# A Geometric Transversals Approach to Analyzing the Probability of Track Detection for Maneuvering Targets

Hongchuan Wei and Silvia Ferrari, *Senior Member*, *IEEE*

**Abstract**—There is considerable precedence in the sensor tracking and estimation literature for modeling maneuvering targets by Markov motion models in order to estimate the target state from multiple, distributed sensor measurements. Although the transition probability density functions of these Markov models are routinely outputted by tracking and estimation algorithms, little work has been done to use them in sensor coordination and control algorithms. This paper presents a geometric transversals approach for representing the probability of track detection by multiple, distributed sensors, as a function of the Markov model transition probabilities. By this approach, the Markov parameters of maneuvering targets that may be detected by the sensors are represented by three-dimensional cones that are finitely generated by the sensors fields-of-view in a spatiotemporal Euclidian space. Then, the problem of deploying a sensor network for the purpose of maximizing the expected number of target detections can be formulated as a nonlinear program that can be solved numerically for the optimal sensor placement. Numerical results show that the optimal sensor placements obtained by this geometric transversals approach significantly outperform greedy, grid, or randomized sensor deployments.

**Index Terms**—Detection theory, geometric transversals, nonlinear optimization, sensor networks, target tracking, track coverage



## 1 Introduction

The problem of placing multiple sensors for the purpose of providing a desired quality-of-service (QoS) in a region-of-interest (RoI), also known as *sensor network deployment*, is relevant to a wide range of sensor applications, including security and surveillance, environmental and atmospheric monitoring, and tracking of endangered species [1], [2]. In particular, when sensors are deployed in order to cooperatively detect and track moving targets in an RoI they can be placed to optimize the network's QoS known as track coverage. Track coverage represents the ability of a sensor network to cooperatively obtain non-simultaneous detections of a single target during its transit through the RoI [3]–[5]. As a result, track coverage is related to the probability of cooperatively detecting target tracks over time.

A target track is said to be detected when it can be formed from multiple independent sensor detections using an assumed prior spatio-temporal model. Multiple independent detections are required by cost-effective sensors that have limited detection capabilities, and are subject to frequent false alarms. Existing track coverage functions have been successfully utilized in deployment, control, and coordination algorithms to significantly increase the effectiveness of the sensor network by controlling and, in some cases, optimizing QoS with respect to the sensors' positions [4]–[12]. However, these existing track coverage functions assume that the targets travel with constant heading and speed [5]–[10], or that the sensors are uniformly distributed and have constant range [4], [11], [12].

The method presented in this paper relaxes all of these assumptions, and extends the geometric transversals approach in [5] to a three-dimensional Euclidian space representing the sensor-target spatio-temporal coordinates. In this space, the Markov parameters of maneuvering targets can be represented by three-dimensional cones that are finitely generated by the sensors' fields-of-view (FOVs). There is considerable precedence in the sensor tracking and estimation literature for modeling target tracks by Markov motion models in order to estimate the target state from multiple, distributed sensor measurements [13]. Although the transition probability density functions of these Markov models are routinely outputted by tracking and estimation algorithms [13], little work has been done to use them as a feedback to sensor coordination and control algorithms.

This paper derives the probability of track detection in closed-form, as a function of the target Markov model transition probabilities and of the sensors' ranges and positions in the RoI. The motivation for deriving coverage functions expressing the QoS of wireless sensor networks in closed form is that they can be utilized to deploy the sensors via control and optimization theory and algorithms [5]–[7]. The numerical simulations presented in this paper show that sensors deployed by optimizing this new track coverage function are significantly more effective than sensors deployed using other applicable deployment methods, such as, greedy or incremental algorithms [14]–[16], grid placement algorithms [17], circle-packing algorithms [18], and randomized strategies [19].

- *The authors are with the Laboratory for Intelligent Systems and Controls (LISC), Duke University, Durham, NC 27708-0005. E-mail: {hongchuan.wei, sferrari}@duke.edu.*

*Manuscript received 21 June 2012; revised 22 Nov. 2012; accepted 28 Jan. 2013; published online 25 Feb. 2013.*
*Recommended for acceptance by K. Li.*
*For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-2012-06-0426.*
*Digital Object Identifier no. 10.1109/TC.2013.43*

## 2 Problem Formulation

This paper addresses the problem of deploying a network of $n$ fixed omnidirectional sensors for the purpose of obtaining multiple track detections for a maneuvering target. The target is assumed to obey a Markov motion model that is commonly implemented by multi-sensor multi-target tracking algorithms [20], [21]. In Markov motion models, the target movement is modeled by a Markov chain characterized by probability distributions that are typically computed via Kalman filtering based on prior sensor measurements.

A Markov chain is defined as a sequence of correlated random variables, $X_1, X_2, \ldots$, that obey the Markov property by which the future states only depend on past states through the present state, i.e.,

$$\begin{aligned}&\Pr(X_{k+1} = x_j \mid X_k = x_i, \ldots, X_1 = x_l)\\ &= \Pr(X_{k+1} = x_j \mid X_k = x_i),\end{aligned} \tag{1}$$

where, lower-case letters denote numerical values of the random variables [22]. The probability law, or probability function, denoted by $\Pr(\cdot)$, obeys the three axioms of probability [22]. Then, based on the Markov property, the evolution of the state can be described by a transition probability function $f_{X_{k+1}|X_k}(x_j \mid x_i)$ [23].

Let the random variables $\theta$ and $v$ represent the target heading and velocity, respectively. Consider the target motion in a region-of-interest (RoI) $\mathcal{A} \subset \mathbb{R}^2$, during a finite time interval $(T_0, T_f]$. A three-dimensional real-valued vector function maps the family of random variables $\{\theta(t), v(t)\}$ into the random vector $\mathbf{x}(t) = [x(t)\, y(t)]^T$ at every time $t \in (T_0, T_f]$,

$$\dot{\mathbf{x}}(t) = \begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\end{bmatrix} = \begin{bmatrix}v(t)\cos\theta(t)\\ v(t)\sin\theta(t)\end{bmatrix}, \quad t \in (T_0, T_f], \tag{2}$$

where $x$ and $y$ denote the target's $xy$-coordinates with respect to an inertial frame of reference embedded in $\mathcal{A}$, and the third component of the vector function is the identity function.

Assuming the target heading and velocity are constant during a time interval $(t_j, t_{j+1}] \subset (T_0, T_f]$, with $j = 1, \ldots, m$, where $\Delta t_j = (t_{j+1} - t_j)$ is not necessarily constant, the target motion can be modeled as a Markov chain as follows. Let $\mathbf{x}_j$ denote the target position at the beginning of the $j^{th}$ time interval, namely, $\mathbf{x}_j = \mathbf{x}(t_j)$, for $j = 1, \ldots, m$. The target heading and velocity during $(t_j, t_{j+1}]$ are denoted by two random parameters $\theta_j \in \mathcal{H}$ and $v_j \in \mathcal{V}$, respectively, where $\mathcal{H} = [\theta_{min}, \theta_{max}]$ represents the range of all possible target heading values, and $\mathcal{V} = [v_{min}, v_{max}]$ represents the range of all possible target velocity values. Then, the linear differential equation (2) can be integrated with respect to time to obtain the Markov motion model,

$$\mathbf{x}_{j+1} = \mathbf{x}_j + [v_j \cos\theta_j \quad v_j \sin\theta_j]^T \Delta t_j, \quad j = 1, \cdots, m, \tag{3}$$

for the time interval $(T_0, T_f]$. By this approach, the target motion is described by the evolution of the random variables $\theta_j$, $v_j$, and $\mathbf{x}_j$, referred to as Markov motion parameters, and denoted by the set $\mathcal{M} = \{\mathbf{x}_j, \theta_j, v_j\}_{j=1,\cdots,m}$.

As illustrated in Fig. 1, a realization of the above Markov motion model is a trajectory in which the heading and velocity are piece-wise constant, while the $xy$-coordinates are variables with discontinuities at every time instant $t_j$, $j = 1, \ldots, m$, when the target is said to *maneuver*, thereby changing both heading and velocity. These maneuvering time instants are not necessarily equally spaced and, in this paper, they are assumed known for simplicity. However, the approach can be easily extended to unknown maneuvering times, by considering $t_j$ as another random variable of the Markov motion model.

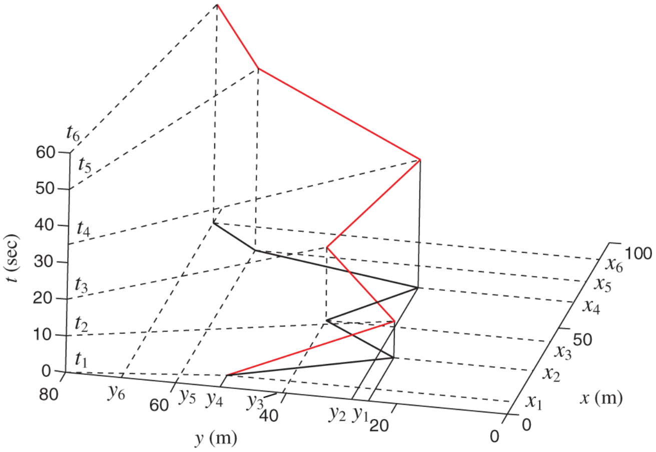


Fig. 1. Example of target track sampled from a Markov motion model in $\mathcal{A}$ (black) and in $\Omega$ (red).

The probability density functions (PDFs) of the Markov motion parameters, $\mathcal{M}$, are computed from prior measurements using target tracking algorithms [20], [21]. For simplicity, in this paper, it is assumed that these PDFs are given, and that, during every interval $(t_j, t_{j+1}]$, the Markov parameters $\{\mathbf{x}_j, \theta_j, v_j\}$ are independent random variables. Also, it is assumed that the target heading and velocity at the $j$th time interval, $(t_j, t_{j+1}]$, are independent of the Markov parameters at the $(j-1)$th interval, $(t_{j-1}, t_j]$. Thus, the probability that $\theta_j$ and $v_j$ take any of the values in their ranges during $(t_j, t_{j+1}]$ is given by the PDFs $f_\Theta^{(j)}(\theta_j)$ and $f_V^{(j)}(v_j)$, respectively. From (3) it can be seen that the probability that $\mathbf{x}_j$ takes any of its possible values in $\mathcal{A}$ depends on the values of the Markov parameters during the $(j-1)$th interval. Thus, the PDF of $\mathbf{x}_j$ can be obtained as follows,

$$\begin{aligned}f_X^{(j)}(\mathbf{x}_j) = \int_{\theta_{min}}^{\theta_{max}}\int_{v_{min}}^{v_{max}} & f_X^{(j-1)}(\mathbf{x}_{j-1} - [\cos\theta_{j-1}\ \sin\theta_{j-1}]^T v_{j-1}\Delta t_{j-1})\\ & \times f_\Theta^{(j-1)}(\theta_{j-1}) f_V^{(j-1)}(v_{j-1}) dv_{j-1} d\theta_{j-1},\end{aligned} \tag{4}$$

for $j = 1, \ldots, m$, where all of the PDFs are known from the previous time interval.

In this paper, the set of $n$ sensors deployed to detect and track the maneuvering target is assumed to be fixed and omnidirectional. In particular, it is assumed that every sensor can be described by an omnidirectional boolean sensing model that has been used to model a variety of sensors, including passive acoustic sensors, radars, and electromagnetic sensors [24]. Every sensor in the network, indexed by $i$, has a constant sensing range $r_i > 0$ that may depend on its detection threshold, and is to be placed at a position $\mathbf{s}_i = [s_{x_i}\ s_{y_i}]^T \in \mathcal{A}$. For simplicity, it is also assumed that the sensor positions are all fixed and deterministic, and that the

RoI is a square region $\mathcal{A} = [0, L] \times [0, L]$. Under these assumptions, the $i$th sensor field-of-view (FOV) can be represented by a disk $\mathcal{C}_i = \mathcal{C}[\mathbf{s}_i, r_i] \subset \mathcal{A}$, centered at $\mathbf{s}_i$, and with a constant radius $r_i$, for $i = 1, \ldots, n$. By definition, a sensor $i$ can only detect a target if it enters its FOV at time $t$. Thus, the sensing model is

$$P_d[\mathcal{C}_i, \mathbf{x}(t)] = \begin{cases} 0 : \|\mathbf{s}_i - \mathbf{x}(t)\| > r_i \\ 1 : \|\mathbf{s}_i - \mathbf{x}(t)\| \leq r_i, \end{cases} \tag{5}$$

where, $P_d$ is the probability of detection or, in other words, the probability that the target track will intersect $\mathcal{C}_i$ at time $t \in (T_0, T_f]$. The approach can also be extended to moving sensors and other ROIs, as will be shown in a separate paper.

We are now ready to formulate the three problems addressed in this paper, namely:

**Problem 2.1 (Probability of Detection).** Given the PDFs of the Markov parameters $\mathcal{M}$ for a target in an RoI $\mathcal{A} = [0, L] \times [0, L]$, find the probability of detection for an omnidirectional sensor $i$ at time $t \in (T_0, T_f]$ as a function of the range $r_i \in \mathbb{R}$, and position $\mathbf{s}_i \in \mathcal{A}$.

**Problem 2.2 (Track Coverage).** Given the PDFs of the Markov parameters $\mathcal{M}$ for a target in an RoI $\mathcal{A} = [0, L] \times [0, L]$, find the *track coverage* of $n$ sensors, defined as the expected number of target detections during a time interval $(T_0, T_f]$, as a function of the sensors' positions, $\mathbf{s}_1, \ldots, \mathbf{s}_n$, and ranges, $r_1, \ldots, r_n$.

**Problem 2.3 (Sensor Placement).** Given the PDFs of the Markov parameters $\mathcal{M}$ for a target in an RoI $\mathcal{A} = [0, L] \times [0, L]$, find the $n$ sensor positions $\mathbf{S}^* = [\mathbf{s}_1^{*T} \cdots \mathbf{s}_n^{*T}]^T$ that maximizes the expected number of target detections during a time interval $(T_0, T_f]$.

In the following sections, Problems 2.1–2.3 are addressed by extending the geometric transversals approach first proposed in [3], [5] to maneuvering targets described by Markov motion models.

## 3 Coverage Cone Representation of Tracks Detected by One Sensor

In this section, we show that all Markov target tracks detected by an omnidirectional sensor at time $t \in (t_j, t_{j+1}]$ are contained by a three-dimensional cone that can be used to define a Lebesgue measure of track coverage (Section 4). In [5], it was shown that in the case of non-maneuvering targets, a two-dimensional (2D) coverage cone (Fig. 2) can be used to derive a Lebesgue measure of all tracks through an intercept $b_y$ that are detected by an omnidirectional sensor positioned at $\mathbf{s}_i$. Because the Lebesgue measure is provided by the opening angle of the coverage cone, $2\alpha_i$, it follows that a measure of track coverage can be obtained by the dot product of the generating unit vectors $\hat{\mathbf{h}}_i$ and $\hat{\mathbf{l}}_i$, and, thus, can be written conveniently in closed-form, as a function of $\mathbf{s}_i$ and $r_i$. It also follows, under proper assumptions, that the probability of detection can be obtained solely as a function of $\alpha_i$ [5].

In the case of maneuvering targets, the track detection problem cannot be viewed as time invariant or, in other words, the probability of detection is an explicit function of time. Hence, we consider the spatio-temporal Euclidian subspace $\Omega = \mathcal{A} \times (T_0, T_f] \subset \mathbb{R}^3$, such that during $(t_j, t_{j+1}]$ the target track can be represented by a time-varying vector with constant orientation $\mathbf{m}_j(t) \in \Omega$, referred to as *Markov track*. Now, let the origin of $\mathbf{m}_j$, denoted by $\mathbf{z}_j = [\mathbf{x}_j^T t_j]^T$, coincide with the origin of a local coordinate frame $\mathcal{F}_j$. Then, as shown in Fig. 3, at any time $t \in (t_j, t_{j+1}]$, the Markov track $\mathbf{m}_j$ can be defined in cylindrical coordinates $(r, \theta, \zeta)$, where $\theta = \theta_j$ is a constant, $\zeta = (t - t_j)$ represents the time elapsed since the maneuvering time $t_j$, and $r = v_j \zeta$ represents the distance traveled. In other words, $\mathbf{m}_j$ is the target position at $t \in [t_j, t_{j+1}]$, relative to $\mathcal{F}_j$, which can be expressed as $(v_j(t - t_j), \theta_j, t - t_j)$ in cylindrical coordinates.

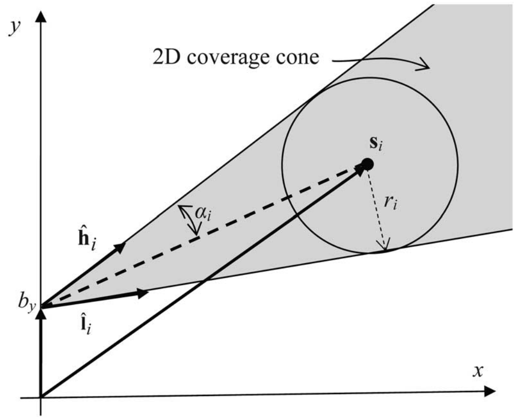


Fig. 2. Coverage cone for a target traveling along a straight line (adapted from [5]).

Given the nonempty subset $\mathcal{C}_i$ of $\Omega$, the cone generated by $\mathcal{C}_i$ is the set of all nonnegative combinations of the elements of $\mathcal{C}_i$, denoted by $K = \text{cone}(\mathcal{C}_i) \subset \Omega$ (see [25] for a review of cones and their properties). Then, in the remainder of the paper, we define the *coverage cone* of the $i$th sensor at $t \in (t_j, t_{j+1}]$ to be the cone generated by $\mathcal{C}_i$, with origin $\mathbf{z}_j \in \Omega$, as illustrated by the example in Fig. 3. The coverage cone is a basic construct for analyzing the probability of track detection because it represents the set of Markov tracks that can be detected by the $i$th sensor at time $t$. Since, in this paper, the origin $\mathbf{z}_j$ is a random but constant vector, the coverage cone is a time-dependent, random object in $\Omega$.

If we consider an inertial frame $\mathcal{F}_\Omega$, embedded in $\Omega$, the coverage cone at time $t \in (t_j, t_{j+1}]$ can be parameterized as follows,

$$K(t) = \Bigg\{ [x\,y\,z]^T \in \mathbb{R}^3 \mid \left\| [x\,y]^T - \frac{z - t_j}{t - t_j}(\mathbf{s}_i - \mathbf{x}_j) - \mathbf{x}_j \right\| \leq \frac{z - t_j}{t - t_j} r_i \Bigg\}, \tag{6}$$

where $\|\cdot\|$ denotes the $L_2$-norm. From the omnidirectional boolean sensing model, sensor $i$ has a non-zero probability to detect the target if and only if $\|\mathbf{x}(t) - \mathbf{s}_i\| \leq r_i$. It follows that the coverage cone (6) contains all Markov tracks that can be detected during $(t_j, t_{j+1}]$, as summarized by the following remark:

**Remark 3.1.** The coverage cone $K(t)$, defined in (6), contains the set of all Markov tracks with origin $\mathbf{z}_j = [\mathbf{x}_j^T t_j]^T \in \Omega$ that intersect the sensor FOV, $\mathcal{C}_i = \mathcal{C}[\mathbf{s}_i, r_i]$, at any time $t \in (t_j, t_{j+1}]$.

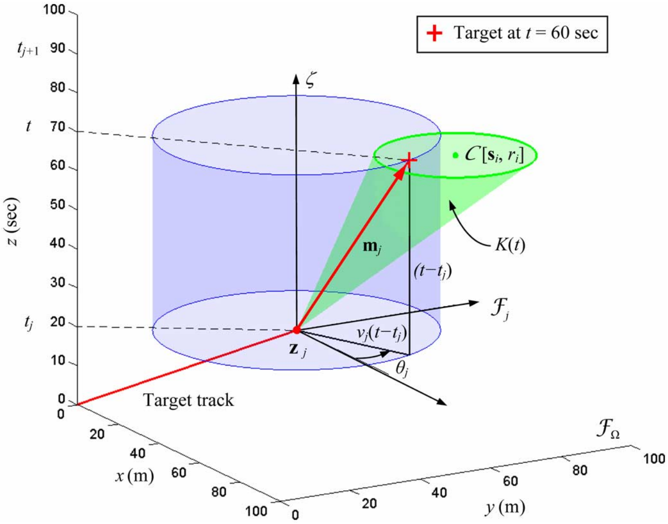


Fig. 3. Definition of Markov track and coverage cone for a maneuvering target in $\Omega$.

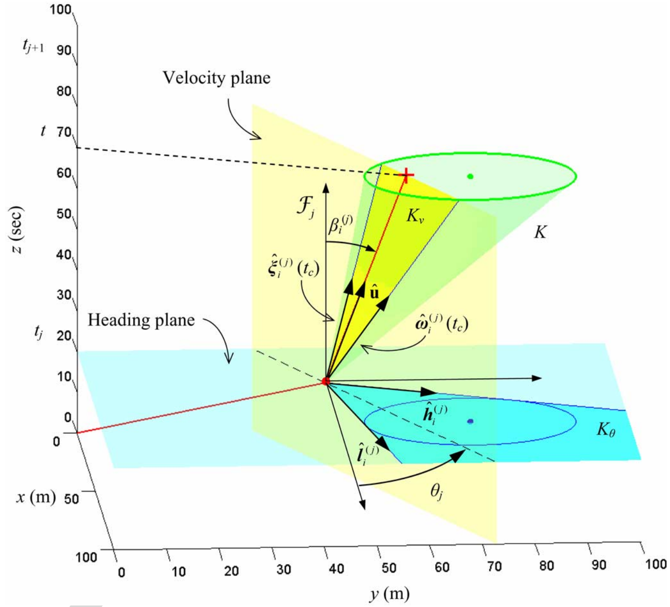


Fig. 4. Heading-cone (cyan) and velocity-cone (yellow) representation of coverage cone (green) for the example in Fig. 3.

**Proof.** Let $\mathbf{m}_j(t) \in \Omega$ be any Markov track with origin $\mathbf{z}_j \in \Omega$ that intersects $\mathcal{C}_i = \mathcal{C}[\mathbf{s}_i, r_i]$ at some time $t \in (t_j, t_{j+1}]$, i.e. $\mathbf{m}_j(t) \cap \mathcal{C}_i \neq \emptyset$. Let $\hat{\mathbf{u}} = [u_x \; u_y \; u_z]^T$ denote a unit vector in $\mathcal{F}_\Omega$ that is collinear with $\mathbf{m}_j$, and shares the same origin $\mathbf{z}_j$. Without loss of generality, assume the unit vector $\hat{\mathbf{u}}$ points from $\mathbf{z}_j$ to the target position in $\Omega$ at $t$, such that $u_z \geq 0$. It follows that $\mathbf{m}_j$ can be written in terms of $\hat{\mathbf{u}}$, i.e.:

$$\mathbf{m}_j(t) = \{\mathbf{y} \in \mathbb{R}^3 \mid \mathbf{y} = \mathbf{z}_j + c\hat{\mathbf{u}}, c \geq 0\}, \tag{7}$$

where the constant $c$ can be determined from the fact that $t = t_j + cu_z$. Moreover, the target position in $\mathcal{F}_\Omega$ at time $t$ can be obtained from $\mathbf{m}_j$ as follows,

$$\mathbf{x}(t) = \mathbf{x}_j + [(t - t_j)u_x/u_z \quad (t - t_j)u_y/u_z]^T.$$

From the boolean omnidirectional sensor model (5), a Markov track intersects the sensor's FOV $\mathcal{C}_i$ at some time $t \in (t_j, t_{j+1}]$ if and only if the target position satisfies $\|\mathbf{x}(t) - \mathbf{s}_i\| \leq r_i$, or:

$$\left\| \mathbf{x}_j + \frac{(t - t_j)}{u_z}[u_x \; u_y]^T - \mathbf{s}_i \right\| \leq r_i, \quad \text{for} \quad t \in (t_j, t_{j+1}]. \tag{8}$$

Considering any point $\mathbf{a} = [a_x \; a_y \; a_z]^T \in \mathbf{m}_j$, it follows from (7) that,

$$[a_x \, a_y \, a_z]^T = \mathbf{z}_j + \frac{(a_z - t_j)}{u_z}\hat{\mathbf{u}}, \tag{9}$$

and, thus, from the definition of $\hat{\mathbf{u}}$, any unit vector collinear with $\mathbf{m}_j$ satisfies

$$\begin{aligned} &\|[a_x \, a_y]^T - (a_z - t_j)(t - t_j)^{-1}(\mathbf{s}_i - \mathbf{x}_j) - \mathbf{x}_j\| \\ &= \left\| \mathbf{x}_j + \frac{(a_z - t_j)}{u_z}[u_x \; u_y]^T - \frac{(a_z - t_j)}{(t - t_j)}(\mathbf{s}_i - \mathbf{x}_j) - \mathbf{x}_j \right\| \\ &= \frac{(a_z - t_j)}{(t - t_j)} \left\| \frac{(t - t_j)}{u_z}[u_x \; u_y]^T - \mathbf{s}_i + \mathbf{x}_j \right\|. \end{aligned} \tag{10}$$

From (8) and (10), it follows that for any target at $\mathbf{a} \in \mathbf{m}_j$ to be detected by the $i$th sensor, the condition,

$$\left\| [a_x \, a_y]^T - \frac{(a_z - t_j)}{(t - t_j)}(\mathbf{s}_i - \mathbf{x}_j) - \mathbf{x}_j \right\| \leq \frac{(a_z - t_j)}{(t - t_j)} r_i, \tag{11}$$

must be satisfied.

Since (11) is equivalent to the parameterized representation of the coverage cone (6), it follows that for any target at $\mathbf{a} \in \mathbf{m}_j$ to be detected by the $i$th sensor at $t \in (t_j, t_{j+1}]$, $\mathbf{m}_j(t)$ must be contained by $K(t)$. □

## 4 Unit Vector Representation of Coverage Cone

The coverage cone $K$, defined in (6) and illustrated in Fig. 3, is a 3D circular cone that is possibly oblique. As a result, it is not easy to identify a Lebesgue measure for $K$ in terms of the Markov parameters $\mathcal{M}$. This section shows that it is possible to represent any 3D coverage cone with the parametrization (6) by two 2D cones that also provide lower and upper bounds for the heading and velocity of any Markov track $\mathbf{m}_j$ contained by $K$. The two cones, referred to as *heading cone* and *velocity cone*, are derived in the following subsections, and then illustrated through an example in Fig. 4.

### 4.1 Heading Cone

The heading cone, denoted by $K_\theta$, is obtained by projecting the 3D coverage cone onto a so-called *heading plane* defined as,

$$\{[x \, y \, z]^T \in \Omega \mid z = t_j\}, \tag{12}$$

such that it is parallel to the $xy$-plane and contains $\mathbf{z}_j$. Since the heading plane contains all of the possible target headings $\mathcal{H}$, the heading cone can be used to represent the headings of all targets detected by the $i$th sensor, during the $j$th time interval

$(t_j, t_{j+1}]$. In particular, the extremals of $K_\theta$ can be described in $\mathcal{F}_j$ by two unit vectors,

$$\hat{\boldsymbol{h}}_i^{(j)} = \begin{bmatrix} \cos\alpha_i^{(j)} & -\sin\alpha_i^{(j)} \\ \sin\alpha_i^{(j)} & \cos\alpha_i^{(j)} \\ 0 & 0 \end{bmatrix} \frac{\mathbf{p}_i^{(j)}}{\|\mathbf{p}_i^{(j)}\|} \equiv \begin{bmatrix} \cos\lambda_i^{(j)} \\ \sin\lambda_i^{(j)} \\ 0 \end{bmatrix}, \tag{13}$$

and

$$\hat{\mathbf{l}}_i^{(j)} = \begin{bmatrix} \cos\alpha_i^{(j)} & \sin\alpha_i^{(j)} \\ -\sin\alpha_i^{(j)} & \cos\alpha_i^{(j)} \\ 0 & 0 \end{bmatrix} \frac{\mathbf{p}_i^{(j)}}{\left\|\mathbf{p}_i^{(j)}\right\|} \equiv \begin{bmatrix} \cos\gamma_i^{(j)} \\ \sin\gamma_i^{(j)} \\ 0 \end{bmatrix}, \tag{14}$$

where, $\mathbf{p}_i^{(j)} \equiv (\mathbf{s}_i - \mathbf{x}_j)$ is the position of the sensor relative to the maneuvering point $\mathbf{x}_j$, and $\alpha_i^{(j)}$ is half the opening angle of $K_\theta$, that is,

$$\alpha_i^{(j)} = \sin^{-1}(r_i/\|\mathbf{p}_i^{(j)}\|). \tag{15}$$

Then, the heading cone can be defined as the cone that is finitely generated by $\hat{\boldsymbol{h}}_i^{(j)}$ and $\hat{\mathbf{l}}_i^{(j)}$, i.e.:

$$K_\theta[\mathcal{C}_i, \mathbf{z}_j] = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = c_1\hat{\boldsymbol{h}}_i^{(j)} + c_2\hat{\mathbf{l}}_i^{(j)}, c_1, c_2 \geq 0\}. \tag{16}$$

It can be seen that the heading cone is analogous to the 2D coverage cone first illustrated in Fig. 2, which was introduced in [5] to analyze track coverage for non-maneuvering targets.

### 4.2 Velocity Cone

Consider now a so-called *velocity plane* that represents the space of all spatio-temporal target coordinates in $\mathcal{F}_\Omega$ with a constant heading $\theta_j$. The velocity plane is defined as,

$$\{[x\, y\, z]^T \in \Omega \ (x\sin\theta_j - y\cos\theta_j) = [\sin\theta_j \ \cos\theta_j]\mathbf{x}_j, \ z \geq t_j\}, \tag{17}$$

such that it is perpendicular to $\mathcal{A}$, and contains $\mathbf{z}_j$. Then, the velocity cone, denoted by $K_v$, can be defined as the intersection of the 3D coverage cone (6), and the velocity plane (17).

Now, let all coplanar unit vectors be ordered based on the orientation of an inertial reference frame embedded in the velocity plane (17) such that, for any two coplanar unit vectors $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{u}}_j$, we say that $\hat{\mathbf{u}}_i \prec \hat{\mathbf{u}}_j$ if when these vectors are translated such that their origins coincide, and $\hat{\mathbf{u}}_i$ is rotated through the smallest possible angle to meet $\hat{\mathbf{u}}_j$, the rotation is in the same direction as the orientation of the reference frame. Then, $K_v$ contains all possible target velocities that would cause a detection by sensor $i$ at $t \in (t_j, t_{j+1}]$, provided the target heading $\theta_j$ satisfies the condition,

$$\hat{\mathbf{l}}_i^{(j)} \prec [\cos\theta_j \ \sin\theta_j]^T \prec \hat{\boldsymbol{h}}_i^{(j)}, \tag{18}$$

or, in other words, the Markov track is contained by both $K_v$, and $K_\theta$.

At time $t$, the extremals of $K_v$ can be described by two unit vectors in $\mathcal{F}_j$,

$$\hat{\boldsymbol{\xi}}_i^{(j)}(t) = \begin{bmatrix} \sin\eta_i^{(j)}\cos\theta_j \\ \sin\eta_i^{(j)}\sin\theta_j \\ \cos\eta_i^{(j)} \end{bmatrix}, \tag{19}$$

and,

$$\hat{\boldsymbol{\omega}}_i^{(j)}(t) = \begin{bmatrix} \sin\mu_i^{(j)}\cos\theta_j \\ \sin\mu_i^{(j)}\sin\theta_j \\ \cos\mu_i^{(j)} \end{bmatrix}, \tag{20}$$

where $\eta_i^{(j)}$ and $\mu_i^{(j)}$ are the angles that $\hat{\boldsymbol{\xi}}_i^{(j)}$ and $\hat{\boldsymbol{\omega}}_i^{(j)}$ make with the $z$-axis, respectively, defined as

$$\eta_i^{(j)}, \mu_i^{(j)} \equiv \tan^{-1}\left[\frac{1}{(t-t_j)}\left([\cos\theta_j \quad \sin\theta_j]\mathbf{s}_i - \mathbf{x}_j\right) \right. \\ \left. \mp\sqrt{r_i^2 - ([\sin\theta_j \quad -\cos\theta_j](\mathbf{s}_i - \mathbf{x}_j))^2}\right)\Bigg]. \tag{21}$$

For $t \geq t_j$, $\eta_i^{(j)}$ and $\mu_i^{(j)}$ are less than or equal to $\pi/2$. Then, the velocity cone can be defined as the cone that is finitely generated by $\hat{\boldsymbol{\xi}}_i^{(j)}$ and $\hat{\boldsymbol{\omega}}_i^{(j)}$, i.e.:

$$K_v[\mathcal{C}_i, \mathbf{x}_j] = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = c_1\hat{\boldsymbol{\xi}}_i^{(j)} + c_2\hat{\boldsymbol{\omega}}_i^{(j)}, c_1, c_2 \geq 0\}. \tag{22}$$

The results obtained in this section can be summarized by the following remark:

**Remark 4.1.** The coverage cone $K$, defined in (6), can be represented by the pair of 2D cones $\{K_\theta, K_v\}$, defined in (16) and (22).

**Proof.** Suppose $\mathbf{m}_j$ is any Markov track in the coverage cone $K$, or $\mathbf{m}_j \in K$, and let $\hat{\mathbf{u}} = [u_x \ u_y \ u_z]^T$ denote a unit vector that is collinear with $\mathbf{m}_j$, and has the same origin $\mathbf{z}_j$ where, all quantities are defined with respect to $\mathcal{F}_j$. Without loss of generality, assume $u_z > 0$. Then, $\mathbf{m}_j$ can be represented in terms of $\hat{\mathbf{u}}$ as shown in (7), and substituted into the coverage cone equation (6) to obtain

$$\left\|\mathbf{x}_j + c[u_x \ u_y]^T - \frac{cu_z}{(t-t_j)}(\mathbf{s}_i - \mathbf{x}_j) - \mathbf{x}_j\right\| \leq \frac{cu_z}{(t-t_j)} r_i. \tag{23}$$

Choose the nonnegative constant $c$ such that $cu_z = (t - t_j) \geq 0$, because $t \in (t_j, t_{j+1}]$ during the $j$th time interval. Then, (23) can be simplified to,

$$\left\|\frac{(t-t_j)}{u_z}[u_x \ u_y]^T + \mathbf{x}_j - \mathbf{s}_i\right\| \leq r_i, \tag{24}$$

where the vector $[(t-t_j)/u_z][u_x \ u_y]^T$ is the projection of the Markov track on the heading plane (12). The projection $[(t-t_j)/u_z][u_x \ u_y]^T$ can be expressed as a linear combination of $\hat{\boldsymbol{h}}_i^{(j)}$ and $\hat{\mathbf{l}}_i^{(j)}$, because $\hat{\boldsymbol{h}}_i^{(j)}$ and $\hat{\mathbf{l}}_i^{(j)}$ are linearly independent. Hence, there exist two nonnegative constants $c_1, c_2 \in \mathbb{R}$, such that

$$\frac{(t-t_j)}{u_z}[u_x \ u_y]^T = c_1\|\mathbf{p}_i^{(j)}\|\hat{\boldsymbol{h}}_i^{(j)} + c_2\|\mathbf{p}_i^{(j)}\|\hat{\mathbf{l}}_i^{(j)}, \tag{25}$$

where $\mathbf{p}_i^{(j)} \equiv (\mathbf{s}_i - \mathbf{x}_j)$ from Section 4.1.

Substituting (25) in (24), and dividing both sides by $\|\mathbf{p}_i^{(j)}\|$, (24) can be rewritten as,

$$\left\| c_1 \hat{\boldsymbol{h}}_i^{(j)} + c_2 \hat{\mathbf{l}}_i^{(j)} - \hat{\mathbf{p}}_i^{(j)} \right\| \leq \frac{r_i}{\|\mathbf{p}_i^{(j)}\|}, \tag{26}$$

where $\hat{\mathbf{p}}_i^{(j)} = \mathbf{p}_i^{(j)} / \|\mathbf{p}_i^{(j)}\|$. From the definition of $\alpha_i^{(j)}$ in (16), the following holds,

$$\begin{cases} \hat{\boldsymbol{h}}_i^{(j)} \cdot \hat{\mathbf{p}}_i^{(j)} = \cos\alpha_i^{(j)} \\ \hat{\mathbf{l}}_i^{(j)} \cdot \hat{\mathbf{p}}_i^{(j)} = \cos\alpha_i^{(j)} \\ \hat{\boldsymbol{h}}_i^{(j)} \cdot \hat{\mathbf{l}}_i^{(j)} = \cos 2\alpha_i^{(j)} \\ \sin\alpha_i^{(j)} = \frac{r_i}{\|\mathbf{p}_i^{(j)}\|} \end{cases}, \tag{27}$$

where $(\cdot)$ denotes the dot product. Then, taking the square of both sides of (26), which are both positive quantities,

$$(c_1 \hat{\boldsymbol{h}}_i^{(j)} + c_2 \hat{\mathbf{l}}_i^{(j)} - \hat{\mathbf{p}})^T (c_1 \hat{\boldsymbol{h}}_i^{(j)} + c_2 \hat{\mathbf{l}}_i^{(j)} - \hat{\mathbf{p}}) \leq \sin^2\alpha_i^{(j)}, \tag{28}$$

and substituting (27) in (28), (28) can be simplified to

$$c_1^2 + c_2^2 + 2c_1 c_2 \cos 2\alpha_i^{(j)} - 2(c_1 + c_2)\cos\alpha_i^{(j)} + \cos^2\alpha_i^{(j)} \leq 0. \tag{29}$$

Because $|\cos 2\alpha_i^{(j)}| \leq 1$, the following inequality holds for any two nonnegative constants $c_1$ and $c_2$:

$$\begin{aligned} c_1^2 + c_2^2 + 2c_1 c_2 \cos 2\alpha_i^{(j)} &\geq c_1^2 + c_2^2 - 2|c_1 c_2| \\ &= (|c_1| - |c_2|)^2. \end{aligned} \tag{30}$$

Substituting (30) in (29), (29) can be rewritten as,

$$(|c_1| - |c_2|)^2 - 2c_1 \cos\alpha_i^{(j)} - 2c_2 \cos\alpha_i^{(j)} + \cos^2\alpha_i^{(j)} \leq 0, \tag{31}$$

where, to satisfy (31) $c_1$ and $c_2$ can not be both less then or equal to zero. We now show that $c_1$ and $c_2$ must have the same sign, by rewriting (29) such that

$$(c_1 + c_2 - \cos\alpha_i^{(j)})^2 - 4c_1 c_2 \sin^2\alpha_i^{(j)} \leq 0. \tag{32}$$

Then, it follows from (31) and (32) that both $c_1$ and $c_2$ are strictly positive. Since $\|\mathbf{p}_i^{(j)}\| \geq 0$, $c_1\|\mathbf{p}_i^{(j)}\|$ and $c_2\|\mathbf{p}_i^{(j)}\|$ are both greater than or equal to zero. Therefore, letting $c_1' \equiv c_1\|\mathbf{p}_i^{(j)}\|$ and $c_2' \equiv c_2\|\mathbf{p}_i^{(j)}\|$, the projection of any Markov track $\mathbf{m}_j \in K$ onto the heading plane (25) can be written as

$$\frac{(t - t_j)}{u_z}[u_x \ u_y]^T = c_1' \hat{\boldsymbol{h}}_i^{(j)} + c_2' \hat{\mathbf{l}}_i^{(j)}, \quad c_1', c_2' \geq 0. \tag{33}$$

Comparing (33) with the definition of the heading cone $K_\theta$ in (16), it can be seen that the projection (33) of any Markov track in $K$ must also lie in the heading cone $K_\theta$. Since (33) holds for any $\mathbf{m}_j \in K$, it follows that $K_\theta$ contains the projections of all Markov tracks in $K$ onto the heading plane (25).

We now prove that any Markov track $\mathbf{m}_j \in K$ must also lie in the velocity cone, $K_v$. In this case, the unit vector $\hat{\mathbf{u}}$ is expressed in spherical coordinates, i.e. $\hat{\mathbf{u}} = (1, \beta_i^{(j)}(t), \theta_j)$, where $\beta_i^{(j)}$ is the angle that $\hat{\mathbf{u}}$ makes with the $z$-axis, and $\theta_j$ is the azimuth angle that the projection of $\hat{\mathbf{u}}$ onto the heading plane (25) makes with the $x$-axis. Using the conversion from Cartesian to spherical coordinates, $\hat{\mathbf{u}}$ can be written as

$$[u_x \ u_y \ u_z]^T = [\cos\theta_j \sin\beta_i^{(j)} \ \sin\theta_j \sin\beta_i^{(j)} \ \cos\beta_i^{(j)}]^T, \tag{34}$$

and the coverage cone equation (6) can be written in spherical coordinates, as follows

$$\|(t - t_j)\tan\beta_i^{(j)}[\cos\theta_j \ \sin\theta_j]^T - (\mathbf{s}_i - \mathbf{x}_j)\| \leq r_i. \tag{35}$$

Since both sides of (35) are nonnegative quantities they can be squared and re-arranged to obtain the inequality

$$\begin{aligned} &(t - t_j)^2 \tan^2\beta_i^{(j)} + \|\mathbf{s}_i - \mathbf{x}_j\|^2 - r_i^2 \\ &\quad - 2(t - t_j)[\cos\theta_j \ \sin\theta_j](\mathbf{s}_i - \mathbf{x}_j)\tan\beta_i^{(j)} \leq 0. \end{aligned} \tag{36}$$

From (36), upper and lower bounds for the angle $\beta_i^{(j)}$ can be obtained as follows,

$$\begin{aligned} &\tan^{-1}\Bigg[\frac{1}{(t - t_j)}\Bigg([\cos\theta_j \ \sin\theta_j](\mathbf{s}_i - \mathbf{x}_j) \\ &-\sqrt{r_i^2 - ([\sin\theta_j - \cos\theta_j](\mathbf{s}_i - \mathbf{x}_j))^2}\Bigg)\Bigg] \leq \beta_i^{(j)}(t) \\ &\leq \tan^{-1}\Bigg[\frac{1}{(t - t_j)}\Bigg([\cos\theta_j \ \sin\theta_j](\mathbf{s}_i - \mathbf{x}_j) \\ &+\sqrt{r_i^2 - ([\sin\theta_j - \cos\theta_j](\mathbf{s}_i - \mathbf{x}_j))^2}\Bigg)\Bigg], \end{aligned} \tag{37}$$

and compared to the tangents of $\eta_i^{(j)}$ and $\mu_i^{(j)}$, defined in (21), to show that,

$$\tan\eta_i^{(j)} \leq \tan\beta_i^{(j)} \leq \tan\mu_i^{(j)}, \tag{38}$$

where, from (21), $\tan\eta_i^{(j)} \leq \tan\mu_i^{(j)}$.

Since $\hat{\mathbf{u}}$ lies in the velocity plane, there exist two constants $c_1, c_2 \in \mathbb{R}$, such that

$$[u_x \ u_y \ u_z]^T = c_1(t - t_j)\hat{\boldsymbol{\xi}}_i^{(j)}(t) + c_2(t - t_j)\hat{\boldsymbol{\omega}}_i^{(j)}(t). \tag{39}$$

From (22), if $c_1$ and $c_2$ are nonnegative, then $\mathbf{m}_j \in K_v$. Substituting (19) and (20) in (39), and dividing both sides by $\cos\theta_j$, it follows that

$$\begin{cases} c_1 \sin\eta_i^{(j)} + c_2 \sin\mu_i^{(j)} = \sin\beta_i^{(j)} \\ c_1 \cos\eta_i^{(j)} + c_2 \cos\mu_i^{(j)} = \cos\beta_i^{(j)}, \end{cases} \tag{40}$$

because $\cos\beta_i^{(j)}(t) \geq 0$ for $t \geq t_j$. The tangent of $\beta_i^{(j)}$ can be obtained as follows,

$$\tan\beta_i^{(j)} = \frac{c_1 \sin\eta_i^{(j)} + c_2 \sin\mu_i^{(j)}}{c_1 \cos\eta_i^{(j)} + c_2 \cos\mu_i^{(j)}}, \tag{41}$$

and substituted into (38), such that the two inequalities in (38) can be re-written as,

$$\tan \eta_i^{(j)} \leq \frac{c_1 \sin \eta_i^{(j)} + c_2 \sin \mu_i^{(j)}}{c_1 \cos \eta_i^{(j)} + c_2 \cos \mu_i^{(j)}}, \tag{42}$$

and

$$\tan \mu_i^{(j)} \geq \frac{c_1 \sin \eta_i^{(j)} + c_2 \sin \mu_i^{(j)}}{c_1 \cos \eta_i^{(j)} + c_2 \cos \mu_i^{(j)}}, \tag{43}$$

thereby eliminating $\beta_i^{(j)}$.

From (21), $\cos \eta_i^{(j)} \geq 0$ and $\cos \mu_i^{(j)} \geq 0$ because $t \geq t_j$. Also, substituting (41), (42), and (43) in (38), the inequalities,

$$c_2 \tan \eta_i^{(j)} \leq c_2 \tan \mu_i^{(j)}, \tag{44}$$

and,

$$c_1 \tan \eta_i^{(j)} \leq c_1 \tan \mu_i^{(j)}, \tag{45}$$

are obtained. Since by assumption $\tan \eta_i^{(j)} \leq \tan \mu_i^{(j)}$, $c_1$ and $c_2$ must be greater than or equal to zero for (44) and (45) to hold simultaneously. Moreover, since $(t - t_j) \geq 0$, it follows that $(t - t_j)c_1 \geq 0$ and $(t - t_j)c_2 \geq 0$. Thus, any Markov track $\mathbf{m}_j \in K$ can be written as,

$$\mathbf{m}_j(t) = c_1' \hat{\boldsymbol{\xi}}_i^{(j)}(t) + c_2' \hat{\boldsymbol{\omega}}_i^{(j)}(t), \quad c_1', c_2' \geq 0, \tag{46}$$

and, thus, from the definition of the velocity cone (22), it follows that $\mathbf{m}_j \in K_v$.

Since the above proof holds for any Markov track with target heading $\theta_j$ that obeys (18), it also follows that $K_v$ contains all Markov tracks in $K$ with this property. In other words, any Markov track in the coverage cone also lies in a velocity cone, such that its projection onto the heading plane simultaneously lies in the corresponding heading cone. As a result, $K$ can be represented by the pair of 2D cones $\{K_\theta, K_v\}$. □

**Example.** Consider the target trajectory plotted in Fig. 5, obtained by sampling the Markov motion model described in Section 8. A sensor located at $\mathbf{s}_i = [66.03 \;\; 85.00]^T$ (m), with range $r_i = 10$ (m), detects the target at a time $t = 52$ (s) with $t \in (t_5, t_6]$. Since $\mathbf{x}_5 = [62.16 \;\; 48.14]^T$, and $\mathbf{z}_5 = [62.16 \;\; 48.14 \;\; 45]^T$, it can be seen from Fig. 5 that the target Markov track, $\mathbf{m}_5$, is contained by $K$, and that $K$ can be represented by the corresponding velocity cone $K_v$ and heading cone $K_\theta$, also plotted in Fig. 5.

From Remark 4.1, the heading cone and the velocity cone contain all Markov tracks with origin $\mathbf{z}_j \in \Omega$ that are detected by sensor $i$ at any time $t \in (t_j, t_{j+1}]$. For any 2D cone with origin $\mathbf{z}_j$, the opening angle defines a Lebesgue measure on the set of line transversals through $\mathbf{z}_j$ [5]. Therefore, $K_v$ and $K_\theta$ can be used to define a Lebesgue measure on the set of tracks detected as a function of the sensor position $\mathbf{s}_i$, and range $r_i$. Furthermore, a cone opening angle can be computed from the cross product of the two unit vectors from which the cone

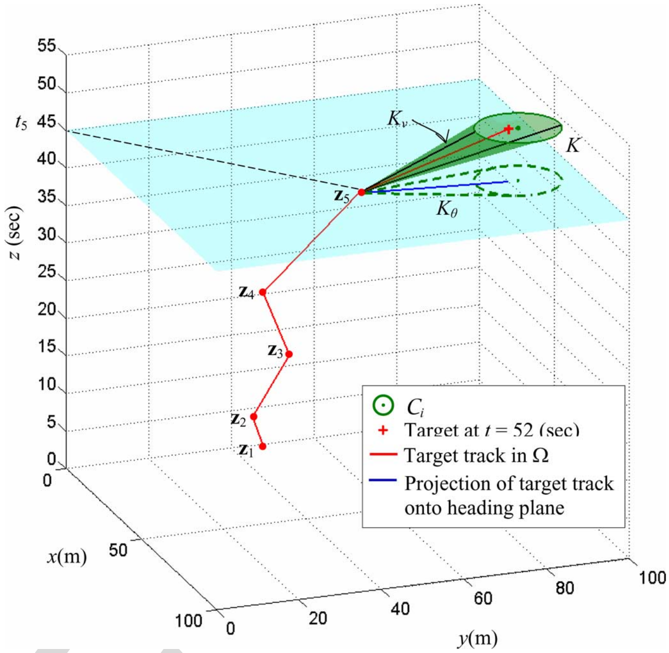


Fig. 5. Example of track detected at $t = 52$ (s), and contained by the coverage cone $K$ (green) or, equivalently, by $K_\theta$ (dashed green line) and $K_v$ (black).

is finitely generated, as follows. The opening angle of the heading cone $K_\theta$, finitely generated by $\hat{\boldsymbol{h}}_i^{(j)}$ and $\hat{\boldsymbol{l}}_i^{(j)}$, is

$$\begin{aligned}\psi_i^{(j)} &= \sin^{-1} \|\hat{\boldsymbol{l}}_i^{(j)} \times \hat{\boldsymbol{h}}_i^{(j)}\| \\ &= H(\det[\hat{\boldsymbol{l}}_i^{(j)} \quad \hat{\boldsymbol{h}}_i^{(j)}]^T) \sin^{-1}(\det[\hat{\boldsymbol{l}}_i^{(j)} \quad \hat{\boldsymbol{h}}_i^{(j)}]^T),\end{aligned} \tag{47}$$

where, $H(\cdot)$ denotes the Heaviside function, and $\det(\cdot)$ denotes the matrix determinant.

Similarly, the opening angle of the velocity cone $K_v$, finitely generated by $\hat{\boldsymbol{\omega}}_i^{(j)}$ and $\hat{\boldsymbol{\xi}}_i^{(j)}$, is

$$\begin{aligned}\phi_i^{(j)} &= \sin^{-1} \|\hat{\boldsymbol{\omega}}_i^{(j)} \times \hat{\boldsymbol{\xi}}_i^{(j)}\| \\ &= H(\det[\hat{\boldsymbol{\omega}}_i^{(j)} \quad \hat{\boldsymbol{\xi}}_i^{(j)}]^T) \sin^{-1}(\det[\hat{\boldsymbol{\omega}}_i^{(j)} \quad \hat{\boldsymbol{\xi}}_i^{(j)}]^T),\end{aligned} \tag{48}$$

where, the unit vectors $\hat{\boldsymbol{\omega}}_i^{(j)}$ and $\hat{\boldsymbol{\xi}}_i^{(j)}$ are a function of time and, thus, so is the opening angle of $K_v$. The Heaviside function $H(\cdot)$ in (47) and (48) ensures that if $\hat{\boldsymbol{l}}_i^{(j)} \succ \hat{\boldsymbol{h}}_i^{(j)}$, or $\hat{\boldsymbol{\omega}}_i^{(j)}(t) \succ \hat{\boldsymbol{\xi}}_i^{(j)}(t)$, the corresponding opening angles are equal to zero, indicating that $K_\theta = \emptyset$ or $K_v = \emptyset$, respectively. In the next section, the heading and velocity cones and their properties, are used to derive the probability of track detection in closed-form.

## 5 Probability of Detection

The Lebesgue measures (47) and (48) can be used to quantify the tracks detected by a sensor $i$, during a time interval $(t_j, t_{j+1}]$, when the PDFs of the Markov parameters are uniform. This case typically comes about when no prior target information is available, and all target tracks are equally probable. In many target-tracking applications, however, the PDFs of the Markov parameters are not uniform, and are computed from sensor measurements via Kalman filtering.

This section builds on results presented in Sections 3–4 to derive the probability of track detection as function of the sensor position, range, and the PDFs of the Markov parameters.

From the definition of joint PDF [22] and the Markov property of the target motion model, the probability that the values of the Markov parameters during the $j$th time interval $(t_j, t_{j+1}]$ fall in a subset $B \subset \{\mathcal{A} \times \mathcal{H} \times \mathcal{V}\}$ is,

$$\Pr(\{\mathbf{x}_j, \theta_j, v_j\} \in B) = \int_B f^{(j)}_{X,\Theta,V}(\mathbf{x}_j, \theta_j, v_j) d\mathbf{x}_j d\theta_j dv_j, \tag{49}$$

where $f_{X,\Theta,V}$ denotes the joint PDF of $\mathbf{x}_j$, $\theta_j$, and $v_j$. Based on the assumptions in Section 2, the parameters at the $j$th time interval are independent and, thus, the joint PDF can be factorized such that,

$$\Pr(\{\mathbf{x}_j, \theta_j, v_j\} \in B) = \int_B f^{(j)}_X(\mathbf{x}_j) f^{(j)}_\Theta(\theta_j) f^{(j)}_V(v_j) d\mathbf{x}_j d\theta_j dv_j, \tag{50}$$

where all of the PDFs in (50) can be obtained from (4) and a target-tracking algorithm.

From the results in Section 4, the heading cone and the velocity cone contain the Markov parameters' values that cause a detection by sensor $i$, during the $j$th time interval. Let the binary random variable $D_i(t)$ represent the $i$th sensor detection at time $t$, where event $\{D_i(t) = 1\}$ represents a successful detection, and event $\{D_i(t) = 0\}$ represents a failed detection. Then, the probability of detection at a time $t \in (t_j, t_{j+1}]$ is the probability that $\mathbf{m}_j(t) \in K(t)$. Based on the 2D-cone representation of $K$ (Section 4), the unit vectors that finitely generate the heading cone and the velocity cone can be used to define lower and upper bounds for the Markov parameters that correspond to $\{D_i(t) = 1\}$. In particular, from Remark 4, $\mathbf{m}_j \in K$ if and only if $\gamma^{(j)}_i \leq \theta_j \leq \lambda^{(j)}_i$ and $\tan \eta^{(j)}_i \leq v_j \leq \tan \mu^{(j)}_i$, where $\lambda^{(j)}_i$ and $\gamma^{(j)}_i$ are defined in (13) and (14), and $\eta^{(j)}_i$ and $\mu^{(j)}_i$ are defined in (21). Thus, given the PDFs of the Markov parameters, the probability that $\mathbf{m}_j$ is contained in $K$ is

$$\begin{aligned} &\Pr[\mathbf{m}_j(t) \in K(t)] \equiv P_d(i,t) \\ &= \int_{\mathcal{A}} f^{(j)}_X(\mathbf{x}_j) \int_{\gamma^{(j)}_i}^{\lambda^{(j)}_i} f^{(j)}_\Theta(\theta_j) \int_{\tan \eta^{(j)}_i}^{\tan \mu^{(j)}_i} f^{(j)}_V(v_j) dv_j d\theta_j d\mathbf{x}_j. \end{aligned} \tag{51}$$

Then, the probability mass function (PMF) of the discrete random variable $D_i$ can be written in terms of (51), as shown by the following result.

**Theorem 5.1.** *The probability mass function (PMF) of a discrete random variable $D_i$ that represents the $i$th sensor detection at time $t$ is,*

$$p_{D_i}(d_i) = \begin{cases} P_d(i,t), & \text{if} \quad d_i(t) = 1 \\ 1 - P_d(i,t), & \text{if} \quad d_i(t) = 0, \end{cases} \tag{52}$$

*where, $d_i$ denotes any possible value of $D_i$ in the range $\mathcal{D} = \{0, 1\}$, and $P_d(i,t)$ is an integral function defined in terms of the Markov parameters, as shown in (51).*

**Proof.** Given the PDFs of the Markov parameters, $f_X$, $f_\Theta$, and $f_V$, and the independence assumptions in Section 2, the probability of event $\{D_i(t) = 1\}$ can be obtained through marginalization of the joint probability of $\mathbf{x}_j$, $\theta_j$, and $v_j$, [22], such that

$$\begin{aligned} &\Pr[D_i(t) = 1] \\ &= \Pr[v_j \in [\tan \eta^{(j)}_i, \tan \mu^{(j)}_i], \theta_j \in [\gamma^{(j)}_i, \lambda^{(j)}_i], \mathbf{x}_j \in \mathcal{A}] \\ &= \Pr[v_j \in [\tan \eta^{(j)}_i, \tan \mu^{(j)}_i] \mid \theta_j \in [\gamma^{(j)}_i, \lambda^{(j)}_i], \mathbf{x}_j \in \mathcal{A}] \\ &\quad \times \Pr[\theta_j \in [\gamma^{(j)}_i, \lambda^{(j)}_i], \mathbf{x}_j \in \mathcal{A}] \\ &= \Pr[v_j \in [\tan \eta^{(j)}_i, \tan \mu^{(j)}_i] \mid \theta_j \in [\gamma^{(j)}_i, \lambda^{(j)}_i], \mathbf{x}_j \in \mathcal{A}] \\ &\quad \times \Pr[\theta_j \in [\gamma^{(j)}_i, \lambda^{(j)}_i] \mid \mathbf{x}_j \in \mathcal{A}] \Pr[\mathbf{x}_j \in \mathcal{A}]. \end{aligned} \tag{53}$$

From the definitions of joint and conditional PDFs [22], the above probabilities can be expressed in terms of the PDFs of the Markov parameters as follows

$$\begin{aligned} &\Pr[D_i(t) = 1] \\ &= \int_{\mathcal{A}} \int_{\mathcal{H}} \int_{\mathcal{V}} f_V(v_j \mid \theta_j, \mathbf{x}_j) f_\Theta(\theta_j \mid \mathbf{x}_j) f_X(\mathbf{x}_j) dv_j d\theta_j d\mathbf{x}_j \\ &= \int_{\mathcal{A}} f^{(j)}_X(\mathbf{x}_j) \int_{\gamma^{(j)}_i}^{\lambda^{(j)}_i} f^{(j)}_\Theta(\theta_j) \int_{\tan \eta^{(j)}_i}^{\tan \mu^{(j)}_i} f^{(j)}_V(v_j) dv_j d\theta_j d\mathbf{x}_j \\ &= P_d(i,t). \end{aligned} \tag{54}$$

Then, from the normalization property, the probability of a failed detection at time $t \in (t_j, t_{j+1}]$ is,

$$\Pr[D_i(t) = 0] = [1 - P_d(i,t)], \tag{55}$$

and, thus, the PMF of $D_i$ is given by (52). □

Since the integral function $P_d(i,t)$ is provided solely as a function of the PDFs of the Markov parameters during the $j$th time interval, it holds for all $j = 1, \dots, m$. Then, the PMF of $D_i$ in (52) provides the probability of detection for sensor $i$, at any time $t \in (T_0, T_f]$, as formulated in Problem 2.1. Also, now that the PMF of $D_i$ is known from (52), it can be used to compute the *expected value* of $D_i$ at time $t \in (T_0, T_f]$,

$$\begin{aligned} \mathrm{E}[D_i(t)] &\equiv \sum_{d_i \in \mathcal{D}} d_i f_{D_i}(d_i) = (1) P_d(i,t) + (0)[1 - P_d(i,t)] \\ &= P_d(i,t), \end{aligned} \tag{56}$$

where $\mathrm{E}[\cdot]$ denotes the expectation [22].

## 6 Track Coverage

In many monitoring and surveillance applications, it may be relevant to optimize the number of detections obtained by the sensor network over a time interval $(T_0, T_f]$. Typically, detection events are defined in discrete time because the sensor requires a finite amount of time to process the target information and declare a detection. Also, by this approach, the detection events are countable and, thus, over a finite period of time, the sensor obtains a finite number of detections, even if the target is always in its FOV.

Let $\delta t$ denote the time required by a detection event, and assume $\delta t$ is a finite positive constant chosen such that $\delta t \ll \Delta t_j$ and $\Delta t_j$ is a multiple of $\delta t$, for any $j = 1, \ldots, m$. Then, the time interval $(T_0, T_f]$ can be discretized as follows,

$$\tau_k = T_0 + k\delta t, \quad k = 0, \ldots, N, \tag{57}$$

where, $N = (T_f - T_0)/\delta t$. With these assumptions, the two discrete-time indices, $t_j$ and $\tau_k$ can be reconciled because $(t_j - T_0)/\delta t$ is a positive integer for any $j = 1, \ldots, m$, and at the initial time $T_0 = t_1 = \tau_0$.

Then, the expected number of detections for a sensor $i$ during $(T_0, T_f]$ is,

$$\begin{aligned} J_i &= \mathrm{E}\left[\sum_{k=1}^{N} D_i(t = \tau_k)\delta t\right] = \sum_{k=1}^{N} \mathrm{E}[D_i(\tau_k)]\delta t \\ &= \delta t \sum_{j=1}^{m} \sum_{\tau_k \in (t_j, t_{j+1}]} P_d(i, \tau_k), \end{aligned} \tag{58}$$

where, $P_d$ is defined in (51), and $\delta t$ is a known constant. Also, the track coverage of $n$ sensors, as defined in Problem 2.2, can be determined as follows

$$\begin{aligned} J = \sum_{i=1}^{n} J_i = \delta t \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{\tau_k \in (t_j, t_{j+1}]} \\ \left\{ \int_{\mathcal{A}} f_X^{(j)}(\mathbf{x}_j) \int_{\gamma_i^{(j)}}^{\lambda_i^{(j)}} f_\Theta^{(j)}(\theta_j) \int_{\tan \eta_i^{(j)}}^{\tan \mu_i^{(j)}} f_V^{(j)}(v_j) dv_j d\theta_j d\mathbf{x}_j \right\}. \end{aligned} \tag{59}$$

For some PDFs, the track coverage function in (59) may not be readily solved analytically. In this case, the track coverage can be computed numerically, for example using the MATLAB® *int* function or the *quadl* function which uses recursive adaptive Lobatto quadrature [26]. When track coverage is to be optimized, an approximate closed-form representation may be obtained by approximating the integrals in (59) using Riemann sum. By this approach, the ranges of the Markov motion, $\mathcal{A}$, $\mathcal{H}$, and $\mathcal{V}$ are also discretized using constant increments $[\delta x_j \quad \delta y_j]^T$, $\delta\theta_j$, and $\delta v_j$, respectively. Since the PDFs of target heading and speed are smooth, the granularity of discretization does not affect the accuracy of the track coverage performance greatly. Then, the discrete Markov track origin $\mathbf{x}_j \in \mathcal{A}$ is indexed by,

$$\mathbf{x}_j^{(\imath,\jmath)} = [\imath\delta x_j \quad \jmath\delta y_j]^T, \imath = 0, \cdots, N_x, \jmath = 0, \cdots, N_y, \tag{60}$$

where, $N_x = L/\delta x_j$ and $N_y = L/\delta y_j$. The discrete target heading is indexed by,

$$\theta_j^{(l)} = \gamma_i^{(j)} + l\delta\theta_j, \quad l = 0, \cdots, N_\theta, \tag{61}$$

where, $N_\theta = (\lambda_i^{(j)} - \gamma_i^{(j)})/\delta\theta_j$, and the discrete velocity is indexed by,

$$v_j^{(\ell)} = \eta_i^{(j)} + \ell\delta v_j, \quad \ell = 0, \cdots, N_v, \tag{62}$$

where, $N_v = (\mu_i^{(j)} - \eta_i^{(j)})/\delta v_j$.

With the above discretization, the track coverage function (59) can be approximated as follows,

$$\begin{aligned} J \approx J_D = \delta t \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{\tau_k \in (t_j, t_{j+1}]} \sum_{\imath=0}^{N_x} \sum_{\jmath=0}^{N_y} f_X^{(j)}(\mathbf{x}_j^{(\imath,\jmath)}) \\ \times \sum_{l=0}^{N_\theta} f_\Theta^{(j)}(\theta_j^{(l)}) \sum_{\ell=0}^{N_v} f_V^{(j)}(v_j^{(\ell)}) \delta v_j \delta\theta_j \delta x_j \delta y_j, \end{aligned} \tag{63}$$

evaluating the PDFs of the Markov parameters at all discrete values that fall in the range specified by the heading and velocity cones. It can be seen that $J$ and $J_D$ both are functions of the sensor positions and ranges, because so are the angles $\lambda_i^{(j)}$, $\gamma_i^{(j)}$, $\mu_i^{(j)}$, and $\eta_i^{(j)}$ derived from the heading and velocity cones in (13), (14), and (21). Now that a track coverage function has been obtained in closed form it can be optimized with respect to the sensor placement, as shown in the following section.

## 7 Sensor Placement

The track coverage function derived in the previous section can be used to determine the placement of $n$ sensors in $\mathcal{A}$ such that the expected number of target detections is maximized during a time interval $(T_0, T_f]$ (Problem 2.3). The optimal sensor placement $\mathbf{S}^*$ can be computed using a nonlinear program (NLP) that optimizes the coverage function (63), subject to a set of equality and inequality constraints. These constraints can be used to specify additional requirements to be satisfied by the sensor deployment. For example, many surveillance systems require sensors to obtain multiple independent detections [12], [27]–[30]. In this case, the NLP may be used to avoid intersections between the sensor FOVs. Also, the sensor FOVs typically must be contained by the RoI in order to maximize area coverage [7].

Then, a sensor placement $\mathbf{S}^*$ that maximizes $J_D$, while satisfying the above requirements, can be obtained by solving the NLP,

$$\begin{aligned} &\text{maximize } J_D(\mathbf{S}), \\ &\text{subject to } (s_{x_i} - s_{x_l})^2 + (s_{y_l} - s_{y_l})^2 - (r_i + r_l)^2 > 0, \\ &\qquad\qquad\qquad\qquad i, l = 1, \ldots, n, i \neq l \\ &\qquad (s_{x_i} - r_i) > 0 \\ &\qquad (s_{y_i} - r_i) > 0 \\ &\qquad (L - s_{x_i} - r_i) > 0 \\ &\qquad (L - s_{y_i} - r_i) > 0, \quad i = 1, \ldots, n, \end{aligned} \tag{64}$$

in $\mathbf{S}$, using a sequential quadratic programming (SQP) algorithm [31], [32].

## 8 Simulation Results

In this section, the probability of detection $P_d$, derived in closed form in (51), and used to obtain the track coverage function (59), is first validated numerically using Monte Carlo simulations. Then, the sensor placement method presented in the previous section is demonstrated numerically and compared to greedy, grid, and random deployment algorithms.

TABLE 1
Markov Motion Model Probability Density Functions (PDFs)

| **Maneuvering Interval** $(t_j,\ t_{j+1}]$ (s) | **Heading PDF** $f_\Theta^{(j)}(\theta_j)$ | **Velocity PDF** $f_V^{(j)}(v_j)$ |
|---|---|---|
| (0, 5] (s) $(j=1)$ | $\mathcal{U}(\mathcal{H})$, $\mathcal{H}=[-\pi/4,\ -\pi/6]$ | $\mathcal{U}(\mathcal{V})$, $\mathcal{V}=[1,\ 5]$ (m/s) |
| (5, 15] (s) $(j=2)$ | $\mathcal{N}(\mu,\sigma)$, $\mu=\pi/4$, $\sigma=\pi/8$ | $\mathcal{U}(\mathcal{V})$, $\mathcal{V}=[1,\ 4]$ (m/s) |
| (15, 30] (s) $(j=3)$ | $\mathcal{N}(\mu,\sigma)$, $\mu=-\pi/4$, $\sigma=\pi/8$ | $\mathcal{N}(\mu,\sigma)$, $\mu=2$ (m/s), $\sigma=0.25$ (m/s) |
| (30, 45] (s) $(j=4)$ | $\mathcal{U}(\mathcal{H})$, $\mathcal{H}=[\pi/4,\ 3\pi/4]$ | $\mathcal{N}(\mu,\sigma)$, $\mu=2$ (m/s), $\sigma=0.25$ (m/s) |
| (45, 55] (s) $(j=5)$ | $\mathrm{Mult}_2(w_i;\mu_i,\sigma_i)$, $w_1=0.5$, $\mu_1=-\pi/4$, $\sigma_1=\pi/6$, $w_2=0.5$, $\mu_2=\pi/4$, $\sigma_2=\pi/6$ | $\mathcal{U}(\mathcal{V})$, $\mathcal{V}=[3,\ 5]$ (m/s) |

The simulations are performed using the Markov motion model described in Table 1, and are representative of all simulations conducted with other models and parameters.

The Markov motion model in Table 1 describes the target motion over a time interval $(T_0, T_f]$ with $T_0 = 0$ (s), and $T_f = 55$ (s), characterized by five maneuvering intervals, $(t_j, t_{j+1}]$, indexed by $j = 1, \ldots, m$, with $m = 5$. At the initial time $T_0 = t_1 = 0$ (s), the PDF of the initial target position, $\mathbf{x}_1$, is a two-dimensional multivariate Gaussian PDF,

$$f_X^{(1)}(\mathbf{x}_1) = \frac{1}{(2\pi)^{n/2}\det(\boldsymbol{\Sigma})^{1/2}} e^{[-(1/2)(\mathbf{x}_1-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_1-\boldsymbol{\mu})]}, \tag{65}$$

with a mean vector $\boldsymbol{\mu} = [0 \quad 50]^T$ and a diagonal covariance matrix $\boldsymbol{\Sigma} = \mathrm{diag}([20 \quad 20])$. Where, $\det(\cdot)$ denotes the matrix determinant, $(\cdot)^{-1}$ denotes the matrix inverse, and $\mathrm{diag}(\cdot)$ denotes an operator that places a row vector on the diagonal of a zero matrix.

The PDFs of the heading and velocity parameters for every maneuvering interval are described in Table 1. Gaussian PDFs are denoted by $\mathcal{N}(\mu, \sigma)$, where $\mu$ is the mean, and $\sigma$ is the standard deviation. Uniform PDFs are denoted by $\mathcal{U}(a, b)$, where $[a, b]$ is the range of the random variable (also known as support of the distribution). PDFs modeled by mixtures of $g$-Gaussian components are denoted by $\mathrm{Mult}_g(w_i; \mu_i, \sigma_i)$, where $w_i$, $\mu_i$, and $\sigma_i$ are the weight, mean, and standard deviation of the $i$th component, respectively. Then, the PDF of the heading angle at $j = 5$ is,

$$f_\Theta^{(5)}(\theta_5) = \sum_{i=1}^{g} w_i \frac{1}{\sigma_i\sqrt{2\pi}} e^{[-(1/2)(\theta_5-\mu_i)/\sigma_i]}, \tag{66}$$

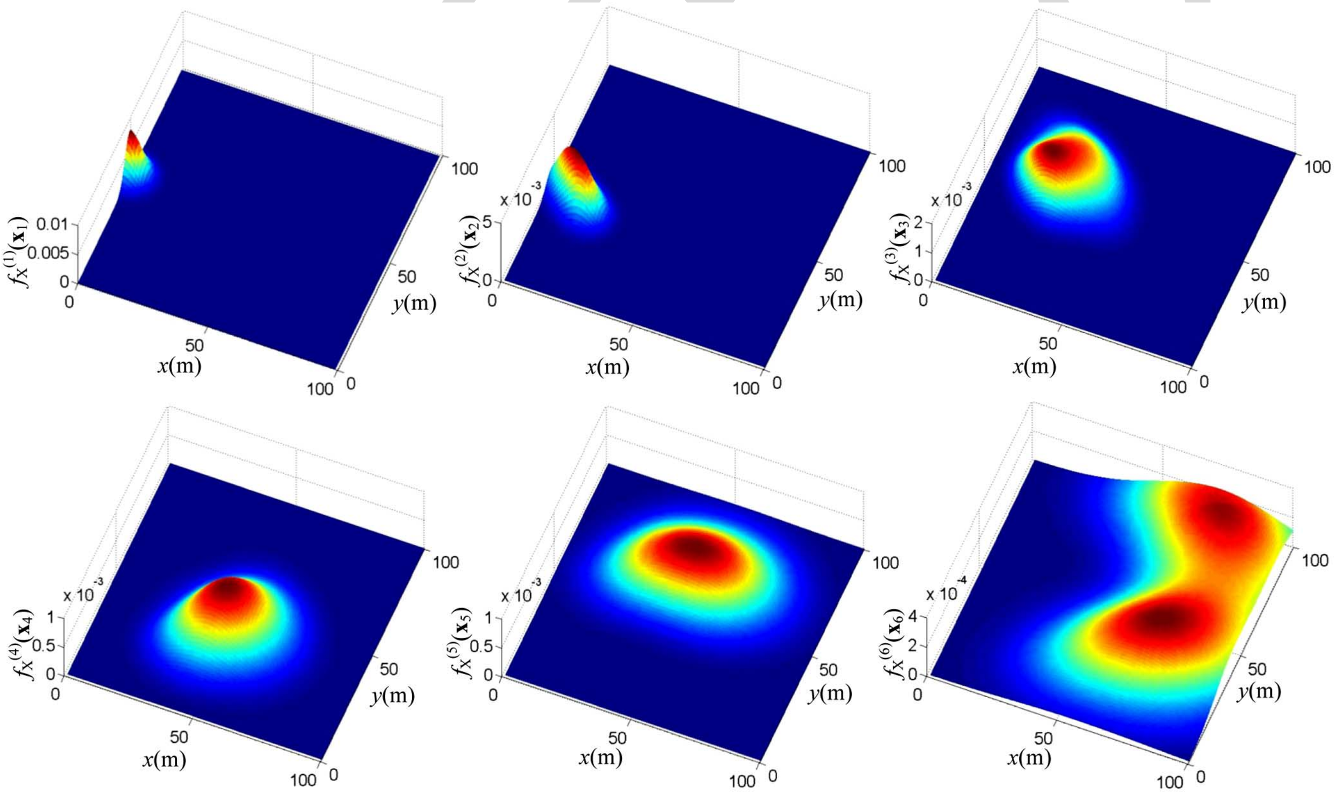


Fig. 6. PDFs of initial target positions derived from (4).

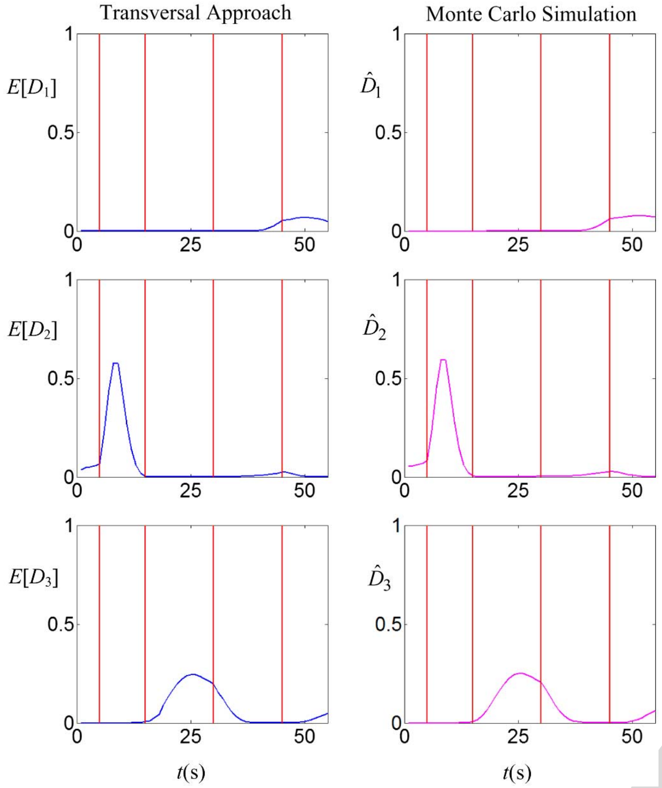


Fig. 7. Probability of detection for $n = 3$ sensors computed by the integral function (51) (blue), and by MC simulation (magenta).

where, $0 \leq w_i \leq 1$ for all $i = 1, \dots, g$, and $\sum_{i=1}^{g} w_i = 1$ [33]. Once the PDFs of the headings and velocities are specified, the PDFs of the initial positions $\mathbf{x}_j$ are computed from (4). For the Markov motion model in Table 1, the PDFs of $\mathbf{x}_j$ are computed from $f_{\Theta}^{(j)}$ and $f_V^{(j)}$ for $j = 2, \dots, 6$, and then plotted in Fig. 6. These results, obtained from (4), were also validated numerically using Monte Carlo simulations (plots omitted for brevity).

### 8.1 Probability of Detection Simulation Results

The probability of detection (51) is validated numerically by considering $n = 3$ sensors at known positions. The detection probability is first evaluated by integrating (51) over time using MATLAB® *dblquad* and *quad* functions. Then, the results are plotted in Fig. 7, and compared to the detection probability obtained from a Monte Carlo (MC) simulation. Monte Carlo simulations are a computationally intensive but useful approach for approximating the evolution of a dynamical systems involving random variables with known PDFs [34]. A statistically significant number of trials or, in this case, target tracks, denoted by $N_{MC}$, are drawn by sampling the PDFs of the Markov parameters described in Table 1. A 3D Boolean array $B = \{b_{ijk}\}$ is used to store detection outcomes, such that when a target track is sampled, the event $\{D_i(\tau_k) = 1\}$ is stored by letting element $b_{ijk} = 1$ for any time $\tau_k \in (t_j, t_{j+1}]$ at which $\mathbf{x}(\tau_k) \in \mathcal{C}_i$ during the simulation. For any time $\tau_k \in (t_j, t_{j+1}]$ at which $\mathbf{x}(\tau_k) \notin \mathcal{C}_i$, the event $\{D_i(\tau_k) = 0\}$ is stored by letting element $b_{ijk} = 0$.

In the MC simulation, the expected value of $D_i$ at time $\tau_k$ is computed by summing the detection events for the entire target track, and dividing by the total number of trials, i.e.:

$$\hat{D}_i(\tau_k) = \frac{1}{N_{MC}} \sum_{j=1}^{m} b_{ijk}. \tag{67}$$

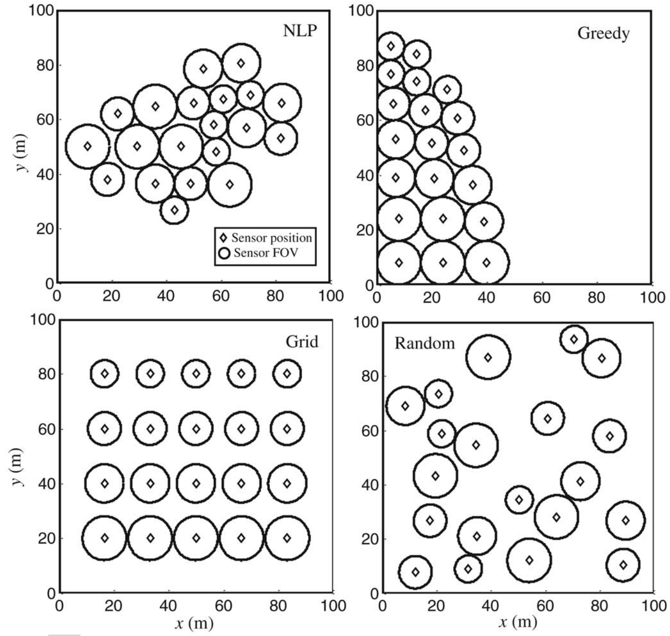


Fig. 8. Optimal, greedy, grid, and random sensor placements for $n = 20$, and the target model in Table 1.

TABLE 2
Size and Ranges of Simulated Sensor Networks

| Size, $n$ | Ranges, $\{r_1, \dots, r_n\}$ |
|---|---|
| 2 | {8, 8} |
| 3 | {8, 8, 8} |
| 4 | {8, 8, 8, 8 } |
| 5 | {8, 8, 8, 8, 8} |
| 6 | {8, 8, 8, 8, 8, 7} |
| 7 | {8, 8, 8, 8, 8, 7, 7} |
| 8 | {8, 8, 8, 8, 8, 7, 7, 7} |
| 9 | {8, 8, 8, 8, 8, 7, 7, 7, 7} |
| 10 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7} |
| 11 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6} |
| 12 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6, 6} |
| 13 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6, 6, 6} |
| 14 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6, 6, 6, 6} |
| 15 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6} |
| 16 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6, 5} |
| 17 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6, 5, 5} |
| 18 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6, 5, 5, 5} |
| 19 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6, 5, 5, 5, 5} |
| 20 | {8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6, 5, 5, 5, 5, 5} |

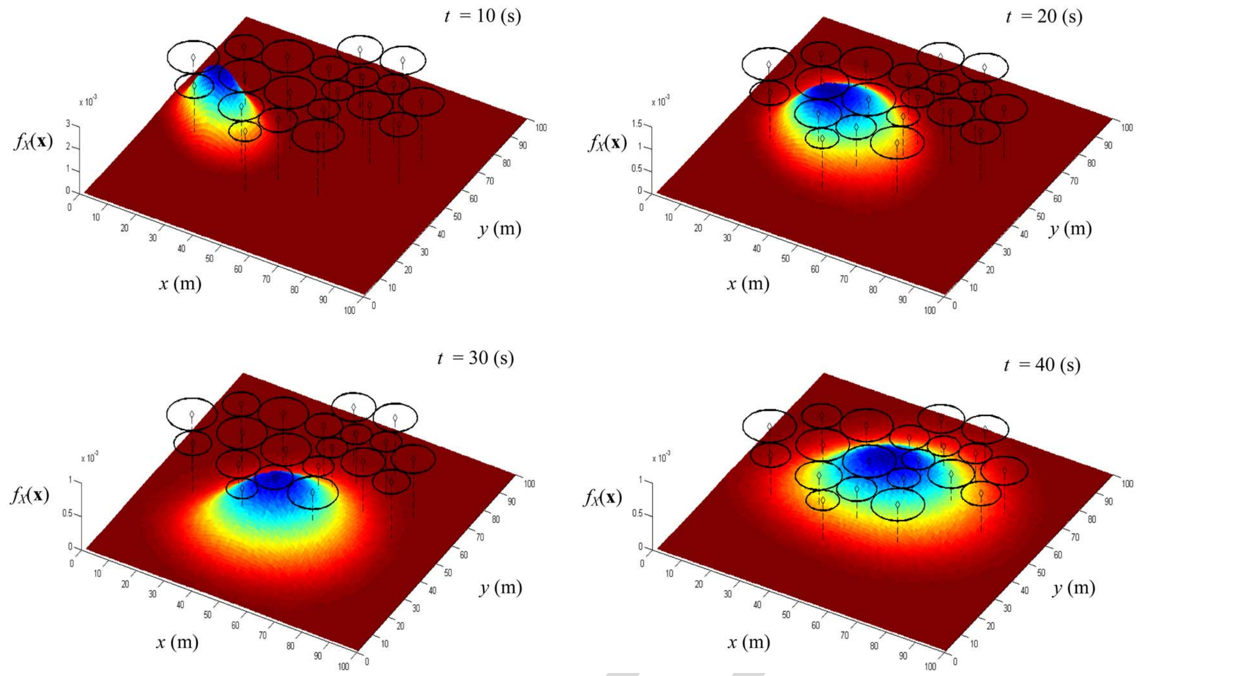


Fig. 9. PDF of target position at four instants in time, and optimal placement obtained by NLP for $n = 20$ sensors.

From (56) the expected value of $D_i$ is equal to the probability of detection $P_d$. Therefore, (67) can be used to validate the integral function $P_d$ in (51). The results are plotted for comparison in Fig. 7, where it can be seen that the MC simulations validate the probability of detection $P_d$ in (51), derived using a geometric transversals approach.

### 8.2 Sensor Placement Optimization Results

The track coverage function $J_D$, derived in closed form in (63), represents the expected number of detections during a time interval $(T_0, T_f]$, as a function of the sensor placement $\mathbf{S}$. Using the approach described in Section 7, $J_D$ can be optimized to obtain a sensor placement $\mathbf{S}^*$ that maximizes the network's ability to track a target based on its Markov motion model (Problem 2.3). The sensor placement $\mathbf{S}^*$ is obtained by solving the NLP in (64) using an SQP algorithm implemented by the MATLAB® Optimization Toolbox *fmincon* function, described in [35]. The MATLAB® SQP algorithm was found to outperform other NLP software packages, such as [36], while also allowing for an easier implementation. In every case, multiple random initializations are used to avoid local maxima by ultimately picking the solution $\mathbf{S}^*$ with the highest value of $J_D$.

An example of sensor placement $\mathbf{S}^*$ obtained from the NLP solution is shown in Fig. 8 for a network with the ranges shown in Table 2, for $n = 20$. The target motion model in Table 1 is then used to simulate the PDF of the target position, $\mathbf{x}$, over time, and to plot it at four instants in time in Fig. 9. Superimposing a plot of $\mathbf{S}^*$ onto the PDFs in Fig. 9, it can be seen that sensors are placed in regions where the probability of detection is high throughout the time interval $(T_0, T_f]$.

The effectiveness of the proposed sensor placement approach is compared to that of three existing sensor deployment methods known as greedy algorithm [16], grid algorithm [17], and randomized algorithm [19]. The greedy algorithm places the sensors by packing unequal circles into a 2D rectangular container according to the maximum hole degree rule [16]. Given the same sensor network size and ranges used in the NLP example, the greedy algorithm produces the sensor placement plotted in Fig. 8. The grid algorithm uses the approach presented in [17] to place sensors on a grid that takes into account the sensor ranges and the dimensions of the RoI, as shown in Fig. 8. The randomized algorithm, inspired by the approach presented in [19], places sensors randomly in the RoI, preventing intersections between FOVs, or with the RoI boundaries, in order to maximize area coverage and obtain independent detections, as shown in Fig. 8.

The performance of the sensor placements obtained by the four algorithms are compared in Fig. 10, using nineteen sensor

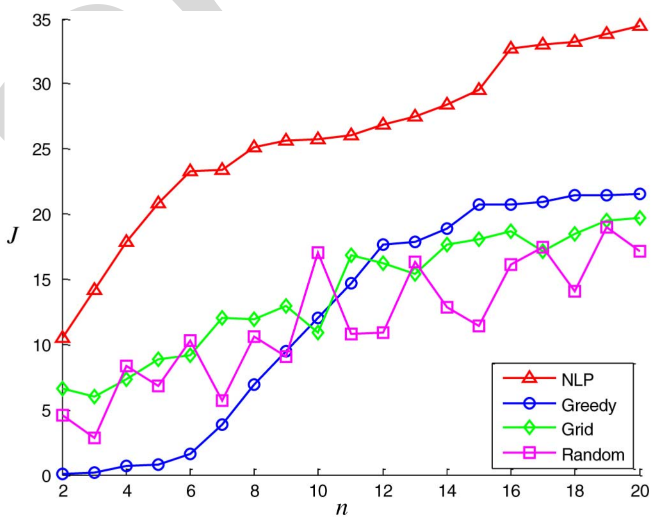


Fig. 10. Track coverage of sensor placements obtained by NLP, greedy, grid, and randomized algorithms for the networks in Table 2.

networks with the characteristics shown in Table 2. For each sensor placement, the track coverage function $J$, in (59), is evaluated using MATLAB® *dblquad* and *quad* functions. The results, plotted in Fig. 10, show that the sensor placements obtained by NLP significantly outperform all others in the expected number of target track detections for the Markov model in Table 1.

## 9 Summary and Conclusions

There is considerable precedence in the sensor tracking and estimation literature for modeling maneuvering targets by Markov motion models in order to estimate the target state from multiple, distributed sensor measurements. Although the transition probability density functions of these Markov models are routinely outputted by tracking and estimation algorithms, little work has been done to use them as a feedback to sensor coordination and control algorithms. Geometric transversals and convex theory have been previously utilized to derive track coverage functions for deploying and controlling sensor networks such that their ability to track targets traversing the RoI at constant speed and heading is maximized.

This paper extends this theory to maneuvering targets described by Markov motion models, by analyzing the geometric properties of track detections in a spatio-temporal Euclidian space. Using this novel approach, the probability of track detection can be derived in closed-form for a time-varying problem formulation, in which the Markov transition probabilities are not necessarily uniform. The concept of three-dimensional spatio-temporal coverage cone is introduced, along with its two-dimensional representations referred to as heading and velocity cones, which may be utilized to define a Lebesgue measure of track coverage for omnidirectional sensors. The probability of track detection derived by the geometric transversals approach is validated numerically through Monte Carlo simulations. Then, a related track coverage function is utilized to formulate the optimal sensor deployment problem as an NLP. The numerical results presented in this paper show that sensor deployments obtained from the NLP significantly outperform deployments obtained by existing greedy, grid, and randomized algorithms.

## Acknowledgment

This work was supported by the National Science Foundation, under Grant NSF ECCS #1028506.

## References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.
[2] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proc. IEEE Wireless Commun. Netw. (WCNC'03)*, 2003, pp. 1609–1614.
[3] S. Ferrari, "Track coverage in sensor networks," in *Proc. Amer. Control Conf. (ACC'06)*, 2006, pp. 2053–2059.
[4] 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 J. Control Optim.*, vol. 48, no. 1, pp. 292–320, 2009.
[5] K. Baumgartner and S. Ferrari, "A geometric transversal approach to analyzing track coverage in sensor networks," *IEEE Trans. Comput.*, vol. 57, no. 8, pp. 1113–1128, Aug. 2008.
[6] K. Baumgartner, S. Ferrari, and T. Wettergren, "Robust deployment of dynamic sensor networks for cooperative track detection," *IEEE Sensors J.*, vol. 9, no. 9, pp. 1029–1048, 2009.
[7] K. Baumgartner, S. Ferrari, and A. Rao, "Optimal control of an underwater sensor network for cooperative target tracking," *IEEE J. Ocean. Eng.*, vol. 34, no. 4, pp. 678–697, Oct. 2009.
[8] M. Ranasingha, M. Murthi, K. Premaratne, and X. Fan, "Transmission rate allocation in multisensor target tracking over a shared network," *IEEE Trans. Syst. Man Cybern. B*, vol. 39, no. 2, pp. 348–362, 2009.
[9] S. Ferrari, C. Cai, R. Fierro, and B. Perteet, "A multi-objective optimization approach to detecting and tracking dynamic targets in pursuit-evasion games," in *Proc. 2007 Amer. Control Conf.*, 2007, pp. 5316–5321.
[10] B. Perteet, J. McClintock, and R. Fierro, "A multi-vehicle framework for the development of robotic games: The Marco Polo case," in *Proc. IEEE Int. Conf. Robot. Autom.*, April 10–14, 2007, pp. 3717–3722.
[11] T. A. Wettergren, "Performance of search via track-before-detect for distributed sensor networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 314–325, Jan. 2008.
[12] J.-P. Le Cadre and G. Souris, "Searching tracks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 4, pp. 1149–1166, Oct. 2000.
[13] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*. Hoboken, NJ, USA: Wiley, 2001.
[14] A. Howard, M. J. Matari, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Auton. Robots*, vol. 13, pp. 113–126, 2002.
[15] V. Isler, S. Khanna, and K. Daniilidis, "Sampling based sensor-network deployment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 100, pp. 1780–1785, 2004.
[16] H. Chi-Fu and T. Yu-Chee, "The coverage problem in a wireless sensor network," *Mobile Netw. Appl.*, vol. 10, no. 4, pp. 519–528, 2005.
[17] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1448–1453, Dec. 2002.
[18] J. A. George, J. M. George, and B. Lamer, "Packing different-sized circles into a rectangular container," *Eur. J. Opl. Res.*, vol. 84, pp. 693–712, 1995.
[19] H. González-Banos, "A randomized art-gallery algorithm for sensor placement," in *Proc. 17th Annu. Symp. Comput. Geom. (SCG'01)*, 2001, pp. 232–240.
[20] Y. Bar-Shalom and X. Li, *Multitarget–Multisensor Tracking: Principles and Techniques*. Storrs, CT, USA: YBS Publishing, 1995.
[21] Y. Bar-Shalom and E. W. D. Blair, *Multitarget-Multisensor Tracking: Applications and Advances*, vol. 3. Norwood, MA, USA: Artech House, 2000.
[22] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*. Belmont, MA, USA: Athena Scientific, 2008.
[23] S. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[24] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proc. 1st ACM Int. Workshop Wireless Sensor Netw. Appl. (WSNA'02)*. New York, NY, USA: ACM, 2002, pp. 32–41.
[25] D. P. Bertsekas, *Convex Analysis and Optimization*. Belmont, MA, USA: Athena Scientific, 2003.
[26] Mathworks. (2004). *MATLAB* [Online]. Available: http://www.mathworks.com.
[27] T. A. Wettergren, "Statistical analysis of detection performance for large distributed sensor systems," Naval Undersea Warfare Center, Newport, RI, Tech. Rep. ADA417136, June 2003.
[28] T. A. Wettergren, R. L. Streit, and J. R. Short, "Tracking with distributed sets of proximity sensors using geometric invariants," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 4, pp. 1366–1374, Oct. 2004.
[29] B. Koopman, *Search and Screening: General Principles with Historical Applications*. New York, NY, USA: Pergamon Press, 1980.
[30] H. Cox, "Cumulative detection probabilities for a randomly moving source in a sparse field of sensors," in *Proc. Asilomar Conf.*, 1989, pp. 384–389.
[31] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 2007.
[32] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*. Hoboken, NJ, USA: Wiley, 2006.

[33] G. McLachlan, *Finite Mixture Models*. New York, NY, USA: Wiley, 2000.
[34] R. Y. Rubinstein and D. P. Kroese, "Simulation and the Monte Carlo method," in *Probability and Statistics*, 2nd ed. Hoboken, NJ, USA: Wiley, 2008.
[35] Mathworks. (2004). *Matlab Optimization Toolbox* [Online]. Available: http://www.mathworks.com, function: fmincon.
[36] R. J. Vanderbei, "LOQO: An interior point code for quadratic programming," *Optim. Methods. Softw.*, vol. 11, no. 1–4, pp. 451–484, 1999.

**Hongchuan Wei** received the BS degree from the Department of Automotive Engineering, Tsinghua University, Beijing, China. Currently, he works with the Laboratory for Intelligent Systems and Controls, Duke University, Durham, NC. His research interests include sensor network deployment and motion planning, human pilot controller design, and nonparametric Bayesian model.

**Silvia Ferrari** (SM) received the BS degree from Embry-Riddle Aeronautical University, Daytona Beach, FL, and the MA and PhD degrees from Princeton University, Princeton, NJ. She is Professor of Engineering and Computer Science at Duke University, Durham, NC, where she directs the Laboratory for Intelligent Systems and Controls (LISC). Her principal research interests include adaptive control, learning and approximate dynamic programming, and optimal control of mobile sensor networks. She is a member of ASME, SPIE, and AIAA. She is the recipient of the ONR Young Investigator Award (2004), the NSF CAREER Award (2005), and the Presidential Early Career Award for Scientists and Engineers (PECASE) Award (2006).