji} = 1/c_j \mid e_{ji} \leq r_i\}, \tag{21}$$

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, $P_{\mathcal{R}}$ is assumed to be uniform over $\mathcal{CR}_j^t$, and if the l^{th} cell is void $P_{\mathcal{R}}(\kappa_l) = 0$. In general, this probability can be obtained from the tracking algorithm, and is dependent on time and on the distance from the last detection.

This completes the definition of the sensing reward in (4). The next section presents a methodology for computing control policies for sensors in detection and pursuit mode, based on $\mathcal{G}_i^t$ and its labels computed according to (4).

C. Detection and Pursuit Policies

At the on-set of the game, $Z^0 = \emptyset$, and all tracks are unobserved, with $P_{\mathcal{R}} = 0$. Thus, all N pursuers in $\mathcal{P}$ are placed simultaneously in $\mathcal{S}$ by maximizing (19). Since the sensors are omnidirectional, the orientation does not influence the region covered by each field of view, and the pursuers are placed by determining their initial positions $\mathcal{X}_0 = \{p_{\mathcal{T}}^1(t_0), \ldots, p_{\mathcal{T}}^N(t_0)\}$ from the following optimization problem,

$$\mathcal{X}_0^* = \arg\max_{\mathcal{X}} \mathcal{P}_{\mathcal{S}}(\mathcal{X}) \tag{22}$$

with $0 \leq x_{\mathcal{P}}^i \leq L$ and $0 \leq y_{\mathcal{P}}^i \leq L$, $\forall i \in I_{\mathcal{P}}$. The above optimization amounts to a nonlinear program that can be solved by sequential quadratic programming [23], [24]. After all sensors are placed at $\mathcal{X}_0^*$ at an user-defined orientation, their initial configurations, $q_{\mathcal{T}}^1(t_0), \ldots, q_{\mathcal{T}}^N(t_0)$, are known and the game begins.

At every game round, a sensor is deployed in detection or pursuit mode, depending on whether the new track is partially or fully observed, respectively. When the sensor is deployed in detection mode, its obstacle-free trajectory is computed from the sequence of cells or *channel* that maximizes its total reward, i.e.,

$$\mu_d^* \equiv \{\kappa_0, \ldots, \kappa_f\}^* = \arg\max_{\mu} \sum_{(\kappa_l, \kappa_i) \in \mu} R(\kappa_l, \kappa_i). \tag{23}$$

where, κ_f is chosen as the observation cell with the highest cumulative probability in $\mathcal{G}_i^t$, i.e., $\kappa_f = \arg\max_{\underline{\kappa}_i}(P_{\mathcal{R}}(\underline{\kappa}_i) + P_{\mathcal{S}}(\underline{\kappa}_i))$. Since the detection probabilities may vary slightly within each cell, they are computed in reference to the geometric centroid $\bar{q}_i$ of every cell κ_i. Then, the optimal channel μ_d^* is computed from $\mathcal{G}_i^t$ using the A* graph searching algorithm [14], and it is mapped into a set of waypoints that are used by a trajectory generator and trajectory tracking controller to determine the pursuer policy $u_{\mathcal{P}}^i = c^i[q_{\mathcal{P}}^i(t), Z^t]$.

When a sensor is switched to pursuit mode, an approach motivated by the potential field controller described in [25] is used to compute a policy that will lead the sensor to an interception waypoint, $\delta \in \mathcal{S}$. The time to interception $t_{c_i^j}$ and δ are computed simultaneously by Newton's method [26], using the approach described in [18]. Then, in order to find an obstacle-free shortest path between the sensor position at the time of deployment, say $p_{\mathcal{T}}^i(t)$, and δ, the connectivity graph $\mathcal{G}_i^t$ is modified by attaching to every arc $(\kappa_l, \kappa_i) \in \mathcal{G}_i^t$ the Euclidean distance in (5). The channel μ_p^* of shortest overall distance between $\kappa_\ell \ni q_{\mathcal{T}}^i(t)$ and $\kappa_r \ni \delta$ (assuming zero heading at δ) in $\mathcal{G}_i^t$ is determined by the graph searching algorithm A* [14]. Subsequently, μ_p^* is mapped into a set of waypoints in $\mathbb{R}_+^2$ that are used by a inner-loop potential-field controller described in [25].

The methodology described in this section is demonstrated through numerical simulations in the next section.

IV. Simulations and numerical results

In many surveillance applications, multiple heterogeneous sensors are deployed in the AOI to detect, track, and eventually pursue multiple moving targets with little or no prior information about their tracks. The methodology presented in Section III is demonstrated by simulating a surveillance system comprised of sensors installed on UGVs, and sensors installed on UAVs, that cooperate to pursue moving ground targets. The sensors on UGVs have smaller FOVs and are slower than those on UAVs, but can capture the targets. Unlike UGVs, the UAVs can fly over obstacles, and but can only be used in detection mode. The properties of both types of sensors and platforms are taken into account by the control methodology and algorithm.

The dot product of the target velocity vector with its initial velocity $[\dot{x}_{\mathcal{T}}^j(t_0)\ \dot{y}_{\mathcal{T}}^j(t_0)]^T$ is positive at all times to ensure that the targets traverse $\mathcal{S}$. The targets' headings are sampled randomly from $f_\theta(\theta_{\mathcal{T}_\jmath}^j)$ using user-defined time intervals $\Delta t_\jmath$, $\jmath = 1, \ldots, \rho^j$, that are based on the properties of the targets and the obstacles. The sensor network is assumed to be equipped with wireless communication devices that allow the sensors to download control policies every time their mode or configuration change, or when new target measurements are obtained. A sensor detects targets with a non-zero probability when they enter its FOV, allowing the sensor to also obtain noisy measurements of the target's position. A fusion and tracking algorithm is used to update the tracks' estimates every time new measurements are obtained and associated with a target.

Figures 2-6 illustrate a simulated game from the initial time t_0 to the final time t_f, when all targets have been captured and

removed from the AOI. In the surveillance system simulation, red squares represent UGVs in detection mode, green squares represent UGVs in pursuit mode, purple squares represent UAVs, and the circles denote the corresponding FOVs. The targets are represented by blue polygons, and their tracks are plotted only once they have been estimated from Z. The obstacles are represented by black polygons, and the C-targets are represented by light-blue cone-like regions.

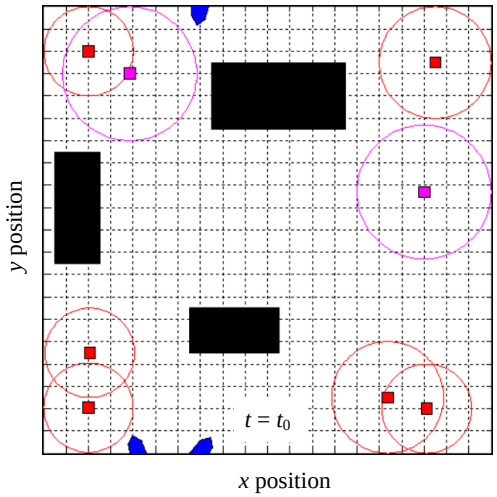

Fig. 2. Simulated workspace and sensor network at the initial time t_0.

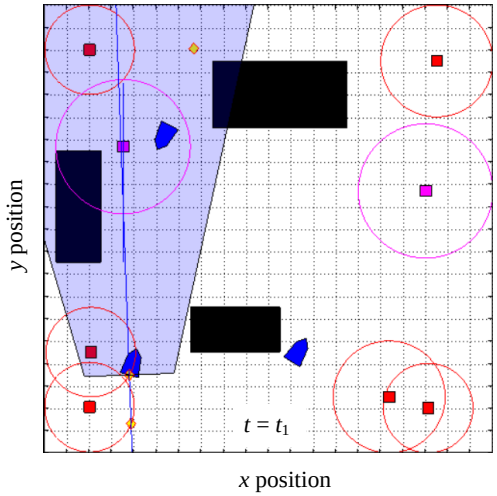

Fig. 3. The sensor network reconfigures to maximize the sensing reward based on three detections at t_1.

As shown in Fig. 2, six sensors on UGVs and two sensors on UAVs are placed in $\mathcal{S}$ to detect, track, and pursue three targets, with $k = 2$. In this game, the targets enter $\mathcal{S}$ at t_0 (Fig. 2), when all sensors are in *detection* mode. As multiple $1 < l < k$ sensor detections (symbolized by yellow diamonds) are obtained, the sensors move to maximize the probability of detecting partially-observed tracks, based on their current estimates (Figs. 3-4). When $k = 2$ detections are obtained

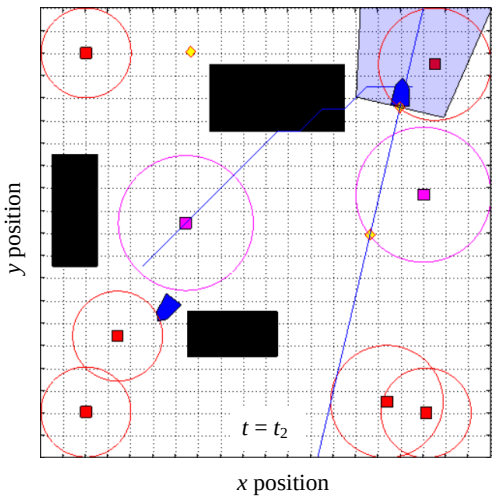

Fig. 4. The sensor network reconfigures to maximize the sensing reward after one target has been captured based on two additional detections at t_2.

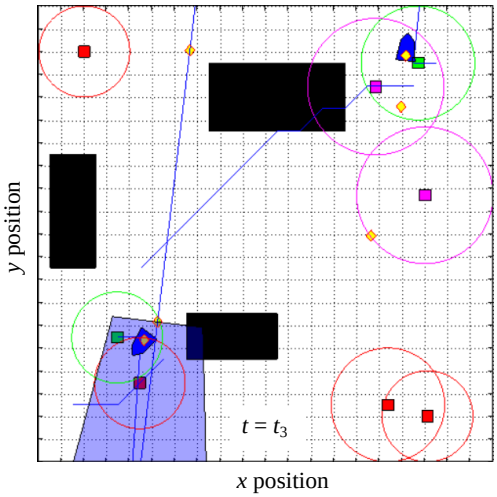

Fig. 5. Two sensors are switched to pursuit mode based on two fully-observed tracks at t_3.

and associated with the same target, its track becomes fully observed, and the connectivity graphs $\mathcal{G}_i^t$, $i = 1, \ldots, N$, are used to select a sensor candidate to switch to *pursuit* mode (Fig. 5). By this approach, the game is guaranteed to terminate [18], and at the end time t_f all targets are captured and removed from $\mathcal{S}$, as shown in Fig. 6.

V. Conclusion

This paper presents an integrated geometric-optimization framework for computing sensor control policies in networks comprised of multiple heterogeneous sensors that seek to detect and intercept multiple mobile targets. The sensor policies are computed to optimize the probability of detecting target tracks that can be modeled by a Markov motion process, with little or no prior information. After one or more detections are obtained from the targets, the control policies also optimize a tradeoff between the probability of obtaining additional

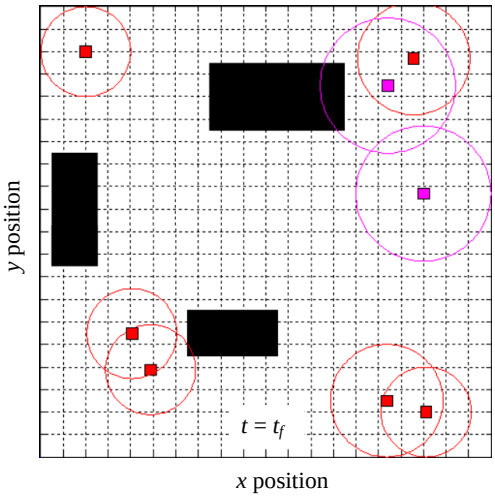

Fig. 6. The game ends when all targets have been captured and removed from $\mathcal{S}$ at t_f.

detections from partially-observed tracks, and the other sensors' objectives. The approach allows to integrate obstacle-avoidance and minimum-distance objectives, and to compute pursuit control policies by means of the same connectivity graphs and track estimates that are used to maximize the sensing objectives. The approach is demonstrated through progressive simulation scenarios involving multiple sensors installed on UGVs and UAVs, which are characterized by different sensing and motion capabilities, but are deployed to cooperatively detect, track, and pursue the same set of maneuvering targets.

ACKNOWLEDGMENTS

This work is supported in part by the Office of Naval Research (Code 321), and by NSF grant ECS CAREER #0448906. The work of R. Fierro was supported by NSF grants ECCS CAREER #0811347, IIS #0812338, and CNS #0709329.

REFERENCES

[1] S. Ge and F. Lewis, *Autonomous Mobile Robots: Sensing, Control, Decision-Making, and Applications*. Boca Raton, FL: CRC Press, Taylor and Francis Group, 2006.

[2] A. Arsie and E. Frazzoli, "Efficient routing of multiple vehicles with no explicit communications," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 2, pp. 154–164, January 2007.

[3] P. Yang, R. Freeman, and K. Lynch, "Distributed cooperative active sensing using consensus filters," Rome, Italy, April 2007, pp. 405–410.

[4] 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.

[5] J. C. F. Bullo and S. Martinez, *Distributed Control of Robotic Networks*. Princeton University Press, 2008.

[6] E. U. Acar, "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods," *International Journal of Robotic Research*, vol. 22, pp. 7–8, 2003.

[7] H. Choset, "Coverage for robotics: A survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[8] H. Gonzales-Banos, "A randomized art-gallery algorithm for sensor placement," in *Proc. 17^{th} Annual Symposium on Computational Geometry (SCG'01)*, Medford, Massachusetts, 2001, pp. 232 – 240.

[9] A. Ganguli, J. Cortés, and F. Bullo, "Maximizing visibility in nonconvex polygons: Nonsmooth analysis and gradient algorithm design," *SIAM Journal on Control and Optimization*, vol. 45, no. 5, pp. 1657–1679, 2006.

[10] M. Qian and S. Ferrari, "Probabilistic deployment for multiple sensor systems," in *Proc. of the 12^{th} SPIE Symposium on Smart Structures and Materials: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, vol. 5765, San Diego, 2005, pp. 85–96.

[11] C. Cai and S. Ferrari, "Information-driven sensor path planning by approximate cell decomposition," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, vol. 39, no. 3, pp. 672–689, 2009.

[12] 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.

[13] B. Perteet, J. McClintock, and R. Fierro, "A multi-vehicle framework for the development of robotic games: The Marco Polo case," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, April 10-14 2007, pp. 3717–3722.

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

[15] K. C. Baumgartner and S. Ferrari, "A geometric transversal approach to analyzing track coverage in sensor networks," *IEEE Transactions on Computers*, vol. 57, no. 8, pp. 1113–1128, 2008.

[16] D. Tolic, R. Fierro, and S. Ferrari, "Cooperative multi-target tracking via hybrid modeling and geometric optimization," in *Proc. of the 19^{th} Mediterranean Conference on Control and Automation*, Thessaloniki, Greece, 2009, pp. 440–445.

[17] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*. Hoboken, NJ: Wiley Interscience, 2001.

[18] S. Ferrari, R. Fierro, B. Perteet, C. Cai, and K. Baumgartner, "A geometric optimization approach to detecting and intercepting dynamic targets using a mobile sensor network," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 292–320, 2009.

[19] J. Gubner, *Probability and Random Processes for Electrical and Computer Engineers*. Cambridge, UK: Cambridge University Press, 2006.

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

[21] J. E. Goodman, R. Pollack, and R. Wenger, "Geometric transversal theory," in *New Trends in Discrete and Computational Geometry*, J. Pach, Ed. Springer Verlag, 1991, pp. 163–198.

[22] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[23] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 2007.

[24] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. Hoboken, NJ: Wiley Interscience, 2006.

[25] J. Clark and R. Fierro, "Mobile robotic sensors for perimeter detection and tracking," *ISA Transactions*, vol. 46, no. 1, pp. 3–13, 2007.

[26] J. J. Moré and M. Y. Cosnard, "Numerical solution of nonlinear equations," *ACM Transactions on Mathematical Software*, vol. 5, no. 1, pp. 64–85, 1999.