A Nonlinear Filter Coupled With Hospitability and Synthetic Inclination Maps for In-Surveillance and Out-of-Surveillance Tracking

Zaher M. Kassas, *Member, IEEE*, and Ümit Özgüner, *Senior Member, IEEE*

***Abstract*—This paper presents an algorithm for in-surveillance and out-of-surveillance mobile ground target tracking. In this respect, a nonlinear Bayesian estimation filter is presented. Then, an adaptive algorithm is derived to reduce remarkably the computational burden of this filter, while not degrading the accuracy of the state estimates. Additionally, an algorithm employing the concepts of hospitability and synthetic inclination maps is introduced and is coupled with the nonlinear Bayesian filter to track the mobile ground target once it goes out-of-surveillance. Loosely speaking, the hospitability map can be viewed as a terrain-based map defining a likelihood or a "weight" for each point on the earth's surface proportional to the ability of the target to move and maneuver at that location. On the other hand, the synthetic inclination map describes how the target favors certain regions within the search area, hence being "synthetically" inclined to move toward them.**

***Index Terms*—Estimation, Fokker–Planck equations, nonlinear filters, terrain mapping, tracking.**

I. Introduction

TRADITIONAL target-tracking algorithms are based on Kalman filters (KFs), extended KFs (EKFs), or similar weighted-sum Gaussian filters, which inherently assume that the target dynamics is linear or can be approximated by a linearized model [1]. However, real-world target tracking experiences tell us that target dynamics are typically nonlinear, terrain-dependent, and behavior-dependent. Many mobile sensor systems, e.g., aerial uninhabited autonomous vehicles (UAVs), have limited sensing capabilities, due to spatially restricted footprints and physical motion constraints. Consequently, a mobile ground target can be in-surveillance and out-of-surveillance frequently. In order for the mobile sensor system to take good observations of the mobile target, the mobile sensor must plan its route according to its prediction of the target state. Therefore, maintaining good predictions before taking observations is as important as generating filtered estimates when observations are available.

Manuscript received August 31, 2008; revised January 5, 2009. First published June 23, 2009; current version published November 2, 2009. This work was supported by the Air Force Research Laboratory (AFRL/VA) and Air Force Office of Scientific Research (AFOSR) Collaborative Center of Control Science under Grant F33615-01-2-3154. This paper was recommended by Associate Editor T. Busch.

Z. M. Kassas was with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA. He is now with National Instruments, Austin, TX 78759 USA, and also with the University of Texas at Austin, Austin, TX 78712, USA (e-mail: zkassas@ieee.org).

Ü. Özgüner is with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA (e-mail: umit@ece.osu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2009.2024004

It is not straightforward to fuse various useful tracking information, such as nonlinear complex target models, knowledge of the environment, and knowledge of the target behavior and "inclinations" in most traditional KF-like tracking approaches. To mitigate the difficulty in modeling complex target dynamics, interactive multiple model (IMM) filters were proposed [2]. In the IMM algorithm, a finite number of explicit models are chosen to jointly represent the target system. Consequently, the target state estimate is generated as a weighted sum of the outcomes of KFs or EKFs. Although IMM filters have proven to be successful in many tracking applications, it is difficult to straightforwardly extend them to the mobile sensor tracking scenario dealt with in this paper. This is due to two reasons. First, since the IMM algorithm prediction stage only runs right before a new observation is available, as the time interval between consecutive observations increase, the prediction degrades very quickly. Second, each KF or EKF in the IMM approach requires a linear or a linearizable system model. Consequently, nondetection information cannot be utilized directly into the IMM filter.

This has led to regained interest in alternative filtering techniques, such as particle filtering [3], [4] and nonlinear Bayesian filtering [5]–[10]. These alternative filtering techniques are being coupled with useful additional, possibly nonanalytic, target, and environment information. Among the earlier algorithms that incorporated terrain-based information into tracking mobile targets is the work by Reid and Bryson [11]. The Reid and Bryson model represented the terrain as a grid such that each cell has eight adjacent neighboring cells. For each grid cell, eight terrain factors (one per adjacent grid cell) were precalculated. These terrain factors represented the estimated modification to target velocity to each adjacent grid cell. Nougues utilized Reid and Bryson's work as a foundation to demonstrate that for tracking a single vehicle, the KF lacks the ability to integrate terrain constraints [12]. Nougues used terrain information to create a potential surface and then utilized the surface to integrate terrain information beyond the eight adjacent grid cells. Sobiesk *et al.* presented a method for improving the performance of Nougues' algorithm for tracking multiple vehicles moving across terrain [13]. They further demonstrated that parallel programming techniques made the calculation of the probability densities involved feasible. Sodtke and Llinas considered the problem of different observation rates and a terrain-based tracking algorithm was presented and compared with the KF [14].

Several different approaches have been proposed to track targets moving on road networks. Yang *et al.*, for instance, treated the problem of air-to-ground surveillance of targets

moving along road networks in the constrained estimation framework [15]. Ulmke and Koch studied the case of several intersecting roads within the IMM framework [16]. Tang and Ozguner [17] proposed an algorithm for coupling particle filters (PFs) with terrain-based maps to track known targets, whereas Sidenbladh and Wirkander presented a PF formulation for tracking an unknown and varying number of vehicles in terrain [18].

PFs have a major disadvantage in that they are extremely computationally expensive, because thousands of particles are needed to maintain certain accuracy in tracking applications [19]. This makes existing PFs impossible for real-time applications. Aimed at reducing the computational complexity, adaptive PFs have been developed to automatically determine the number of particles by bounding approximation errors with a predefined threshold [20]. Limited computational savings have been achieved by adaptive PFs. Motivated by reducing the computational burden further, an adaptive IMM–PF has been developed, where the state space is densely partitioned in high-density region(s) and coarsely partitioned in low-density region(s) to meet the error bound at an earlier stage with less particles [21]. On the basis of the premise that in most surveillance tracking applications, targets do not maneuver significantly most of the time, a multirate IMM–PF was proposed for even more computational savings [22].

This paper tackles the problem of target tracking in the nonlinear Bayesian filtering framework, where the state evolution is governed by the Fokker–Planck–Kolmogorov equation (FPKE). In nonlinear Bayesian filtering, the whole probability density function (pdf) of the target state is maintained, rather than just the first- and second-order moments as in the KF-like approaches. This premise yields better prediction and estimation of the target states and provides invaluable flexibility for modeling purposes as will be exploited in this paper. We extend the work developed in [6], [7], and [23]. In particular, two problems are studied. The first problem assumes the following scenario. In a given search area, a UAV detects a mobile ground target using its own sensor(s). While the target is in-surveillance, the UAV makes several noisy observations of the target. These observations are inputted to a real-time nonlinear Bayesian filter, which produces optimal target state estimates. The optimality criterion is defined here in the minimum mean square error (MMSE) sense. Nonlinear Bayesian filters are inherently computationally exhaustive. To overcome this problem, an adaptive algorithm will be derived to reduce remarkably the computational burden of this filter, while not degrading the accuracy of the state estimates.

The second problem assumes the following scenario. Some time after the ground target was in-surveillance, the same or another UAV views the target area again, but the target is not detected this time. There are three possibilities that could explain this discrepancy: 1) the first set of observations were not accurate enough, i.e., it was a false detection, 2) the target is still there, but the second UAV could not detect it, i.e., it was a misdetection, and 3) the target has moved away. We will assume that the target has moved away. Since the target has moved away, where should we look for it? Traditional estimation filters based on historic kinematics information alone may not work well [6], [8]. The target kinematics information could be diluted quickly as the radius of possible target locations from that of the first set of observations gets bigger. However, the previous kinematics (target route history) at least provide a center location for future possible target locations. The proposed estimation algorithm employs the knowledge gained from the first set of observations taken by the UAV and the concepts of hospitability maps (HMaps) and synthetic inclination maps (SIMaps) embedded into the optimal nonlinear Bayesian filter that was derived for the in-surveillance case. The HMap can be defined as a gridded spatial terrain-based map describing the effect of different terrain surfaces on the mobility and localization of the target objects [6], [24]. On the other hand, the SIMap describes the behavior of the target by capturing how the target favors certain regions within the search area, hence being "synthetically" inclined to move toward them [7], [23].

The rest of this paper is organized as follows. Section II presents the design of a general optimal nonlinear Bayesian filter. It also outlines an adaptive algorithm that will reduce the computational burden of the filter remarkably, while not degrading the accuracy of the state estimates. Sections III and IV introduce the concepts of HMaps and SIMaps, respectively. Section V couples the concepts presented in Sections II–IV into a dual-mode algorithm for in-surveillance and out-of-surveillance target state estimation. Section VI illustrates the concepts presented in this paper by deriving the algorithm to a particular example problem and presents the accompanying simulations and results. Concluding remarks and future work are discussed in Section VII.

II. Optimal Nonlinear Bayesian Estimation Filter Design

A. System Definition

The target dynamics will be modeled by the Itô stochastic differential equation, given by

$$d\mathbf{x}(t) = \mathbf{f}[\mathbf{x}(t), t]dt + \mathbf{G}[\mathbf{x}(t), t]d\boldsymbol{\beta}(t), \qquad t \in [t_0, T] \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the target state vector, $\mathbf{f}[\mathbf{x}(t), t]$[1] $\in \mathbb{R}^n$ is a linear or a nonlinear vector-valued function of the state variables, $\mathbf{G}[\mathbf{x}(t), t] \in \mathbb{R}^{n\times r}$, and $\{\boldsymbol{\beta}(t), t \geq t_0\} \in \mathbb{R}^r$ is a vector of independent Brownian motion processes with $\mathbb{E}\{d\boldsymbol{\beta}(t)d\boldsymbol{\beta}^{\mathsf{T}}(t)\} = \mathbf{Q}(t)dt$.

The UAV makes observations on this target that are taken at discrete-time instants t_k according to

$$\mathbf{z}_{t_k} = \mathbf{h}[\mathbf{x}_{t_k}, t_k] + \mathbf{v}_{t_k}, \qquad k = 0, 1, 2, \ldots \quad (2)$$

where $\mathbf{z}_{t_k} \in \mathbb{R}^m$ is the observation vector, $\mathbf{h}[\mathbf{x}_{t_k}, t_k] \in \mathbb{R}^m$ is a linear or a nonlinear vector-valued function of the states of the stochastic dynamical system (SDS), and $\mathbf{v}_{t_k}$ is a zero-mean white Gaussian noise process with covariance matrix $\mathbf{R}_{t_k} \in \mathbb{R}^{m\times m} > 0$.

It will be assumed that the initial pdf of the SDS defined in (1) to be known and that it is once continuously differentiable with respect to time t, and twice continuously differentiable

[1] $\mathbf{f}[\mathbf{x}(t), t] = [f_1[\mathbf{x}(t), t], \ldots, f_n[\mathbf{x}(t), t]]^{\mathsf{T}}$, $\mathbf{x}(t) = [x_1(t), \ldots, x_n(t)]^{\mathsf{T}}$.

with respect to the state variable $\mathbf{x}(t)$. In addition, it will be assumed that the function $\mathbf{h}[\mathbf{x}_{t_k}, t_k]$ in (2) is continuous in both arguments, t and $\mathbf{x}(t)$, and to be bounded for each t_k with probability 1. Moreover, the initial state $\mathbf{x}(t_0)$, the system noise $\boldsymbol{\beta}(t)$, and the measurement noise $\mathbf{v}_{t_k}$ are assumed to be all mutually independent. Finally, it will be assumed that (1) has a solution on $[t_0, T]$ in the mean square sense [25].

B. Filter Predictor and Corrector Equations

Given this SDS with the stated assumptions, it can be shown that between observations, the conditional pdf $p\left[\mathbf{x}(t), t|\mathbf{Z}_{t_k}\right]$[2] satisfies the FPKE [25], which is described by the *diffusion* process given by the partial differential equation (pde)

$$\frac{\partial}{\partial t}p = -\sum_{i=1}^{n}\frac{\partial}{\partial x_i}\{pf_i\} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial}{\partial x_i \partial x_j}\{p\,(\mathbf{G}\mathbf{Q}\mathbf{G}^{\mathsf{T}})_{ij}\} \tag{3}$$

where the following shorthand notation has been used in (3), $p = p\left[\mathbf{x}(t), t|\mathbf{Z}_{t_{k-1}}\right]$, $f_i = f_i[\mathbf{x}(t), t]$, $\mathbf{G} = \mathbf{G}[\mathbf{x}(t), t]$, and $\mathbf{Q} = \mathbf{Q}(t)$. The term $(\mathbf{G}\mathbf{Q}\mathbf{G}^{\mathsf{T}})_{ij}$ stands for the ijth element of the matrix product. Whenever a new observation, $\mathbf{z}_{t_k}$, at time t_k becomes available, the conditional pdf $p\left[\mathbf{x}(t), t|\mathbf{Z}_{t_{k-1}}\right]$ needs to be updated to the new pdf $p\left[\mathbf{x}(t), t|\mathbf{Z}_{t_k}\right]$. This can be accomplished by applying Bayes' rule, which is given as

$$p\left[\mathbf{x}(t), t|\mathbf{Z}_{t_k}\right] = \frac{p\left[\mathbf{x}(t), t|\mathbf{Z}_{t_{k-1}}\right] p\left[\mathbf{z}_{t_k}|\mathbf{x}(t)\right]}{\int p\left[\mathbf{x}(t), t|\mathbf{Z}_{t_{k-1}}\right] p\left[\mathbf{z}_{t_k}|\mathbf{x}(t)\right] d\mathbf{x}(t)} \tag{4}$$

where the term $p\left[\mathbf{z}_{t_k}|\mathbf{x}(t)\right]$ in (4) can be evaluated from

$$p\left[\mathbf{z}_{t_k}|\mathbf{x}(t)\right] = \frac{1}{(2\pi)^{m/2}\,|\mathbf{R}_{t_k}|^{1/2}} \mathrm{e}^{-(1/2)\left[\mathbf{z}_{t_k}-\mathbf{h}\right]^{\mathsf{T}}\mathbf{R}_{t_k}^{-1}\left[\mathbf{z}_{t_k}-\mathbf{h}\right]}$$

where $\mathbf{h} = \mathbf{h}[\mathbf{x}_{t_k}, t_k]$. Essentially, (3) and (4) constitute the predictor and corrector equations, respectively, of the nonlinear Bayesian filter. The mean of the conditional pdf in (4) with respect to the state variable yields the optimal state estimate in a MMSE sense, i.e.

$$\hat{\mathbf{x}}(t) = \int_{-\infty}^{\infty}\mathbf{x}(t)p\left[\mathbf{x}(t), t|\mathbf{Z}_{t_k}\right] d\mathbf{x}(t).$$

The nonlinear Bayesian filter just defined generally possesses two undesirable characteristics: difficulty to achieve a closed-form analytical solution for higher order complicated FPKEs and infinite-dimensionality. These two issues will be dealt with separately in the following two sections.

C. Numerical Solutions to the FPKE Via Finite-Difference Method

Generally, it is extremely difficult to obtain an analytical solution to the FPKE, except in two special cases: 1) when the solution of the FPKE is stationary and 2) when the SDS has its pdf in an analytic tractable functional form. In the case when the FPKE exhibits a stationary solution, the evolving densities approach a stationary pdf, which settles around a certain mean. Thereafter, as time evolves, the significant values of the pdf do not move over the state variable domain. However, in certain nonlinear filtering problems (e.g., target tracking), the SDS rarely exhibits a stationary solution.

This motivates employing numerical techniques to solve the FPKE. The commonly used methods to solve the FPKE numerically are explicit and implicit finite-difference methods [26], [27] and finite-element methods [28], [29]. Other candidate methods include spectral methods [30], probabilistic data association, and multiple hypothesis, multiple frame methods. In this paper, the explicit finite-difference method is presented for its ease of implementation and for quick demonstration.

In the finite-difference method, the time and state variable domains are discretized and defined over a finite number of equally spaced grid points. Consequently, the partial derivatives are evaluated at each of the grid points/cells, which are represented as $(x_j, t_i) = (j\Delta x, i\Delta t)$, $1 \le j \le k+1$, and $i \ge 0$, where $k \in \mathbb{N}$, Δx is the discretization step in the state variable domain, and Δt is the discretization step in time. The following finite-difference approximations can be used to approximate the solution of the FPKE in (3)

$$\frac{\partial f(x,t)}{\partial t}\Big|_{t=t_{i-1}} = \frac{f(x,t_i) - f(x,t_{i-1})}{\Delta t}$$

$$\frac{\partial f(x,t)}{\partial x}\Big|_{t=t_{i-1}} = \frac{f(x_{j+1},t_{i-1}) - f(x_{j-1},t_{i-1})}{2\Delta x}$$

$$\frac{\partial^2 f(x,t)}{\partial x^2}\Big|_{t=t_{i-1}} = \frac{f(x_{j+2},t_{i-1}) - 2f(x_j,t_{i-1}) + f(x_{j-2},t_{i-1})}{(2\Delta x)^2}.$$

The stability condition for the explicit finite-difference method states that the temporal and spatial grid resolutions must satisfy the inequality

$$0 < \Delta t\sum_{i=1}^{n}\frac{1}{(\Delta x_i)^2} \le \frac{1}{2} \tag{5}$$

where Δx_i is the grid spacing of the ith state variable.

D. Adaptive State Variable Domain Truncation Algorithm

The fact that nonlinear Bayesian filters employing the FPKE are generally infinite-dimensional implies that some form of state-variable-domain truncation would be necessary, when numerical techniques are employed. The most common approach for domain truncation adopted in literature is to set the state-variable-domain limits very wide in order to ensure that the significant mass of the pdf would lie within these preset limits at all time. However, in most practical applications, this would burden the filter with exhaustive computations that would contribute too little to the overall mass of the pdf. To overcome the computational burden without sacrificing the estimation accuracy, adaptive state-variable-domain truncation algorithm has been proposed [7], [31].

The algorithm of adaptive state-variable-domain truncation employs the Chebyshev inequality theorem, also known as the Chebyshev bound (CB), along with the moment evolution theorem to "optimally adjust and truncate" the state-variable-domain limits, and hence the pdf size. This adjustment and truncation

[2] $\mathbf{Z}_{t_k} \triangleq \{\mathbf{z}_{t_k}, \mathbf{Z}_{t_{k-1}}\}$.

ensures that the significant mass of the pdf lies within these limits at every time step; hence, not degrade the accuracy of the estimates.

The Chebyshev inequality theorem [32] states that given an arbitrary random variable x, with mean $\mathbb{E}\{x\}$, and finite variance σ_x^2, for any $\lambda > 0$

$$P\{|x - \mathbb{E}\{x\}| \geq \lambda\sigma_x\} \leq \frac{1}{\lambda^2}.$$

One way to look at the CB is that it provides an upper bound on the amount of probability in the "tails" of a given pdf. Therefore, we may think of the CB as providing an upper bound on the *loss* of probability due to truncation of the pdf.

The union bound [32], states that given n events $\{E_1, E_2, \ldots, E_n\}$ (not necessarily disjoint), the probability of the union of these events is less than or equal to the summation of the events' individual probabilities, i.e.

$$P\left\{\bigcup_{i=1}^{n} E_i\right\} \leq \sum_{i=1}^{n} P\{E_i\}.$$

Using the union bound, the CB can be generalized to the case of n random variables $[x_1, \ldots, x_n]$, with respective means $[\mathbb{E}\{x_1\}, \ldots, \mathbb{E}\{x_n\}]$ and variances $\left[\sigma_{x_1}^2, \ldots, \sigma_{x_n}^2\right]$ as

$$P\left\{\bigcup_{i=1}^{n} [|x_i - \mathbb{E}\{x_i\}| < \lambda_i\sigma_{x_i}]\right\} \geq 1 - \varepsilon \tag{6}$$

where ε is the design parameter to be selected, given by

$$\varepsilon = \sum_{i=1}^{n} \frac{1}{\lambda_i^2}.$$

There are two approaches for solving for ε, one is based on other Chebyshev-type inequalities, and the other is based on use of weighted values of λ's [33]. The approach we employed is to select $\{\lambda_i\}_{i=1}^{n} \equiv \lambda$, which is rationalized by the fact that the σ-limits of all the state variables are of equal importance, hence no preferred weighting is given to one state variable on the expense of the other.

In order to adaptively find the optimal state-variable-domain limits, the present and future domain limits need to be calculated. The union of these limits yields the optimal state variable domain for the complete time pdf evolution, from the present time through the future time. On one hand, since, at a given time step, the current pdf is available, finding the first and second moments is straightforward. On the other hand, at a given time step, the future evolving pdf is unknown; hence, the future first and second moments cannot be computed as directly. Nevertheless, future first and second moments may still be predicted through the moment evolution theorem [25].

Given an SDS characterized by (1), observations made according to (2), and under some boundedness, Lipschitzianity, and independence hypotheses and assumptions, the moment evolution theorem states that between observations, the conditional mean and covariance of the pdf $p[\mathbf{x}(t), t|\mathbf{Z}_{t_k}]$ satisfy

$$\frac{d}{dt}\mathbb{E}\{\mathbf{x}(t)\} = \mathbb{E}\{\mathbf{f}[\mathbf{x}(t), t]\} \tag{7}$$

$$\frac{d}{dt}\mathbf{P}(t) = \mathbb{E}\left\{\mathbf{x}\mathbf{f}^{\mathsf{T}}\right\} - \mathbb{E}\{\mathbf{x}\}\mathbb{E}^{\mathsf{T}}\{\mathbf{f}\} + \mathbb{E}\left\{\mathbf{f}\mathbf{x}^{\mathsf{T}}\right\} - \mathbb{E}\{\mathbf{f}\}\mathbb{E}^{\mathsf{T}}\{\mathbf{x}\} + \mathbb{E}\left\{\mathbf{G}\mathbf{Q}\mathbf{G}^{\mathsf{T}}\right\} \tag{8}$$

where the following shorthand notation has been used in (8), $\mathbf{x} = \mathbf{x}(t)$, $\mathbf{f} = \mathbf{f}[\mathbf{x}(t), t]$, $\mathbf{G} = \mathbf{G}[\mathbf{x}(t), t]$, and $\mathbf{Q} = \mathbf{Q}(t)$. Actually, it was discussed in [7] that in certain cases, we need not evolve the entire covariance matrix, since computing only the diagonal terms of the evolved covariance matrix would suffice. This reduces the number of computations from $(n(n+1))/2$ to n, where n is the dimension of the state vector $\mathbf{x}(t)$.

In summary, the adaptive state-variable-domain truncation algorithm consists of the following steps. Given the pdf $p[\mathbf{x}(t), t|\mathbf{Z}_{t_k}]$ at time t_k, with associated state-variable-domain limits. First, evaluate the mean vector $\mathbb{E}\{\mathbf{x}(t)\}$ and the covariance matrix $\mathbf{P}(t)$ from $p[\mathbf{x}(t), t|\mathbf{Z}_{t_k}]$. Then, for a chosen design parameter λ, apply the CB (6) to obtain the current state-variable-domain limits. Next, apply the mean and covariance evolution theorems (7) and (8) to evaluate the future mean vector and covariance matrix. Equations (7) and (8) can be evaluated numerically from the pdf available at the current time by means of numerical integration methods (e.g., Euler). Consequently, for the same λ apply the CB (6) to obtain the future-state-variable-domain limits. The union of the current- and future-state-variable-domain limits yields the optimal state variable domain for the complete time pdf evolution, from the present time through the future time. The union of these limits should be used to evolve the FPKE (3).

III. Hospitability Map

The earth's surface is rarely *flat* and *unvarying* in a given search area of large enough dimensions. Hence, it is quite logical to assume that the different terrain a search area may exhibit should influence where and how the target moves at a specific locale. This motivates introducing the concept of the HMap to constrain the search area into regions of higher/lower hospitability for a given class of targets.

A number of US Army programs have worked extensively to characterize vehicle motion preferences in terms of a quantity referred to as "hospitability for maneuver" (HM) [24], [34], [35]. These preferences vary with position, weather, vehicle type, and mission. HM characterizes the ease with which a vehicle can traverse a particular area and can be used to generate the preferred heading and velocity with which a vehicle will operate in an area. HM data are quite fine grained and can vary by an order of magnitude over spatial scales on the order of tens of meters, say the width of a road.

Layne *et al.* [6] defined the HMap as a map providing a likelihood or a "weight" for each point on the earth's surface proportional to the ability of a target to move and maneuver at that location. Therefore, one may think of the HMap as partitioning the search area into different "contours" depending on how a particular cell/region is hospitable for a given class of targets.

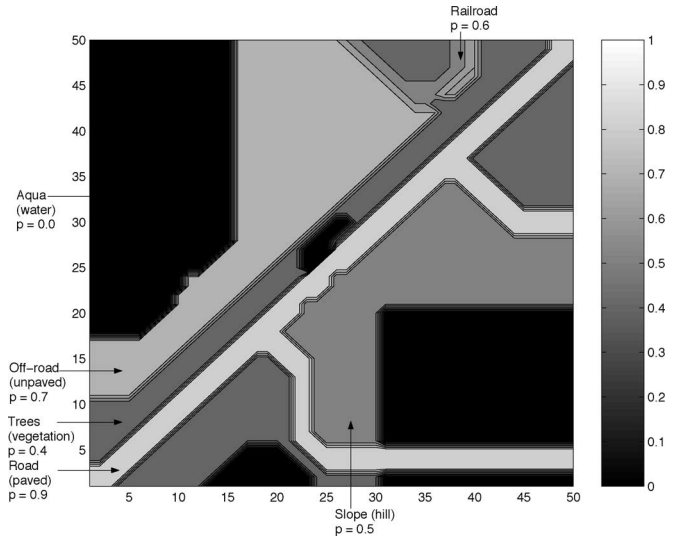

Fig. 1. Typical HMap.

Here, high HMap values denote that the target can move and maneuver quickly and easily over the corresponding terrain. In contrast, low HMap values indicate that the target cannot move easily over that terrain. A thorough study on the construction of HMaps can be found in [24]. The following factors are considered while deriving the hospitability value: slope, surface roughness, transportation, geology, landform, soil, vegetation, hydrology, urban areas, and climate. Additionally, the HMap is a function of the nature of the target itself, i.e., there exists different models for deriving the HMap for different classes of targets (e.g., cars, tanks, amphibious vehicles, etc.). A typical HMap is illustrated in Fig. 1, where the HMap values have been normalized to lie in the interval $[0, 1]$, with 1 being the highest hospitability index.

Fig. 2 illustrates a typical data flow for terrain analysis that is used to derive the hospitability index. The process consists of three main steps. First, raw terrain data are extracted from imagery, maps, literature, or data collected during on-site inspections. These data are analyzed and reduced to a set of terrain factor products capturing important terrain features and classifications [34]. Second, the terrain factor products are combined with empirical and doctrinal evidence to generate complex products capturing activities of military significance [35]. Finally, these factors are combined with analytical models, which capture the capabilities of specific vehicles (usually empirically derived), to produce the hospitability index [24].

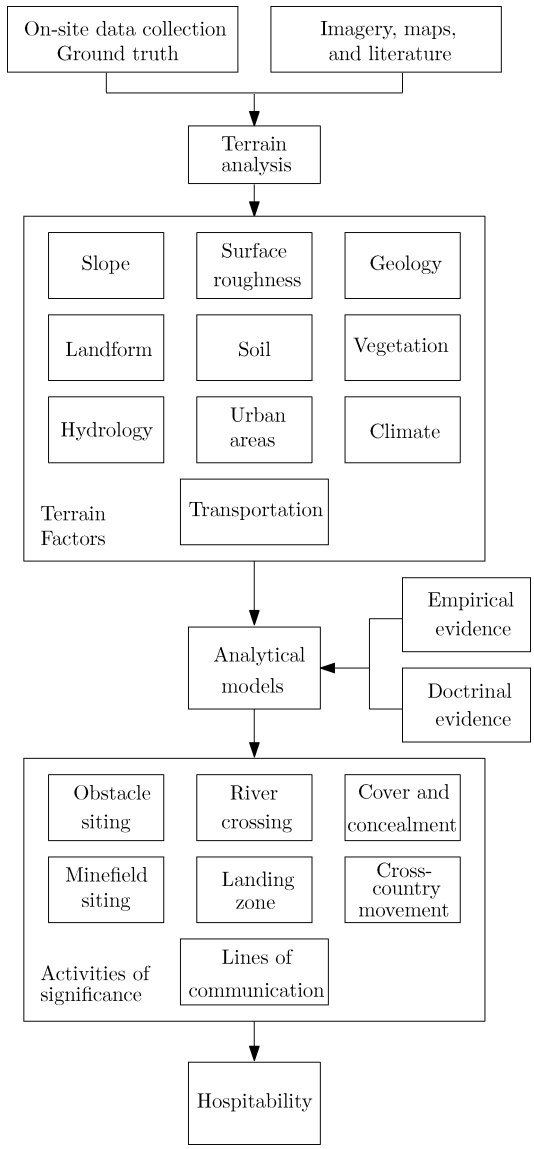

Fig. 2. Data flow in terrain analysis.

IV. Synthetic Inclination Map

Since the problem under consideration deals with mobile ground targets moving over a specific search area, it is quite logical to assume that these targets are not moving *randomly* and that they are heading toward certain regions with higher/lower probabilities. In this respect, we will introduce the concept of the SIMap, which is a function of the velocity components, and hence the heading of the target. Specifically, the SIMap is a two-dimensional (2-D) velocity map in $(\dot{x}, \dot{y})$ defined for each 2-D point (x, y). For a given target located at a particular point (x_0, y_0), the SIMap describes how the target will be *synthetically inclined* to move in different directions with certain velocity components. Therefore, whereas the HMap is a function of the nature of the target and the topography of the search area, the SIMap is a function of the nature of the target and how the target "favors" different zones.

Fig. 3 shows a simple way of looking at SIMaps. In a given search area $\mathcal{S}$, we assume that we are given a priori estimates about the target initial position, denoted by (x_0, y_0). Additionally, we will assume that within $\mathcal{S}$, there is a zone Ω where the target is interested to head toward *as fast as possible*. Hence, we say that the target is *synthetically inclined* to move toward Ω. Consequently, we may define a weighting function that assigns a "weight" for each of the velocity component combinations the target may have.

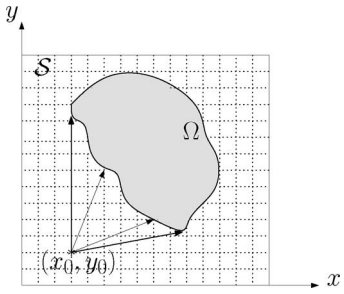

Fig. 3. SIMap illustration.

The construction of SIMaps depend on several factors. The nature of the target itself is one important factor in determining the velocity components domain the target may have. For instance, if the target under consideration is a tank with a maximum speed $s_{\max}$, then we can safely assign the SIMap domain to take positive and negative values of $(\dot{x}, \dot{y})$ satisfying the inequality $\sqrt{(\dot{x})^2 + (\dot{y})^2} \leq s_{\max}$, as we are sure that the target cannot move with higher velocity components. Another consideration affecting the construction of SIMaps is tactical background information. For instance, we can subdivide the search area into different zones, such as zones where the target would like to reach as fast as possible (e.g., regions where there are enemy troops the target would like to attack), zones where the target would like to escape from as fast as possible (e.g., regions where the target might have *a priori* information that they will be struck soon), zones where the target would like to stay stationary at (e.g., hiding regions), and zones where the target would have different $\dot{x}$ and $\dot{y}$ velocity components (e.g., roads and railroads) [7].

It is worth noting that after the SIMaps were first introduced in [7] and [23], somewhat similar approaches have been proposed, referred to as the vehicle preference, which are derived from target activity analysis [8]. The generation of activity-specific products is an advance over classical terrain analysis and intelligence preparation procedures. These products capture interactions between activity requirements (e.g., need for hide site) and constraints or opportunities provided by local terrain.

V. In-Surveillance and Out-of-Surveillance Target-Tracking Algorithm

This section integrates the concepts developed in Sections II–IV. First, we combine the HMap and SIMap into a 4-D map in $(x, y, \dot{x}, \dot{y})$. The combined HMap–SIMap may be understood by fixing the (x, y) coordinates at a particular point (x_0, y_0) and varying the velocity components $(\dot{x}, \dot{y})$ over their domain. To illustrate this concept, assume that the velocity components are restricted to the discrete values: $\{0, \pm 1, \pm 2, \pm 3\}$. Consequently, each fixed point in the 2-D HMap layer will be associated with a 2-D SIMap layer, as demonstrated in Fig. 4. Note that there are 7^2 different velocity components combinations the target may have, each represented by a vector with corresponding magnitude and direction.

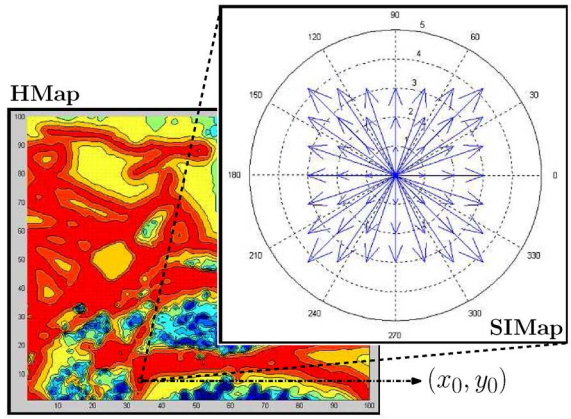

Fig. 4. Relationship between the HMap and SIMap layers. For each fixed point (x_0, y_0) on the HMap layer, an associated SIMap is constructed.

Next, we will couple the combined 4-D HMap–SIMap with the nonlinear Bayesian filter derived in Section II. The algorithm objective is to produce target state estimates for in-surveillance and out-of-surveillance mobile ground targets. For the in-surveillance case, whenever a UAV detects a mobile ground target using its onboard sensors, the optimal nonlinear Bayesian filter derived in Sections II should take over and produce target state estimates. However, once the target goes out-of-surveillance, the filter cannot update the pdf anymore, since no new observations are being made on the target. Consequently, we will define the following update equation:

$$p^{+}[\mathbf{x}(t_k), t_k] = \frac{1}{a} p^{-}[\mathbf{x}(t_k), t_k] \cdot c^{-1}(\mathbf{x}) \tag{9}$$

where $p^{+}[\mathbf{x}(t_k), t_k]|$ and $p^{-}[\mathbf{x}(t_k), t_k]|$ are the updated and unupdated pdfs at time t_k, respectively, and a is a normalizing constant to ensure that the pdf $p^{+}[\mathbf{x}(t_k), t_k]$ integrates to 1.

The combined HMap–SIMap cost function $c(\mathbf{x})$ is essentially a 4-D array, defined for all points $\mathbf{x}_0$, where $\mathbf{x}_0$ is the state vector $\mathbf{x}$ evaluated at a particular point $\mathbf{x}_0$. In particular, $\mathbf{x}_0 = [x_0, y_0, \dot{x}_0, \dot{y}_0]^{\mathsf{T}}, \forall\,(x_0, y_0) \in \mathcal{S}$ and $\forall\,(\dot{x}_0, \dot{y}_0) \in [-v_{\max}, +v_{\max}]$, with $\mathcal{S}$ being the search area under consideration and $v_{\max}$ being the maximum x or y velocity component the target may have.

The inverse of the cost function $c(\mathbf{x})$ will be defined as a linear combination of the hospitability weight and the exponential of the negative Euclidean norm between a given point $\mathbf{x}_0' = (x_0 + \dot{x}_0 \Delta t, y_0 + \dot{y}_0 \Delta t)$ and a boundary condition (final point) $\mathbf{x}_f = (x_f, y_f)$, with Δt being the discretization time step. In particular, $c^{-1}(\mathbf{x})$ will be defined as

$$c^{-1}(\mathbf{x}_0) = \alpha\, \mathbf{H}(x_0, y_0) + \gamma\, \mathrm{e}^{-\boldsymbol{\delta}}$$

$$\boldsymbol{\delta} = \sqrt{[x_f - (x_0 + \dot{x}_0 \Delta t)]^2 + [y_f - (y_0 + \dot{y}_0 \Delta t)]^2}$$

where α and γ are weighting scalars, $\mathbf{x}_0$ is the 4-D point defined previously, $\mathbf{H}(x_0, y_0)$ is the hospitability weight at $\mathbf{x}_0$, and $\boldsymbol{\delta} = \|\mathbf{x}_f - \mathbf{x}_0'\|_2$ is the Euclidean norm between the final point $\mathbf{x}_f$ and the point the target would reach if it moves from (x_0, y_0) with velocity components $(\dot{x}_0, \dot{y}_0)$ in Δt time step. Note that the weighting scalars α and γ can be selected according to the

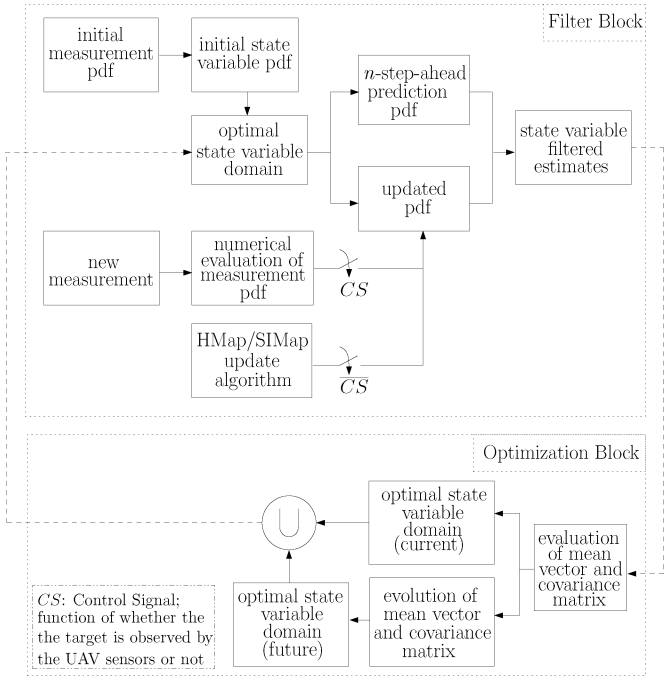

Fig. 5. Filter block diagram.

respective weighting we want to associate the hospitability term with versus the synthetic inclination term. Fig. 5 summarizes the dual-mode algorithm for tracking an in-surveillance and out-of-surveillance mobile ground target.

VI. Simulations and Results

A. Filter Derivation

The ground target will be assumed to be moving in the 2-D Cartesian xy-plane according to the linear motion model with nearly constant velocity, where the acceleration is modeled as a stationary white Gaussian noise. The target states will be defined to be the Cartesian (x, y) position and velocity. So

$$\mathbf{x}(t) = [x(t),\, y(t),\, \dot{x}(t),\, \dot{y}(t)]^{\mathsf{T}},$$

$$\mathbf{f}[\mathbf{x}(t), t] = [\dot{x}(t),\, \dot{y}(t),\, 0,\, 0]^{\mathsf{T}}$$

$$\mathbf{G}[\mathbf{x}(t), t] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{Q}(t) = \begin{bmatrix} \sigma_q^2 & 0 \\ 0 & \sigma_q^2 \end{bmatrix}.$$

The target tracking will be realized via the onboard sensor(s) of a UAV flying in the 3-D Cartesian xyz-space also according to the nearly constant velocity model. The UAV may detect the target, and hence produce observations, if the target falls within a predefined rectangular footprint of the UAV sensors. The measurements provided by the UAV sensors are the range r, and the bearing angle ϕ, both corrupted by additive white Gaussian noise. Note that the measurements are produced with respect to the UAV as being the origin of the spherical coordinate frame system, defined by the triplet (r, θ, ϕ). Hence, given that the UAV is located at (x_u, y_u, z_u) and the target located at $(x_t, y_t, 0)$, (2) simplifies to

$$\mathbf{z}_{t_k} = [\, r_{t_k} \quad \phi_{t_k} \,]^{\mathsf{T}}, \qquad \mathbf{R}_{t_k} = \mathrm{diag}[\sigma_r^2, \sigma_\phi^2]$$

$$\mathbf{h}[\mathbf{x}(t), t] = \begin{bmatrix} \sqrt{(\Delta x_{ut})^2 + (\Delta y_{ut})^2 + z_u^2} \\ \tan^{-1}\left[\frac{\Delta y_{ut}}{\Delta x_{ut}}\right] \end{bmatrix}$$

where $\Delta x_{ut} = x_u - x_t$ and $\Delta y_{ut} = y_u - y_t$. Through a spherical to Cartesian pdf function transformation, we can extract from the initial observation, $\mathbf{z}_{t_0}$, the initial state variable pdf $p\,[\mathbf{x}(t), t|\mathbf{Z}_{t_0}]$ that will eventually evolve with time. For simulation purposes, we will assume that the marginal pdfs $p\,(\dot{x}_t)$ and $p\,(\dot{y}_t)$ to be independent and uniformly distributed over the different velocity components. Assuming that the initial position pdf $p\,(x_t, y_t)$ and the initial velocity pdf $p\,(\dot{x}_t, \dot{y}_t)$ to be jointly independent, and after some derivation, it can be shown that the initial pdf is given by

$$p = \frac{1}{(2v_{x,\max})}\frac{1}{(2v_{y,\max})} \cdot \frac{1}{\sqrt{(\Delta x_{ut})^2 + (\Delta y_{ut})^2 + z_u^2}}$$
$$\times \frac{1}{\sqrt{2\pi\sigma_r^2}} e^{-\left[\sqrt{(\Delta x_{ut})^2 + (\Delta y_{ut})^2 + z_u^2} - r(0)\right]^2 / 2\sigma_r^2}$$
$$\times \frac{1}{\sqrt{2\pi\sigma_\phi^2}} e^{-[\tan^{-1}(\Delta y_{ut}/\Delta x_{ut}) - \phi(0)]^2 / 2\sigma_\phi^2}$$

where $v_{x,\max}$ and $v_{y,\max}$ are the x and y maximum velocity components the target may have, respectively, $\{\dot{x}_t, \dot{y}_t\} \in [-v_{\max}, +v_{\max}]$, $\{x_t, x_u, y_t, y_u\} \in (-\infty, +\infty)$, $z_u \in \mathbb{R}^+$, $r \in [z_u, \infty)$, and $\phi \in [-\pi, \pi)$. The boundary conditions (B.C.) of the pdf are the Dirichlet conditions, i.e., the solution at the boundaries are kept constant at 0. These boundaries are usually defined at ∞, but since we are dealing with numerical methods, we set the B.C. to be defined at the limits obtained from the CB as described in Section II-D.

The filter predictor equation (3) is readily evaluated as

$$\frac{\partial p}{\partial t} = -\dot{x}\frac{\partial p}{\partial x} - \dot{y}\frac{\partial p}{\partial y} + \frac{\sigma_q^2}{2}\frac{\partial^2 p}{\partial \dot{x}^2} + \frac{\sigma_q^2}{2}\frac{\partial^2 p}{\partial \dot{y}^2} \tag{10}$$

where we used the shorthand notation $p = p\,[\mathbf{x}(t), t|\mathbf{Z}_{t_{k-1}}]$ for simplicity of notation. By applying the finite-difference approximations discussed in Section II-C, we can approximate the solution of the FPKE pde in (10) as

$$p\,(k+1, i, j, m, n)$$
$$= p\,(k, i, j, m, n) - \frac{\Delta t\, \dot{x}}{2\Delta x} A - \frac{\Delta t\, \dot{x}}{2\Delta x} B$$
$$+ \frac{\Delta t\, \sigma_q^2}{2\,(2\Delta \dot{x})^2} C + \frac{\Delta t\, \sigma_q^2}{2\,(2\Delta \dot{y})^2} D$$
$$A = p\,(k, i+1, j, m, n) - p\,(k, i-1, j, m, n)$$
$$B = p\,(k, i, j+1, m, n) - p\,(k, i, j-1, m, n)$$
$$C = p\,(k, i, j, m+2, n) - 2p\,(k, i, j, m, n) + p\,(k, i, j, m-2, n)$$
$$D = p\,(k, i, j, m, n+2) - 2p\,(k, i, j, m, n) + p\,(k, i, j, m, n-2)$$

where $p(k,i,j,m,n)$ stands for the probability $p\{[\mathbf{x}(t_{t_k})|\mathbf{Z}_{t_k}] = [i\Delta x, j\Delta y, m\Delta\dot{x}, n\Delta\dot{y}]^{\mathsf{T}}\}$.

In order to apply the adaptive state-variable-domain truncation algorithm presented in Section II-D, we need to evaluate the current and future mean vectors and covariance matrices. On one hand, the current-state-variable-domain limits can be found by applying the CB (6), which evaluates to $x_i(t) \in (\mathbb{E}\{x_i(t)\} - \lambda_i\sigma_{x_i}(t), \mathbb{E}\{x_i(t)\} + \lambda_i\sigma_{x_i}(t))$, where $\mathbb{E}\{x_i(t)\}$ and $\sigma_{x_i}(t)$ are the mean and standard deviation associated with the ith state variable $x_i(t)$, respectively.

On the other hand, applying (7) from the mean evolution theorem, and using Euler approximation to compute the derivative, we get the future mean vector components as

$$\mathbb{E}\{x(t+\Delta t)\} = \mathbb{E}\{x(t)\} + \mathbb{E}\{\dot{x}(t)\}\Delta t$$
$$\mathbb{E}\{y(t+\Delta t)\} = \mathbb{E}\{y(t)\} + \mathbb{E}\{\dot{y}(t)\}\Delta t$$
$$\mathbb{E}\{\dot{x}(t+\Delta t)\} = \mathbb{E}\{\dot{x}(t)\}, \qquad \mathbb{E}\{\dot{y}(t+\Delta t)\} = \mathbb{E}\{\dot{y}(t)\}.$$

Before proceeding to compute the future covariance matrix, we note that in order to apply the CB to find the future state variable domain, all what we still need to find are the standard deviations $\sigma_x(t+\Delta t)$, $\sigma_y(t+\Delta t)$, $\sigma_{\dot{x}}(t+\Delta t)$, and $\sigma_{\dot{y}}(t+\Delta t)$. These standard deviations are nothing but the square roots of the diagonal of the evolved covariance matrix. Employing Euler approximation in (8) yields

$$\sigma_x^2(t+\Delta t) = \sigma_x^2(t) + [2\,\mathbb{E}\{x(t)\,\dot{x}(t)\} - 2\,\mathbb{E}\{x(t)\}\,\mathbb{E}\{\dot{x}(t)\}]\Delta t$$
$$\sigma_y^2(t+\Delta t) = \sigma_y^2(t) + [2\,\mathbb{E}\{y(t)\,\dot{y}(t)\} - 2\,\mathbb{E}\{y(t)\}\,\mathbb{E}\{\dot{y}(t)\}]\Delta t$$
$$\sigma_{\dot{x}}^2(t+\Delta t) = \sigma_{\dot{x}}^2(t) + \sigma_q^2\Delta t, \qquad \sigma_{\dot{y}}^2(t+\Delta t) = \sigma_{\dot{y}}^2(t) + \sigma_q^2\Delta t.$$

Therefore, the future-state-variable-domain limits evaluate to $x_i(t+\Delta t) \in (\mathbb{E}\{x_i(t+\Delta t)\} - \lambda_i\sigma_{x_i}(t+\Delta t), \mathbb{E}\{x_i(t+\Delta t)\} + \lambda_i\sigma_{x_i}(t+\Delta t))$. The union of the current and future limits yields the state-variable-domain limits for the complete evolution time.

When coding the adaptive algorithm, the following observation is realized. Since the new lower and upper domain limits could be a stretched, contracted, and/or translated version of the old lower and upper domain limits, we have to consider $(2n)^2$ cases in order to ensure proper domain assignment, where n is the number of state variable domains to which we are applying the adaptive algorithm. Essentially, all the cases fall under two main themes, the first being that the new lower/upper limits are smaller than the old limits, while the second being that the new limits are larger than the old limits. In the first case, we should truncate the old pdf, whereas in the second case, we should lengthen the old pdf by zero-padding.

B. Results

The target and UAV initial positions were set at $(0,0,0)$ and $(-1200,-3200,900)$, respectively. The velocity state variables $(\dot{x},\dot{y})$ were restricted to the set $\{0.0,\pm0.5,\pm1.0\}$. For this domain, the grid spacing in $(\dot{x},\dot{y})$ is $\Delta\dot{x} = \Delta\dot{y} = 0.5$. The grid spacing in (x,y) were chosen to be $\Delta x = \Delta y = 1$. The temporal resolution was chosen to be $\Delta t = 0.04$, in accordance with the stability condition in (5). Since the sizes of the velocity domains were relatively small, we opted to apply the adaptive algorithm to the position domains only.

Two scenarios were simulated. The first assumed that the mobile ground target is in-surveillance. The noisy observations made on the target were filtered through three filters: two nonlinear Bayesian filters (with and without domain truncation) and an EKF. The nonlinear Bayesian filter without domain truncation employed only the filter block shown in Fig. 5, while setting the x and y limits wide enough to ensure that the significant mass of the pdf lied within these limits at all time. The question that arose when setting these static limits was: how to characterize "wide enough"? Several factors were considered. First, the noise statistics of the system under consideration. This is due to two reasons: 1) as the system becomes noisier, the evolving pdf will tend to "wander" around more and 2) as the measurements become noisier their corresponding pdf will be "spread-out" more. Second, the duration over which we wish to run the filter, since as we track a mobile target for a longer period of time, the pdf will evolve over a larger domain. Third, the resolution of the grid.

For the tracking problem in hand, we wanted to run the filter for 100 iterations. Upon calculating the pdf $p(x,y)$ from $p(x,y,\dot{x},\dot{y})$ and plotting its evolution, we found the lower and upper limits to be $(x_l = -100, y_l = -50)$ and $(x_u = 100, y_u = 150)$, respectively. Consequently, the size of the 4-D pdf that needed to be maintained for the nonlinear filter that is uncoupled with the domain truncation algorithm is $200\times200\times5\times5 = 1\times10^6$. As for the nonlinear filter that is coupled with the adaptive algorithm, the significance level was chosen to be 99% of the overall mass of the pdf, thus yielding $\lambda = \sqrt{200}$.

Figs. 6 and 7 show the mean-squared estimation error (MSEE) in the x and y positions of the target, respectively, for 50 Monte Carlo (MC) simulation runs for the following datasets: noisy observations, EKF estimates, and untruncated and truncated Bayesian nonlinear filter estimates. It was found that the error induced in the x and y position estimates due to pdf truncation had a mean of 2.3×10^{-4} and 7.5×10^{-5}, respectively, and variance of 2.5×10^{-9} and 3.9×10^{-10}, respectively. Fig. 8 illustrates the computational load of each of the Bayesian nonlinear filters for 50 MC runs. The computational load is defined as the size of the 4-D pdf that needs to be maintained at every time step.

The following observations can be made. First, the superiority of the nonlinear filter estimates versus their EKF counterparts is noted. Second, the estimates produced by the truncated and the untruncated versions of the nonlinear Bayesian filters were pretty close, and the induced error in estimation due to truncation is almost negligible. This is due to the fact that the CB is a *loose* bound; hence, even though we have set the significant mass of the pdf to 99%, the new limits will tend to ensure a much higher pdf mass. The computational load reduction due to truncation was tremendous, however.

The second scenario assumed that the target went out-of-surveillance after it was detected by the UAV sensors. Hence, the pdf is now updated through the HMap–SIMap updating algorithm that was outlined in Section V. We used the HMap shown in Fig. 1 for this scenario. The location at which the

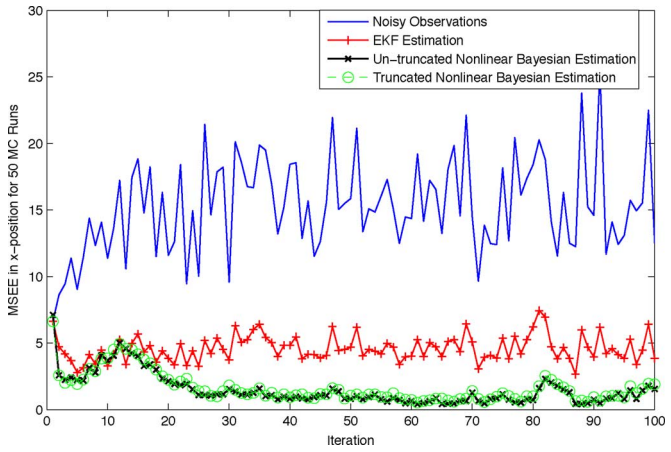

Fig. 6. MSEE in x-position for noisy measurements, EKF estimates, and untruncated and truncated Bayesian filters.

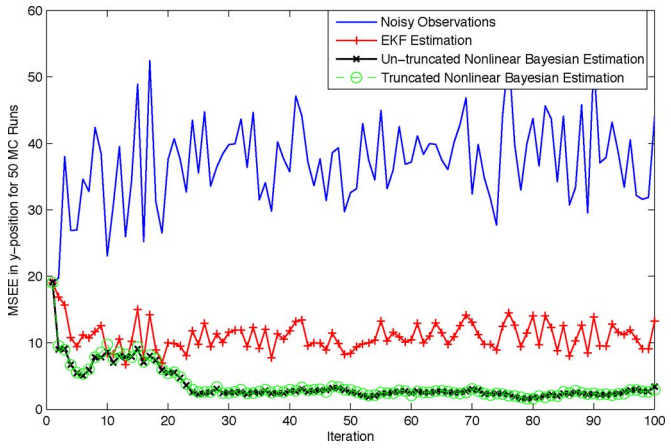

Fig. 7. MSEE in y-position for noisy measurements, EKF estimates, and untruncated and truncated Bayesian filters.

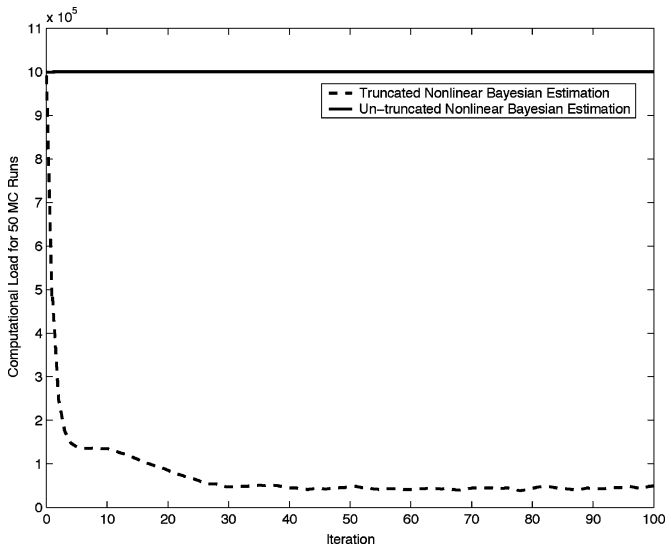

Fig. 8. Computational load of untruncated versus truncated Bayesian filter.

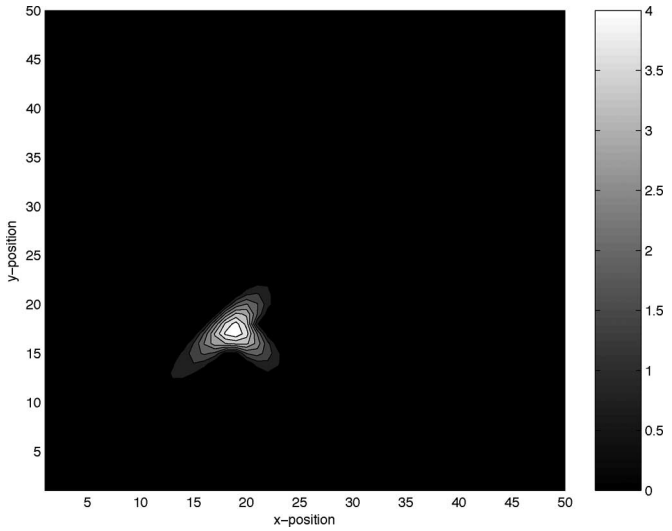

Fig. 9. Pdf propagation of target position with HMap term alone.

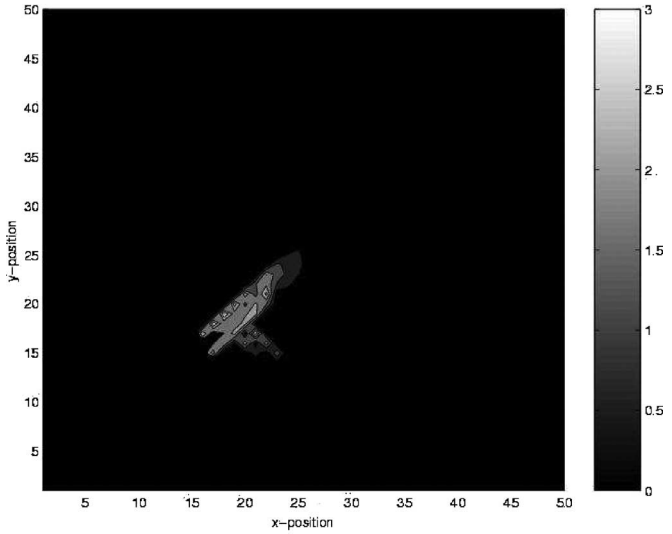

Fig. 10. Pdf propagation of target position with HMap and SIMap terms.

target went out-of-surveillance was at $(15, 15)$ and we assumed that the target was interested to head toward the boundary point $(40, 40)$. Two cases have been studied. The first propagated the pdf in the proposed filter for 40 iterations, but the cost function consisted of the HMap term alone. The second propagated the same pdf in the proposed filter for 40 iterations, but the cost function consisted of both the HMap and SIMap terms. Figs. 9 and 10 illustrate the resulting pdfs, where the color bars represent the pdf value $p(x_i, y_j)$ (in percent) $\forall (i, j) \in \mathcal{S}$. It can be seen that the pdf resulting from the incorporation of the HMap term alone yielded a pdf that is splitting at the junction of the two roads, since they are both less costly (as hospitable) for the pdf to propagate through. However, when incorporating the HMap term along with the SIMap term, the pdf did not split into the two junctions, since the SIMap term was *redirecting* the pdf to move toward the boundary point rather than merely following the hospitable cells in the search area.

Further simulations showed that if we merely propagated the pdf without constraining it through neither the HMap nor the SIMap resulted in a pdf that kept *diffusing* over time in all directions; hence, making the probable areas at which the target may be located larger and larger. However, in the presence of the HMap-only, the evolving pdf was more constrained, yet it kept "splitting" whenever a junction of road segments and/or a junction of relatively equally hospitable regions were present in the search area. Finally, the incorporation of the SIMap along with the HMap constrained the evolving pdf even more, which, in turn, aided the same or another UAV to recapture the target in a shorter period of time.

VII. Conclusion

This paper presented an algorithm for estimating the states of in-surveillance and out-of-surveillance mobile ground targets. For the in-surveillance case, an optimal nonlinear Bayesian filter was presented. Two setbacks are typically associated with this filter, namely difficulty to achieve closed-form analytical solutions and computational burden. To overcome the first setback, a numerical approach based on finite-difference approximation was presented. To overcome the second setback, an adaptive algorithm was presented and coupled to the filter to reduce the computational burden tremendously, while not sacrificing the accuracy of the estimates. For the out-of-surveillance case, an update algorithm exploiting the concepts of HMaps and SIMaps was introduced and coupled to the filter derived for the in-surveillance case. Simulations showed that incorporating SIMaps along with HMaps yielded more constrained pdfs over the case where HMaps were present alone in the update algorithm. This is particularly true whenever the search area exhibited a junction of road segments or equally hospitable terrain within the same neighborhood.

Further research is suggested to extend this work to the multitarget case. In addition, further research is suggested in the line of work of constructing SIMaps, particularly to construct and simulate SIMaps for the case where several favorable zones are present within the search area. Moreover, further research is contemplated in the line of work of integrating the HMap and SIMap concepts into the target kinematics model. In particular, the HMap can be integrated as a position-dependent transition function, whereas the SIMap can be encoded as the target's reactive decision policy (i.e., the control input as a function of state). Finally, the authors believe that the concept of limited-memory filters that are usually adopted whenever the SDS dynamics are not precisely known would be useful to implement and test within the optimal nonlinear Bayesian filter derived in this paper.

Acknowledgment

The authors would like to thank Dr. Jeff Layne from The Air Force Research Laboratory. His feedback and helpful discussions have added value to this research and is much appreciated. The authors would like also to thank the reviewers and the associate editor for their constructive suggestions that improved the presentation of this work.

References

[1] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: Wiley–Interscience, 2001.

[2] T. Kirubarajan, Y. Bar-Shalom, K. Pattipati, and I. Kadar, "Ground target tracking with variable structure IMM estimator," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 1, pp. 26–46, Jan. 2000.

[3] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filtering for online nonlinear/non-gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[4] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002.

[5] L. Stone, C. Barlow, and T. Corwin, *Bayesian Multiple Target Tracking*. Norwood, MA: Artech House, 1999.

[6] J. Layne, M. Eilders, Z. Kassas, and U. Ozguner, *Recent Developments in Cooperative Control and Optimization*, S. Butenko, R. Murphey, and P. M. Pardalos, Eds. Boston, MA: Kluwer, 2003, pp. 117–124.

[7] Z. M. Kassas, "An optimal nonlinear Bayesian filter design and combined hospitability and synthetic inclination approach for tracking and estimation," M.S. thesis, Ohio State Univ., Columbus, OH, 2003.

[8] K. Kastella and C. Kreucher, "Multiple model nonlinear filtering for low signal ground target applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 2, pp. 549–564, Apr. 2005.

[9] A. Challa and Y. Bar-Shalom, "Nonlinear filter design using Fokker–Planck–Kolmogorov probability density equations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 1, pp. 309–315, Jan. 2000.

[10] F. Daum, "Nonlinear filters: Beyond the Kalman filter," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 20, no. 8, pp. 57–69, Aug. 2005.

[11] D. Reid and R. Bryson, "A non-Gaussian filter for tracking targets moving over terrain," in *Proc. Annu. Asilomar Conf. Circuits Syst. Comput.*, 1979, pp. 112–116.

[12] P. Nougues, "Tracking intelligent objects in terrain," Ph.D. dissertation, Univ. Virginia, Charlottesville, VA, 1996.

[13] E. Sobiesk, J. Hamilton, J. Marin, D. Brown, and M. Gini, "Tracking multiple objects in terrain," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 1998, pp. 2842–2847.

[14] E. Sodtke and J. Llinas, "Terrain based tracking using position sensors," in *Proc. Int. Conf. Inform. Fusion*, 2001, vol. 2, pp. 27–32.

[15] C. Yang, M. Bakich, and E. Blasch, "Nonlinear constrained tracking of targets on roads," in *Proc. 8th Int. Conf. Inform. Fusion*, Philadelphia, PA, Jul. 2005, pp. 235–242.

[16] M. Ulmke and W. Koch, "Road-map assisted ground moving target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1264–1274, Oct. 2006.

[17] Z. Tang and U. Ozguner, "Pf-hmap: A target track maintenance approach for mobile sensor platforms with intermittent and regional measurements," in *Proc. 45th IEEE Conf. Decis. Control*, 2006, pp. 6757–6762.

[18] H. Sidenbladh and S. Wirkander, "Tracking random sets of vehicles in terrain," presented at IEEE Workshop Multi-Object Tracking, Madison, WI, 2003.

[19] F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *Proc. IEEE Proc. Aerosp. Conf.*, 2003, vol. 4, pp. 1979–1993.

[20] D. Fox, "KLD-sampling: Adaptive particle filters," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2001.

[21] N. Cui, L. Honga, and J. Layne, "A comparison of nonlinear filtering approaches with an application to ground target tracking," *Signal Process.*, vol. 85, no. 8, pp. 1469–1492, Aug. 2005.

[22] L. Hong, N. Cui, M. Bakich, and J. Layne, "Multirate interacting multiple model particle filter for terrain-based ground target tracking," *IEE Proc. Control Theory Appl.*, vol. 153, pp. 721–731, Nov. 2006.

[23] Z. Kassas, U. Ozguner, and J. Layne, "Out-of-surveillance target state estimation: A combined hospitability and synthetic inclination approach," in *Proc. Conf. Decis. Control*, Paradise Island, Bahamas, Dec. 2004, pp. 710–715.

[24] J. Priddy, "Stochastic vehicle mobility forecasts using the NATO reference mobility model," US Army Corps Eng., Washington, DC, Tech. Rep. 3, 1995.

[25] A. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic, 1970.

[26] K. Kastella, "Finite difference methods for tracking and automatic target recognition," in *Multitarget/Multisensor Tracking: Applications and Advances*, Y. Bar-Shalom and W. Blair, Eds. Norwood, MA: Artech House, 2000, ch. 5.

[27] B. Spencer and L. Bergman, "On the numerical solution of the Fokker–Planck equation for nonlinear stochastic systems," *Nonlinear Dyn.*, vol. 4, no. 4, pp. 357–372, 1993.
[28] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," in *Proc. IEE Radar Signal Process.*, 1993, vol. 140, pp. 107–113.
[29] M. El-Gebeily and H. E. Shabaik, "Approximate solution of the Fokker–Planck–Kolmogorov equation by finite elements," *Commun. Numerical Methods Eng.*, vol. 10, no. 10, pp. 763–771, 2005.
[30] S. Lototsky, R. Mikulevicius, and B. Rozovskii, "Nonlinear filtering revisited: A spectral approach," *SIAM J. Control Optim.*, vol. 35, no. 2, pp. 435–461, Mar. 1997.
[31] S. Challa and F. Faruqi, "Application of Chebychev's inequality theorem in the design of optimal non-linear filters," in *Proc. 1998 IEEE Int. Conf. Acoust., Speech, Signal Process.*, May, pp. 1277–1280.
[32] H. Stark and J. Woods, *Probability and Random Processes With Applications to Signal Processing*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.
[33] S. Challa, "Nonlinear state estimation and filtering with applications to target tracking" Ph.D. dissertation, Signal Process. Res. Center, Queensland Univ. Technol., Brisbane, Qld., Australia, Oct. 1998.
[34] "U.S. army material systems analysis activity, the Natick landform classification system," Tech. Rep., Aberdeen Proving Ground, MD, Tech. Rep. 100, Oct. 1974.
[35] "Terrain analysis procedural guide for surface configuration," Eng. Topographic Lab., U.S. Army Corp Eng., Fort Belvoir, VA, Tech. Rep., 1984.

Zaher M. Kassas (S'98–M'08) received the B.E. degree in electrical engineering from the Lebanese American University, New York, NY, in 2001, and the M.S. degree in electrical engineering from the Ohio State University, Columbus, OH, in 2003. He is currently working toward the Ph.D. degree from the University of Texas at Austin, Austin.

From 2002 to 2004, he was a Graduate Research Associate with the Collaborative Center of Control Science, Ohio State University, where he was engaged in designing nonlinear estimation filters for uninhibited autonomous vehicles. He is currently a Research and Development Engineer with the Control Design and Simulation Group, National Instruments, Austin. He is also an Adjunct Faculty at Texas State University. His current research interests include filtering and estimation, stochastic control, ITS, and embedded systems.

Ümit Özgüner (S'72–M'75–SM'09) received the Ph.D. degree from the University of Illinois at Urbana-Champaign, Urbana-Champaign.

He was with I.B.M., University of Toronto, and Istanbul Technical University. Since 1981, he has been with the Ohio State University, where he is a Professor of electrical and computer engineering. His current research interest include ITS, decentralized control, and autonomy in large systems. He is the author of over 400 publications.

Dr. Özgüner was the first President of the IEEE ITS Council in 1999, and currently, he is the ITS Society Vice President for Conferences and TRC Inc. Chair on ITS. He was also a member of the IEEE Control Society. He has organized many conferences and was the General Chair of the 2002 CDC, IV 2003, and ICVES 2008. He has coordinated the teams that participated successfully in the 1997 Automated Highway System Technology Demonstration, the Defense Advanced Research Projects Agency (DARPA) 2004 and 2005 Grand Challenges, and 2007 Urban Challenge.