Measurement-Guided Likelihood Sampling for Grid-Based Bayesian Tracking

JASON M. AUGHENBAUGH
BRIAN R. LA COUR

A grid-based Bayesian tracking approach is proposed that uses the observed measurements to guide the sampling of the likelihood function during the measurement update step. This leads to computational savings over standard sampling methods while also providing a more accurate estimate of the likelihood function. The likelihood model assumes an exponential distribution of returns with a mean based on a predictive model that incorporates an assumed signal-to-noise ratio (SNR) of the targets, background clutter, beam response, and waveform ambiguity functions. Two variations of an example based on simulated frequency modulated (FM) and continuous wave (CW) signals are used to assess target detection, localization, and computational performance.

Manuscript received January 5, 2010; revised June 9, 2010; released for publication July 9, 2010.

Refereeing of this contribution was handled by Peter Willett.

This work was supported under United States Office of Naval Research Contract No. N00014-06-G-0218-01.

Authors' addresses: Applied Research Laboratories, University of Texas, PO Box 8029, Austin, TX 78713-8029, E-mail: {jason, blacour}@arlut.utexas.edu.

1557-6418/10/$17.00 © 2010 JAIF

1. INTRODUCTION

Bayesian inference is recognized as a general framework for performing optimal target tracking. Fundamentally, it assumes that the uncertainty in our knowledge of the state of the target (or targets) may be well represented by probabilities. Bayes' theorem then provides the basic mechanism whereby measurements update these probabilities and, hence, our knowledge of the target state.

For computer implementation of a Bayesian scheme, a representation of the probabilities must be selected. Various approaches have been developed, including Kalman filters, grid-based models, and particle filters, as summarized in [1, 35]. Existing approaches are valuable in a diverse set of applications, but there is room for improvement in other applications.

The application area driving this research involves the goal of detecting and localizing a single target in a very loud environment, such as an active sonar system trying to detect and track a quiet target in a cluttered, reverberant environment. The undersea active sonar presents a rich diversity of contextual information, which can be vital for situational awareness, but too often is ignored by automated tracking and classification systems.

In order to incorporate such details in the tracker, we pursue a track-before-detect paradigm. In this approach, the normalized matched filter output of the signal processing chain is incorporated directly into the tracker, as opposed to a contact-level approach in which clustered data is used. At this lower level in the signal processing, more information should be available. In order to keep the data load manageable, the matched filter output is thresholded. This thresholding, as well as details of the waveform ambiguity functions and beam patterns, are folded directly into the likelihood functions used in the Bayesian tracker.

The form of these functions, which is described in Section 3, requires a detailed sampling of the likelihood function. We propose an advanced grid-based approach to Bayesian tracking in which the likelihood evaluations are performed using an intelligent sampling procedure.

Previous work on Bayesian tracking is described in Section 2. The mathematical models used for our tracking applications are described in Section 3. The advanced implementation of the measurement update, which is the core contribution of this paper, is described in Section 4. The example problems and results comparing the proposed measurement update to standard implementations are given in Section 5. Additional discussion is given in Section 6, and a brief summary closes the paper in Section 7. An appendix contains a derivation of the likelihood function.

2. CONTEXT AND OVERVIEW OF GRID-BASED METHODS

Most early target tracking algorithms were based on the Kalman filter, which can be derived via either leastsquares optimization or as a special case of Bayesian filtering [14]. The Kalman filter represents an exact solution to the Bayesian filtering problem under the conditions of linearity in the relationship between the state and the measurements, linearity in the motion update, Gaussian errors in the measurements, and Gaussian process noise in the motion updates. Various modifications and approximations have been made to relax these assumptions, such as the extended Kalman filter (see the edited collection of papers in [33]) and the unscented Kalman filter [17]. Other early Bayesian methods are summarized in [29]. Non-parametric approaches, such as the grid-based methods described in the following, move away from any assumptions of linearity and normality.

2.1. Overview of Grid-Based Methods

As early as 1971, researchers suggested non-parametric models using point masses on a rectangular grid, but computational limitations prohibited any realistic implementations for continuous problems [7, 11]. However, when the target state is inherently discrete, a discrete tracker is optimal. Early examples include the Baum-Welsh filter [28] and dynamic programming approaches, such as those based on the Viterbi algorithm [4, 5, 40]. The goal of the dynamic programming approaches is to find the most likely path through the state space over time. While still based on hidden Markov models, these approaches are not Bayesian in nature and are based instead of maximum likelihood modeling. These methods are extended to continuous state estimation problems by using a discretized approximation [16, 38].

By 1987, computational power had increased sufficiently for Kitagawa to resurrect the idea of direct numerical approaches for modeling posterior probability distributions in Bayesian parameter estimation problems, specifically suggesting the use of piece-wise constant approximations to the density function across a set of defined nodes, or essentially over a grid [19]. Kitagawa also suggests the potential for adaptive grids, moving grids, and higher order models. Around the same time, Kramer and Sorenson implemented their own piecewise constant Bayesian estimator and compared its results to both Kalman filter and point-mass approaches [20, 21], showing the superiority of the grid based method for a particular system identification problem.

Another approach that uses a grid is the histogram probabilistic multi-hypothesis tracker (H-PMHT) [36]. However, this is not directly a Bayesian filtering approach. Rather, the grid is used to aggregate measurements into weightings based on received power in a particular grid cell, and the weightings for each cell are then used to form synthetic measurements and measurement error covariances that are handed to the PMHT [37], which is based on point measurements. The advantage of this approach is that the computational costs of the likelihood evaluations are saved, and the disadvantage is that the detailed structure of the measurements is not leveraged for better localization.

The first prominent use of a piecewise constant approximation over an adaptive grid in Bayesian target tracking was by Stone et al. [34, 35]. This approach and variations have performed well in a variety of Bayesian target tracking applications, such as [23, 32, 35]. The proposed research is an extension of these approaches.

2.2. Particle Filters

An alternative implementation for Bayesian tracking is a particle filter (see a summary in [1]). Early particle-based approaches such as the boot-strap filter [13] suffered from the limitation that the updating of particle positions was done without regard to the current observations, potentially leading to important regions of the state space not having enough particles to capture the new information. Later approaches sought to remedy this by using importance sampling to redirect some particles to areas with potentially high likelihoods [12]. Many other variations of the measurement update algorithm have been proposed and used in a wide variety of problems [22, 26, 27, 30, 31].

While such methods have been used successfully for many applications, they are not the most effective for applications in which the likelihood function has very fine structure. In these applications, it is not sufficient just to place particles near peaks; the particles must also be placed in particular locations near the peaks. This motivation is relevant in both particle filters and grid-based methods, although this paper will focus on the implementation only in a grid-based tracker.

2.3. Track-before-Detect Paradigm

In many trackers, the sensor data is processed extensively in order to extract a small number of contacts. A process of data association relates each contact to an existing or proposed track. A particular tracking algorithm, such as a Kalman filter or a particle filter, is then used to update the tracks. A related track management scheme is used to initiate and drop tracks. One drawback of these approaches is their poor performance when trying to track a target that is barely detectable. For these scenarios, a paradigm of track-before-detect has been introduced (see the special issue introduced by [6], or the comparisons of methods in [10, 30]). In these approaches, the data is not aggregated into contacts, but rather is used in a form closer to a full map or image of the sensor data.

2.4. Mathematics of Grid-Based Methods

In Bayesian tracking, the uncertainty in the state of the target (or targets) is represented by a probability density function (PDF) for a continuous state, or a probability distribution for a discrete state. There are two main steps in this tracking procedure. The first is the *measurement update* (also known as the filter step) that takes the prior PDF and incorporates the information from a measurement into it using Bayes' rule. The second is a *motion update* (also known as the system update or the prediction step) that takes the current estimate and "moves" it forward in time. The purpose of the motion update is to account for the evolution of the state (in this case due to the target velocity) over time between measurements.

In a grid-based representation of a PDF, the state space is discretized into a multidimensional grid [3, 15, 23, 35, 39]. For a continuous state space, the grid represents a piece-wise constant approximation of the PDF.

Beginning with the continuous form of the measurement update for a Bayesian filter, we illustrate a basic grid-based approach. We define the posterior density $\rho_n(\mathbf{s} \mid \mathbf{y_n})$ over the state $\mathbf{s}$, meaning posterior to receiving the nth measurement $\mathbf{y_n}$. The motion updated prior is denoted $\rho_n^-(\mathbf{s})$, and the likelihood is given by $L_n(\mathbf{y_n} \mid \mathbf{s})$. The posterior density is found using Bayes' rule as

$$\rho_n(\mathbf{s} \mid \mathbf{y_n}) = \frac{L_n(\mathbf{y_n} \mid \mathbf{s})\rho_n^-(\mathbf{s})}{\int L_n(\mathbf{y_n} \mid \mathbf{s}')\rho_n^-(\mathbf{s}')d\mathbf{s}'}. \qquad (1)$$

In the discretized version, we will refer to each grid cell C_i, which contains a set of states. In this formulation, the density is approximated as $\rho_n(\mathbf{s}) \approx \sum_i p_{i,n}\mathbf{1}_{C_i}(\mathbf{s})$, where $\mathbf{1}_{C_i}(\mathbf{s})$ is an indicator function equal to 1 when $\mathbf{s} \in C_i$ and zero otherwise, and $p_{i,n}$ is the constant value across the ith grid cell. In a piece-wise constant grid-based method, the likelihood function is integrated appropriately to capture the local behavior over a cell, such that letting $p_{i,n}^-$ be the motion updated prior probability value for cell i then Bayes' rule becomes

$$p_{i,n} = \frac{p_{i,n}^- \int_{C_i} L(\mathbf{y_n} \mid \mathbf{s}')d\mathbf{s}'}{\sum_k p_{k,n}^- \int_{C_k} L(\mathbf{y_n} \mid \mathbf{s}')d\mathbf{s}'}. \qquad (2)$$

An important note is the use of the integral in the numerator, which means the value in the grid cell is updated considering the entire local behavior, not just a point estimate. This increases the accuracy of the piece-wise constant approximation. However, the evaluation of this integral can be challenging, as discussed in Section 4. The efficient and accurate approximation of this integral is the focus of this article.

2.5. Motion Updates for Grid-Based Methods

The purpose of the motion update is to account for the evolution of the state over time. Even if the state is perfectly measured at time t_1, the state is uncertain at future time t_2 due to process noise. By making assumptions about the target's motion, one can predict the state at some time in the future. In traditional grid-based methods, the motion update is computationally expensive as it involves integration with a Markov kernel, which is nominally a costly $O(N^2)$ operation, where N is the number of grid cells.

The basic motion model that we use is described in [25] and summarized as follows. The target is taken to follow an Integrated Ornstein-Uhlenbeck (IOU) process [35]. Rather than applying a Markov kernel to evaluate the state transitions [18, 35], we instead draw inspiration from particle filters. The idea is to model the actual transitions using particles by reversing the way the problem is viewed—instead of focusing on the transitions *to* each cell, we focus on the transitions *from* each cell. The resulting algorithm has complexity of $O(MN)$, when M is the number of particles, and generally $M \ll N$. Ongoing research is comparing the performance of different motion updates for grid-based methods, but this model has been used successfully in other work [2] and is accurate and efficient enough for its use in this paper.

2.6. Grid Cell Mesh Size

Central to the success of a grid-based method is the definition of the grid, or mesh. This defines the boundaries of each cell, and thereby the volume contained in cell. Large grid cells lead to lower computational costs (since larger grid cells mean fewer cells), but at the cost of less precision in the posterior estimate. For example, assume the position grid cells are 5 km by 5 km and are referenced using the coordinates of their center. When using the maximum *a posteriori* (MAP) estimate, the target location can at best be known to within ±3.5 km of distance because the MAP estimate is the grid cell with the highest posterior probability, but the actual state can be anywhere within the cell. Smaller grid cells lead to a higher precision, but at higher computational costs.

Three main considerations drive mesh selection, whether constant or locally adaptive. The first is the desired resolution in target localization from the PDF. The second is the structure of the likelihood function. The final consideration is the trade-off between localization and computational cost.

Ideally, one would use a resolution that refines the localization to a level that is strategically or tactically relevant. For example, if the user only needs to refine the position to a 1 km by 1 km block, then a mesh size of 10 m by 10 m is wasteful. Conversely, if the user requires 1 km by 1 km resolution, then a 10 km by 10 km mesh does not suffice. However, the properties of the likelihood function also dictate the scale.

A straightforward implementation of a grid-based method would be to refine the grid until the likelihood function is reasonably well modeled by the resultant piecewise constant model. In our applications, the scale of the variations in the likelihood function can be quite small in the context of the state space, and the grid would need to be made very fine in many regions in order to capture the necessary detail. Also, the overall uncertainty in the posterior distribution dominates this scale, making fine detail at the PDF level unnecessary.

In the approach described in Section 4, a balance is struck whereby the PDF is modeled at a tactically relevant resolution, but the integration of the likelihood function is approximated using an intelligent, measurement-guided numerical integration. The idea of the intelligent approach is to base the sampling of the likelihood on its known properties and the received measurements. One can then focus the sampling on the areas that contain the most information. This detailed sampling is only used to evaluate the integral of the likelihood over each grid cell. By contrast, adaptive grid methods refine the state space itself. This could achieve the necessary resolution in likelihood evaluation, but at a higher computational cost due to the high resolution being carried forward to subsequent measurement and motion updates.

3. APPLICATION DESCRIPTION

Consider a Bayesian tracking scheme for which the state space consists of the number of targets present (either zero or one) and the target's kinematic state, $\mathbf{s}$, given that it is present. Let P_n be the probability that a single target is present in a particular region of interest, and let $\rho_n(\mathbf{s})$ be the posterior probability density function after the nth measurement is incorporated. (Note that $n = 0$ corresponds to the prior distribution.) Two likelihood functions are relevant. The target likelihood function $L_n(\mathbf{y_n} \mid \mathbf{s})$ denotes the likelihood of receiving measurement $\mathbf{y_n}$ given that the target is in state $\mathbf{s}$. The clutter likelihood function $L_n(\mathbf{y_n} \mid \emptyset)$ denotes the likelihood of receiving measurement $\mathbf{y_n}$ when there is no target present. Measurement updates on the kinematic PDF are performed using Bayes' theorem, which for notational purposes we rewrite as

$$\rho_n(\mathbf{s}) = L_n(\mathbf{y_n} \mid \mathbf{s})\rho_n^-(\mathbf{s})/E_n \tag{3}$$

where the partial Bayesian evidence E_n for a target present is given as

$$E_n := \int L_n(\mathbf{y_n} \mid \mathbf{s})\rho_n^-(\mathbf{s})d\mathbf{s}. \tag{4}$$

Similarly, the target probability P_n, meaning the probability that a target is present in the modeled state space, is updated by

$$P_n = \frac{E_n P_n^-}{(1 - P_n^-)L_n(\mathbf{y_n} \mid \emptyset) + P_n^- E_n} \tag{5}$$

where P_n^- is the motion updated target probability, as derived in [24]. Additional description of the motion model and birth/death process used in the tracker is provided in [25].

3.1. Likelihood Functions

The form of the likelihood function is an important aspect of the problem. We choose a model that is based closely on the actual signal processing, as described in more detail in [3]. The model is part of a track-before-detect paradigm using the normalizer output. We define the likelihood model in terms of measured signal-to-noise (SNR) values from the normalized matched filter output of a standard active signal processing chain. Both frequency modulated (FM) and continuous wave (CW) transmit waveforms are considered so that, in general, each SNR value (in units of squared amplitude) z_k is associated with a particular echo time of arrival (TOA) τ_k, angle of arrival (AOA) ϕ_k, and (for CW) Doppler frequency shift ν_k. Each measurement $\mathbf{y}$ consists of K returns y_k such that $\mathbf{y} = (y_1, \ldots, y_K)^{\mathrm{T}}$, where for a CW waveform $y_k = (z_k, \tau_k, \phi_k, \nu_k)$, and for an FM waveform $y_k = (z_k, \tau_k, \phi_k)$.

It is common to model responses such that a target can result in measurements in a neighborhood of the true location [8, 31]. In a basic model, the fluctuations of the SNR values about the means are assumed to be independent and follow an exponential distribution (or more generally, one could use the generalized Pareto distribution), so the likelihood function for the overall measurement $\mathbf{y}$ is

$$L(\mathbf{y} \mid \mathbf{s}) = \prod_{k=1}^{K} \frac{1}{\mu_k(\mathbf{s})} \exp(-z_k/\mu_k(\mathbf{s})) \tag{6}$$

where $\mu_k(\mathbf{s})$ represents the mean SNR that one would expect to receive from measurement element y_k (e.g. the SNR z_k at τ_k and ϕ_k for an FM source signal), given that the target state is $\mathbf{s}$. The calculation of $\mu_k(\mathbf{s})$, which models details of the signal processing chain such as the beam response, is described in Section 3.2. The assumption of independence is valid for measurement spaces that are appropriately constructed to reflect the actual sensitivity of the sensors and signal processing. For numerical stability, all likelihood calculations are actually implemented using logarithms of the likelihood values. This substantially reduces the chances of underflow due to the many small values potentially multiplied in (6).

We choose to use a relatively low threshold for the measurements in order to reduce the computational cost without decreasing detection performance. Accordingly, only individual elements with SNRs that exceed a set threshold η are incorporated in to the measurement. Let $\mathbf{k} = (k_1, \ldots, k_I)$ denote an ordered sequence of indices corresponding to these threshold crossings. If there are no threshold crossings, then $\mathbf{k} = \emptyset$. A derivation and statement of the likelihood function for such measurements is provided in the Appendix.

3.2. SNR Predictive Modeling

Under the hypothesis that no target is present and there are no persistent clutter objects, we assume a uniform clutter background for all points in measurement space, i.e.,

$$\mu_k(\emptyset) = \sigma_0^2. \tag{7}$$

If a single target is presumed to be present, then

$$\mu_k(\mathbf{s}) = \mu_k(\emptyset) + \sigma_{\mathrm{T}}^2 h_k(\mathbf{s}) \tag{8}$$

where σ_T^2 is the target SNR and $h_k(\mathbf{s}) \in [0,1]$ is the response function, specifically (using CW as an example) the product of the array beam response $b(\phi)$ and waveform ambiguity function $\chi_{CW}(\tau,\nu)$ for a target in state $\mathbf{s}$. Note that for this equation, the SNR should be given in units of amplitude squared and not in dB.

The beam response, as modeled here, is given by

$$b(\phi) = \mathrm{sinc}(\phi/\Delta\phi)^2 \tag{9}$$

where $\phi = \phi(\mathbf{s}) - \phi_k$, $\phi(\mathbf{s})$ is the bearing to the hypothesized target state, and $\Delta\phi$ is the nominal beam width, which is set to 0.0873 radians (i.e., 5 degrees). In practice, the actual beam pattern of the beam forming method used would be modeled here.

For the CW waveform, the ambiguity function for $\tau \leq T$ is

$$\chi_{CW}(\tau,\nu) = \left(1 - \frac{|\tau|}{T}\right) \mathrm{sinc}\left(\nu T \left(1 - \frac{|\tau|}{T}\right)\right) \tag{10}$$

where T is the pulse length, $\tau = \tau(\mathbf{s}) - \tau_k$, $\nu = \nu(\mathbf{s}) - \nu_k$, and $\tau(\mathbf{s})$ and $\nu(\mathbf{s})$ are respectively the TOA and Doppler shift corresponding to the hypothesized target state $\mathbf{s}$. For $\tau > T$, $\chi_{CW}(\tau,\nu) = 0$. For a linear FM waveform, it is given by

$$\chi_{FM}(\tau,\nu) = \chi_{CW}(\tau, \nu - \tau B/T) \tag{11}$$

where B is the bandwidth. Combining these results, we have the response function

$$h_k(\mathbf{s}) = b(\phi(\mathbf{s}) - \phi_k)|\chi(\tau(\mathbf{s}) - \tau_k, \nu(\mathbf{s}) - \nu_k)|^2. \tag{12}$$

As described in [3], one can also include known clutter objects (such as persistent bathymetric features or tracked merchants ships) in the model, though this is not included in this paper. The predictive SNR modeling also accounts for detection ranges and blanking regions.

3.3. The Measurement Space

The measurements are based on the normalized matched-filter output of the signal processing chain. A consequence of this is that the time, bearing, and Doppler measurements are discretized. For example, if there are 72 beams, then $\phi_k \in \{0°, 5°, 10°, 15°, \ldots, 355°\}$. In the remainder of the paper, frequent mention will be made to the measurement space. This refers to the grid on which the SNRs are measured, meaning the set of triples (τ_k, ϕ_k, ν_k) for CW and the set of pairs (τ_k, ϕ_k) for FM.

4. MEASUREMENT UPDATES

In a grid implementation, the measurement update appears trivial: it simply involves a point-wise-multiplication of prior grid cells with the value of the likelihood function in each grid cell. However, the value of the likelihood function associated with a particular grid cell is the integral of the likelihood function over that grid cell, as in (2).

In order to properly perform the measurement update, one must approximate this integral. The complexity of this operation depends on the complexity of the likelihood function and its scale relative to the grid cell. For example, if the likelihood is reasonably constant over a grid cell, then the integral may be reasonably approximated by a few samples—perhaps even a single one—within the grid cell. However, the likelihoods for the applications of interest are generally not so well behaved, and a more detailed sampling is necessary.

4.1. General Motivation for Likelihood Sampling method

The basic motivation for this method stems from three things. First, the set of possible measurements generally reflects inherent information about the structure of the response function. Second, there is a natural mapping from the state space to the measurement space. Finally, points in the state space that are "far" from any measurement are essentially unaffected by that measurement.

4.1.1. Measurement Bins Reflect Response Structure

The key assumption made in this method is that the measurement bins are appropriate for the underlying signal processing, and therefore they reflect the approximate scale of ambiguity functions. Thus, they provide a good starting point for defining the likelihood sampling. For example, if the measurements are divided into bins of bearing that are 5° wide, then it is implied that the resolution of the sensor is approximately that fine. It would therefore be incorrect to sample the likelihood function only every 10° because one would be missing states that are perfectly aligned with other measurements.

At the same time, there is structure between the beams that can affect the likelihood function. For example, if there is one beam centered at 15° and another centered at 20°, a target with bearing 17° would affect each beam differently. One needs to sample more finely than the beam width to capture such details. The exact resolution may depend on other features of the problem, and some customization and tuning of the sampling will be necessary to adequately capture the properties of the ambiguity functions.

Such a method can be used for time-delay-only measurements, bearing-only measurements, combined time-and-bearing measurements, and full time-bearing-Doppler measurements. We address the time-and-bearing (Section 4.2) and time-bearing-Doppler (Section 4.3) cases in this paper.

4.1.2. Mapping between Measurement Space and State Space

The relationship between points in measurement space and points in state space is simplest for TOA and AOA measurements. A particular TOA and AOA pair defines a specific point in position; the mapping depends only on the states of the source and receiver, as follows. Let c denote the speed of sound, x_T and y_T denote the target position, x_R and y_R denote the receiver position, x_S and y_S denote the source position, and D denote the distance from the source to the receiver. For a received time delay τ and bearing ϕ corresponding to a target, one finds

$$x_T = x_R + R\cos(\phi) \tag{13}$$

$$y_T = y_R + R\sin(\phi) \tag{14}$$

where the range from the receiver to the scatterer is given in [9] as

$$R = \frac{c\tau}{2}\frac{1-\left(\frac{1}{c\tau}D\right)^2}{1-\frac{1}{c\tau}D\cos(\phi-\psi)} \tag{15}$$

where

$$\psi = \arctan\left(\frac{y_S-y_R}{x_S-x_R}\right). \tag{16}$$

This makes the measurement updates for FM source waveforms independent of velocity, saving considerable computational expense, as will be described in Section 4.2.

The addition of Doppler information with CW source waveforms complicates the mapping. Let unit vector $\hat{u}_{ST}$ point from the source to the target, and let unit vector $\hat{u}_{RT}$ point from the receiver to the target. Let $\vec{V}_T = (\dot{x}_T, \dot{y}_T)$, $\vec{V}_S = (\dot{x}_S, \dot{y}_S)$, $\vec{V}_R = (\dot{x}_R, \dot{y}_R)$ denote the velocities of the target, source, and receiver respectively. Then the observed Doppler shift for source frequency f_0 found from [41] is

$$\nu = \left(\frac{c-\hat{u}_{ST}\cdot\vec{V}_T}{c+\hat{u}_{RT}\cdot\vec{V}_T}\right)\left(\frac{c+\hat{u}_{RT}\cdot\vec{V}_R}{c-\hat{u}_{ST}\cdot\vec{V}_S}\right)f_0 - f_0. \tag{17}$$

The solution to (17) results in a line of ambiguity in target velocity with slope m given by

$$m = \frac{-(R^{-1}a_x + D^{-1}c_x)}{R^{-1}a_y + D^{-1}c_y} \tag{18}$$

and y-intercept b_y given by

$$b_y = \frac{\frac{c\nu}{f_0} - R^{-1}(a_x\dot{x}_R + a_y\dot{y}_R) - D^{-1}(c_x\dot{x}_S + c_y\dot{y}_S)}{-(R^{-1}a_y + D^{-1}c_y)} \tag{19}$$

where $a_x = x_R - x_T$, $a_y = y_R - y_T$, $c_x = x_S - x_T$, and $c_y = y_S - y_T$.

This complicates the sampling of the likelihood function, as discussed in Section 4.3.

4.1.3. Measurement-Guided Likelihood Sampling

In order to perform the measurement update, one needs to evaluate the integral of the likelihood function in every grid cell. In general, this could be computationally expensive. For example, one might choose the naïve approach of randomly sampling 1000 points within each grid cell. However, this would be wasteful (not to mention computationally infeasible), as that level of detail is not needed in every cell. An adaptive method that places more samples where they are needed and fewer samples where acceptable would be much more efficient. Fortunately, such a method can be constructed for this application.

Based on the functions described in Section 3.1, the likelihood of getting a measured time delay of 30 s when the target is actually at a point corresponding to 10 s is low (assuming only direct-path signals; multipath modeling would require a more complex ambiguity function). Similarly, large bearing errors are very unlikely (unless the target is very loud and side-lobing). The combination of a large bearing error and a large time delay error is even less likely. Hence, when evaluating $L(\mathbf{y} \mid \mathbf{s})$ in Bayes' rule, the values are essentially zero unless $\mathbf{y}$ is "near" $\mathbf{s}$, where "near" is loosely defined in terms of the distance between the measurement and the mapping of the hypothesized state space into the measurement space.

This concept can be used to determine where to expend computational resources for sampling the likelihood function. An example response function $h_k(\mathbf{s})$ for an FM signal is shown in Fig. 1.

In the figure, the scales of the state space, likelihood function, and measurement space are different from those used in the actual modeling in order to highlight the structure. Additionally, the ambiguity function for time delay of arrival is defined to fall off exponentially instead of with a *sinc* function to highlight the beam behavior. The height of the surface reflects the SNR (in dB) at a given point in the position state space. The white dots represent the discretized points in state space at which actual measurements can occur.

It is apparent that large regions of the xy-plane have a response of zero, which leads to essentially zero likelihood. There is no reason to sample the likelihood densely in these regions, as one can directly define the likelihood as zero in these regions and make no evaluations. This saved computational effort can then be invested in sampling the non-zero region more densely. The sampling scheme (and areas with no sampling) can be determined as described in the following sections.

4.2. Bearing and Time-Delay Measurements

We first restrict ourselves to the FM case, in which there is no Doppler information, and hence no velocity dependence. Therefore, one can work entirely in the xy-plane, which saves considerable computational expense. To get back to the 4-dimensional state space, one must replicate the likelihood values in the xy-grid across the velocity dimensions of the grid, which is a trivial operation.

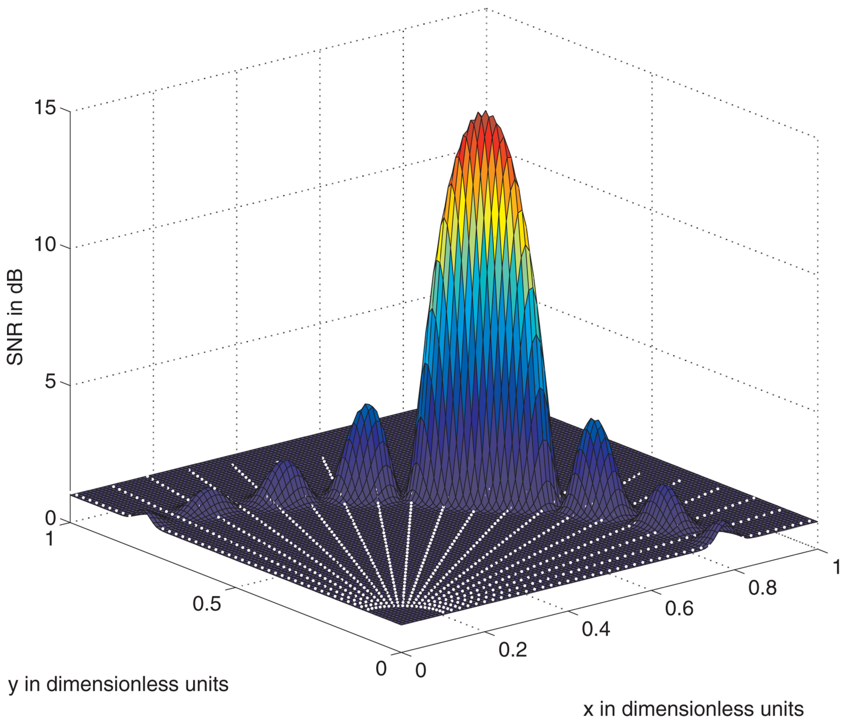

Fig. 1. Example response function for FM waveform.

4.2.1. Identifying Important Regions of State Space

We direct this discussion around one particular measurement $y_k = (z_k, \tau_k, \phi_k)$. This can be mapped to Cartesian coordinates using (13) and (14). We then can determine an approximate range from the hypothesized target state $\mathbf{s}_t = (x_t, y_t)$ around this measurement in which the likelihood $L(y_k \mid \mathbf{s}_t)$ is non-negligible. Essentially, the target will only appear to reflect acoustic energy at a hypothesized point (x_h, y_h) if the target location $\mathbf{s_t} = (x_t, y_t)$ is near it.

A contour plot of the example response from Fig. 1 is shown in Fig. 2. Each discrete measurement point is shown with a small black ×. Two different sets of proposed sampling points are labeled, one with larger red ×s, and one with green circles. The red set clearly covers the regions of the response that have non-negligible values and serves as example of a sampling scheme that samples more finely than the measurements in bearing. The green set is a more aggressive approach to saving computational effort, as it only samples the highest peak. The selection of an exact set will be problem-dependent.

Revisiting Fig. 1, the secondary peaks are at 4.3 dB, considerably lower than the primary peak of 15 dB. For the types of applications that we are pursuing, this is a substantial drop-off, and almost always would fall below the threshold level. For larger SNR targets, the side lobes may become more important. A similar relationship exists with the ambiguity functions used in the actual examples, which differ in structure from the simplified example in this section. The sensitivity to the selection of a sampling region is discussed in Section 5.4.

4.2.2. Selecting Specific Sample Points

The preceding discussion focused on selecting the region of the state space to sample. The secondary question is which actual points within that region to sample. There are two factors that help to guide this decision. The first is a sampling based argument similar to the Nyquist criterion. Essentially, we want to sample densely enough to capture the general form of the function. We do not need to meet a strict Nyquist criterion, because the goal of the sampling is to approximate the integral of the likelihood function over the region of interest, rather than reconstructing the function. The appropriate rate will depend on the ambiguity functions and beam pattern of the specific problem, but in most cases it is sufficient to sample every beam and time step that is a possible measurement, as well as two or three samples in between (assuming the measurement bins appropriately reflect the underlying signal processing).

The second requirement stems from the relationship between the state space, the likelihood function, and the grid. Particularly in regions far from the source and receiver, the measurements can be relatively sparse in state space due to spreading of the beams in Cartesian space as distance from the receiver increases. For example, if beams are 5 degrees apart, then at a distance of 25 km the beams are roughly 2 km apart. If the grid resolution is such that each grid cell has position sides of length 1 km, then a cell could fall entirely in between measurement. Consequently it is necessary to ensure explicitly that each grid cell is sampled, although the sampling can

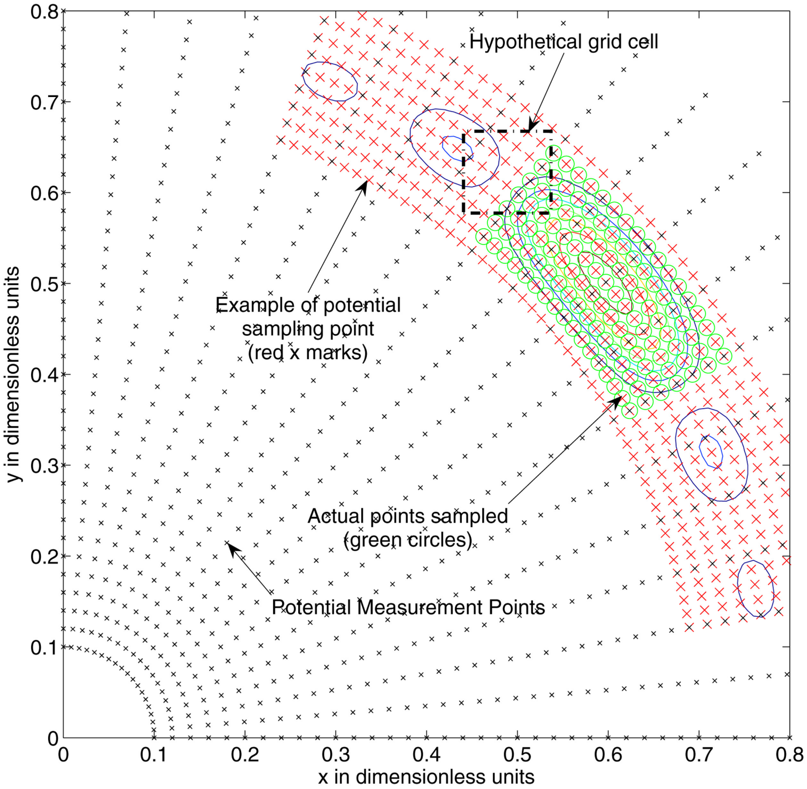

Fig. 2. Intelligent sampling example for FM waveform.

be relatively sparse because the original lack of samples in such a cell reflects the assumption that the likelihood is relatively smooth and flat there.

At a distance of 2 km, the beams are roughly 200 m apart. Thus at short distances, the state space needs to be sampled much more finely. This is another motivation for basing the sampling scheme on the measurement space (rather than the state space): the measurement space inherently reflects the necessary detail across the entire space, whereas the necessary detail varies as one moves across state space.

Once the relevant sampling area is determined for a given problem, an offset template is created. The template defines the points in the neighborhood of a threshold exceedance that should be sampled. These points are defined using constant step sizes $\Delta\tau$ in time and $\Delta\phi$ and bearing and the number of steps in either direction from the threshold exceedance. Once a generic template is defined, it can be reused for each threshold exceedance. The result is a list of points that are important to sample in measurement space, which can be mapped directly to points in Cartesian state space. Note that points that are affected by more than one threshold exceedance are still only sampled once, so duplicates are removed.

One may contemplate using more advanced schemes, such as making a non-constant template (e.g. one that is SNR dependent) or a non-symmetric template (e.g. one that has different step-sizes and/or ranges in the positive and negative directions). These options are not considered in this paper.

4.2.3. Handling Regions that are Not Sampled Explicitly

We consider now a single cell. If the cell does not contain any sample points, then the likelihood function is given a default value for that cell. This default value is essentially the value of the likelihood function at a point very far from any observed threshold exceedance. If the cell does contain sample points, then the sampling is planned as described in the preceding section.

The sampled points may not cover the entire grid cell. For example, consider the hypothetical grid cell labeled in Fig. 2. Assume that the green circles represent the points that are actually sampled. The grid cell also contains many un-sampled points that fall on the potential sampling grid (those with just the red ×s). Because the quantity of interest is the integral of the likelihood function over the grid cell, one must account for the sampled regions and the unsampled regions.

Defining the sampled area as the region of interest (ROI) in a particular grid cell C_i and assuming uniform sampling, the approximation of the integral of the likelihood function requires all of these points, but it can be broken into two summations as

$$L_i(y_k \mid \mathbf{s} \in C_i) = \frac{1}{N}\left(\sum_{\mathbf{s_k}\in \mathrm{ROI}} L(y_k \mid \mathbf{s_k}) + \sum_{\mathbf{s_k}\notin \mathrm{ROI}} L(y_k \mid \mathbf{s_k})\right). \tag{20}$$

TABLE I
Grid Data Structure

Cell	x	y	v_x	v_y
1	10	10	0	0
2	20	10	0	0
3	10	20	0	0
4	20	20	0	0
5	10	10	1	0
6	20	10	1	0
7	10	20	1	0
8	20	20	1	0
9	10	10	0	1
10	20	10	0	1
…	…	…	…	…

For potential sample points in the ROI, the likelihood function is evaluated directly. For points outside of the ROI, the default value is used. This replaces the second summation on the right with a constant times an integer (the number of points in the cell but not in the ROI), which yields large computational savings when most points in the cell fall outside a ROI. In practice, a more complex form of (20) should be used that includes a Jacobian term for mapping between the measurement space and state space.

4.2.4. Additional Implementation Details

In order to improve performance of the approach, additional optimizations have been made. First, the number of potential samples in each grid cell is precomputed (when possible) for all grid cells to facilitate the calculation of (20).

This is highly efficient when the sensors are static, but because this relationship changes if the sensors move, it is not always possible. In cases in which the sensors are moving, it is suboptimal because time might be spent calculating this relationship for cells for which it its unnecessary (i.e. cells that contain no ROIs).

Second, a constant, uniform (per dimension) grid is used. Specifically, each 4-dimensional grid cell has sides of length of Δx, Δy, Δv_x, and Δv_y. The grid cells are identified according to the state at their center, and a data structure is created that, in matrix notation, contains columns corresponding to the x, y, v_x, and v_y dimensions, where each row is a different grid cell. The rows are sorted first by Δv_y, then by Δv_x, then by Δy, and finally by Δx.

Using a dimensionless example, if the possible center speeds are 0 and 1 and the possible center positions are 10 and 20, then the grid data structure has the first 10 rows as in Table I. We are assuming that $\Delta x = 10$, $\Delta y = 10$, $\Delta v_x = 1$, and $\Delta v_y = 1$.

Using this format, an efficient mapping can be made from states to grid cells. We introduce the method with a specific example. Assume we want to the grid cell that contains the state (13, 21, 1.4, 0.4). First consider the x-coordinate and determine in which sub-grid cell (i.e. which x-grid cell) it falls. There are two x-grid cells: one with center 10 and one with center 20, with respective bounds of [5,15] and [15,25]. Clearly it falls within the first cell, but mathematically this can be found as

$$x_{\mathrm{id}} = \lfloor (x - x_{\min})/\Delta x \rfloor + 1 \tag{21}$$

where $x_{\min}$ is the minimum in the x-direction (in this case 5), and the $\lfloor . \rfloor$ operator indicates rounding down to the nearest integer. So for example, in this case with $x = 13$, one finds: $\lfloor (13-5)/10 \rfloor + 1 = \lfloor 8/10 \rfloor + 1 = 1$. Analogously for the y-coordinate, one has

$$y_{\mathrm{id}} = \lfloor (y - y_{\min})/\Delta y \rfloor + 1 \tag{22}$$

which in this case yields $\lfloor (21-5)/10 \rfloor + 1 = 2$, and so on for the velocities. The result is the knowledge that the point of interest falls in the first x-cell, the second y-cell, the second v_x-cell, and the first v_y-cell.

The next step is to map from these marginal indices into the overall grid. Looking at Table I and considering the cell widths, we can determine that the answer should be 7. Let N_x, N_y, N_{vx}, and N_{vy} be the number of cells in each dimension (in this case equal to two for all dimensions). Then one can find the overall grid cell index numerically as

$$\begin{aligned} \mathrm{cell}_{\mathrm{id}} = {} & x_{\mathrm{id}} + (y_{\mathrm{id}} - 1) \times N_x \\ & + (v_{x_{\mathrm{id}}} - 1) \times N_x \times N_y \\ & + (v_{y_{\mathrm{id}}} - 1) \times N_x \times N_y \times N_{vx}. \end{aligned} \tag{23}$$

In this example, this corresponds to $1 + (2-1) \times 2 + (2-1) \times 4 + (1-1) \times 8 = 7$, as expected. This series of calculations exploits the static nature of the grid and is considerably more efficient than brute force searching the grid in order to map every state into a grid cell. Note that additional care must be taken with the boundary cases, i.e. a state that falls exactly on the boundary between grid cells or the state space boundary.

4.3. Method for Doppler, Bearing, and Time-Delay Measurements

The method described in Section 4.2 applies to measurements that contain only bearing, time delay, and SNR information. For CW applications (and potentially for some FM applications), Doppler shift information is also available. The incorporation of this information into the tracker and the likelihood sampling scheme increases computational complexity.

4.3.1. Motivation

Just as with the FM measurements described in Section 4.2, the basic approach is to sample the likelihood function in detail only in areas near actual threshold exceedances. The bearing can be handled as in the FM case, but in general, the ambiguity function for CW involves a coupling of the Doppler and the time delay measurements. However, the response still tends to fall off considerably as one moves several time steps and/or Doppler steps away from the peak.

Considering the functions in detail for a particular application, one can create an offset template that defines the relevant step sizes $\Delta\tau$, $\Delta\phi$, and $\Delta\nu$, as well as the boundaries of the sample region. These can then be applied to each threshold exceedance to create a list of sample points (with duplicates removed). One maintains the option of using more advanced schemes as relevant for the particular application.

4.3.2. Mapping CW Measurements to Grid Cells

The bistatic time, bearing, and Doppler measurements must be converted to Cartesian coordinates in order to evaluate the likelihood for each grid cell. As discussed in Section 4.1.2, a single Doppler shift corresponds to a line of ambiguity in the velocity state space. The likelihood function has a constant value across this line, but that value is mapped into many different grid cells. The approach taken for mapping this line into state space grid cells is to uniformly sample this line. As explained in Section 4.2.2, one must have samples in every grid cell that the line intersects. The heuristic approach for sufficiently sampling this line is as follows.

First, define a sampling increment Δd along the line. Generally, this should be chosen to be smaller than both Δv_x and Δv_y of the state space grid. For example, if $\Delta v_x = \Delta v_y = 1$ m/s then one may choose to sample every 0.5 m/s. This appears sufficient for characterizing the velocity for the types of scenarios discussed in Section 5. For faster moving targets or applications in which the measurements are farther apart in time, the velocity is more important due to the motion update projecting the state forward over a larger effective distance in state space. Such scenarios may require more detailed modeling of the velocity space, either by using a finer grid or by sampling the line of ambiguity more densely.

Using (18) and (19), one can determine where the line of velocity ambiguity intersects the boundaries of the velocity state space. Define one such intersection as $\mathbf{v}_B$ and define the unit vector of the line of ambiguity (found from the slope) as $\hat{\mathbf{v}}_a$. Then the set of sample points $\{\mathbf{v}_j = (v_{x_j}, v_{y_j})\}$ along the line is given by

$$\mathbf{v}_j = \mathbf{v}_B + \hat{\mathbf{v}}_a j \Delta d \tag{24}$$

where j takes on integer values such that the resultant sample point remains inside the modeled velocity state space.

As with the FM case, the time and bearing can be mapped directly to specific points in position space (see (13) and (14)). Combined with the velocity points from (24), a set of points in Cartesian space is defined. Using the method in Section 4.2.4, these points can be related to particular grid cells. Finally, the average likelihood function over the grid cell can be found analogously to the procedure in Section 4.2.3.

4.3.3. Additional Information

In order to facilitate the efficient processing of (20), the number of points on the measurement grid that fall into each Cartesian state space grid cell is precomputed. This operation is more difficult than in the FM case due to the importance of the velocity dimensions and their relationship to the position dimensions. To reduce the computational load, we assume that the Doppler to velocity mapping changes slowly with respect to the x and y coordinates. Effectively, this assumption says that within a particular grid cell, the radial direction is constant. We can then calculate the number of samples in the velocity projection of the state space grid cell using a single point in xy-projection of the cell. The total number of points in each grid cell is this quantity times the number of potential measurement points in the xy-plane, which can be found from the direct relationship between time and bearing and x and y.

This approximation is reasonable far away from the source-receiver pair, but is less valid at close range. At close range, the angle between the target point and the source and receiver can change significantly over a grid cell. For example, consider a source and receiver spaced 5 km apart, and consider a 1 km by 1 km grid cell centered 5 km perpendicular from the midpoint of the source-receiver segment. For this cell, the radial angle can vary by as much as 11.5°, which in extreme cases can lead to an error of 3.5 m/s in the velocity estimate. For the examples of Section 5, this corresponds to one or two grid cells in velocity space. For these examples, this is considered a fair trade-off for the savings in computational cost realized by the approximation, especially as this is only used to the determine the number of potential sample points in each grid cell; for the actual likelihood evaluation at each sample point, the true values are used.

4.4. Summary of Measurement Update Approach

The approach outlined in the preceding sections is to use adaptive sampling of the likelihood function while maintaining a constant grid mesh for representing the PDF. The motivation for this approach is similar to the motivation for adaptive grid methods—to use the most detail where it is most needed. The evaluation of the likelihood function at a small scale is important for identifying the presence of a target and for localizing. This scale is defined by the properties of the likelihood function, and the important regions are identified by the threshold exceedances. In the following section, this approach is tested with an example problem for both the FM and CW cases.

5. EXAMPLE PROBLEMS AND RESULTS

The sampling method described in Section 4 involves some heuristics, tuning parameters, and approximations. To assess its validity, we compare our method to a fixed grid method in which the likelihood function is sampled randomly in each grid cell. We compare the localization, velocity estimation, target detection performance, and computational costs of these approaches to our proposed method for FM and CW waveforms. We

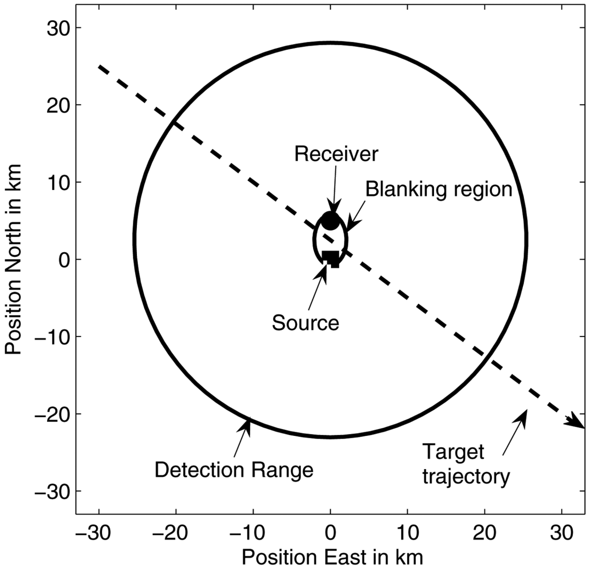

Fig. 3. Example problem geometry.

consider five approaches for each waveform:

1. Intelligently sampling the likelihood
2. Randomly sampling one point in each grid cell
3. Randomly sampling five points in each grid cell
4. Intelligently sampling the likelihood and using a finer state space grid
5. Randomly sampling one point in each grid cell and using a finer state space grid

The details of the examples follow.

5.1. Example Descriptions

The fixed grids used in the problem cover a region from −33 km to 33 km north and −33 km to 33 km east, as well as velocities in the region −12 m/s to 12 m/s north and −12 m/s to 12 m/s east. We consider two different grids. For the "base grid" case, there are 51 grid cells in each of the north and east directions, and 7 grid cells in each velocity dimension. In other words, $N_x = N_y = 51$ and $N_{v_x} = N_{v_y} = 7$, for a total of $N = 127{,}449$ grid cells. This leads to $\Delta x = \Delta y = 1.29$ km and $\Delta v_x = \Delta v_y = 3.43$ m/s.

For the "fine grid" case, $N_{x,\text{fine}} = N_{y,\text{fine}} = 73$ and $N_{v_x,\text{fine}} = N_{v_y,\text{fine}} = 11$, for a total of $N_{\text{fine}} = 644{,}809$ grid cells. This leads to $\Delta x = \Delta y = 0.90$ km and $\Delta v_x = \Delta v_y = 2.18$ m/s. This finer grid is constructed such that the total number of samples in the base grid when sampling five points per cell is approximately equal to the total number of samples using the finer grid when sampling one point per grid cell, that is $N_{\text{fine}} \approx 5N$.

The tracking scenario is summarized in Fig. 3. The source is located at the origin, and the receiver is located at $(0,5)$ km. The receiver has uniform resolution in bearing and can distinguish 72 beams (for a beam width of 5°). For both waveforms, the maximum detection range corresponds to a time delay of arrival of 34.05 s (approximately 25 km) and the minimum (due to blanking region) is 4.333 s. The maximum is a hard cutoff due to the amount of time that is processed. There are no fading effects near the detection limit (i.e. the nominal SNR of the target is independent of range). The nominal mean target SNR is 9.5 dB for FM and CW. Scans are performed every 3 minutes, with either the FM or CW waveform in use, depending on the example problem. No persistent clutter is modeled, but random background clutter with mean SNR of 0 is included.

For FM data, we assume a center frequency of 2350 Hz, a pulse length of 1 s, and a bandwidth of 400 Hz. We use a measurement space with $\Delta\tau = 0.0017$ s and $\Delta\phi = 0.0873$ rad. For the likelihood sampling for FM, we sample a region that includes five time delays and three beams on either side of a measurement y_k. Thus, the sampled swatch is $\pm 5\Delta\tau$ by $\pm 3\Delta\phi$ centered on (τ_k, ϕ_k). We sample three points between each beam (so the spacing is 0.0218 radians) and at each $\Delta\tau$ for the base grid. The fine grid requires more dense sampling in order to make sure each grid cell in a ROI is sampled at least once, so four points between each beam (every 0.0175 radians) are sampled.

For CW, we assume a center frequency of 2575 Hz and the pulse length is 1.5 s. We define a measurement space with $\Delta\tau = 0.4186$ s, $\Delta\phi = 0.0873$ radians, and $\Delta\nu = 1.14$ Hz. The maximum captured Doppler shift (due to assumed windowing in the signal processing) is ±40 Hz. For the likelihood sampling for CW, we sample a region that includes three time delay increments, three beams, and three Doppler increments on either side of a measurement y_k. Thus the sampled swatch is $\pm 3\Delta\tau$ by $\pm 3\Delta\phi$ by $\pm 3\Delta\nu$, centered on (τ_k, ϕ_k, ν_k). We sample three points between each beam (so the spacing is 0.0218 radians) and at each $\Delta\tau$ and $\Delta\nu$ for the base grid. As with FM, we use 0.0175 radians bearing spacing with the fine grid.

We consider a target that originates at $(-30,25)$ km and moves with a constant velocity of $(5.500, -4.125)$ m/s. The target begins outside of the region of detection and enters such that measurement 11 (time 30 min) is the first scan that could potentially contain energy reflected by the target. The target enters the blanking region such that measurement 30 (time 87 min) is the first scan for which the target is blanked. Finally, the target has exited the blanking region at measurement 34 (time 99 min). The last scan in which the target is in the detection region is at measurement 51 (time 150 min).

The data used in the simulation is generated according to the models assumed by the tracker. Specifically, the nominal mean target SNR is input into (8). The nominal mean background clutter SNR is defined to be 0 dB after normalization. The mean at any given point in measurement space is calculated using (12), and then a random variate is generated from the the appropriate exponential distribution. Consequently, the actual data are stochastic.

For localization estimation, a point estimate of the target state is needed. We use the maximum *a posteriori* (MAP) estimate. This yields the grid cell with the highest posterior probability, and the center of this grid cell is used as the point estimate of the state.

The use of the MAP estimate with a grid constrains the ultimate accuracy of the estimate. For example, if the true state is near the corner of a cell, then the best the tracker can do in position is an error of 0.91 km for the base grid and 0.64 km for the fine grid (from geometry, $\sqrt{2}\Delta x/2$). Since the true position will not always be in a corner, actual expected minimum error would be less for a perfect tracker. For these examples, the velocity is constant and therefore falls in the same spot in a grid cell the entire simulation. The minimum attainable error can be found to be 1.5 m/s for the base grid and 1.1 m/s for the fine grid.

We consider 150 different sets of input data for FM and 150 different sets of input data for CW. The target trajectory and source and receiver positions are the same in each run, but the simulated measurements differ, and the random samples (in the random methods) are also different. The localization, detection, and computational performance is averaged across the appropriate 150 FM or CW runs for each approach.

5.2. FM Data Example

For the FM example, the normalized match filter output is thresholded at 10 dB. In this manner, we are trying to track a relatively quiet target (9.5 dB) in a relatively loud environment.

5.2.1. Tracker Performance for FM Data

The root mean squared error (RMSE) in localization for the FM example is given in Fig. 4 for the five sampling methods. The regions in which the target is not detectable are shaded in gray. The “minimum” error value of 0.91 km (as discussed at the end of Section 5.1) for the base grid is also shown. This provides some guidance into the potential accuracy of a particular grid. The value for the fine grid (0.64 km) is not shown to preserve clarity in the figure.

We first consider the base grid examples. The intelligent sampling method clearly outperforms the other two methods. As expected, sampling five points randomly in each grid cell is superior to sampling a single point in each grid cell. The intelligent method yields localization errors on the order of the grid resolution, which suggests that the likelihood sampling is sufficient for realizing something close to a “best possible” performance for the set grid resolution.

The standard deviations on the localization error (shown in Fig. 5) reveal that the superiority of the intelligent sampling method is statistically significant. The error in the position estimates with the random sampling, combined with the lack of any direct measurement of velocity, leads to very poor estimation of the velocity, as shown in Fig. 6. This in turn will feed back

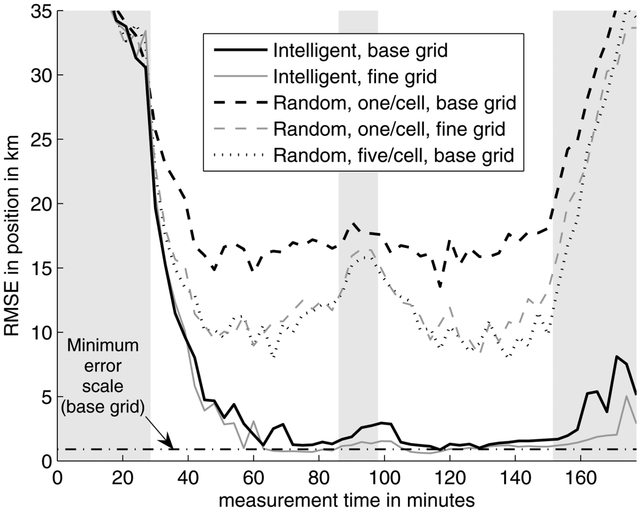

Fig. 4. Localization error for FM data. Gray regions indicate where the target is inherently undetectable.

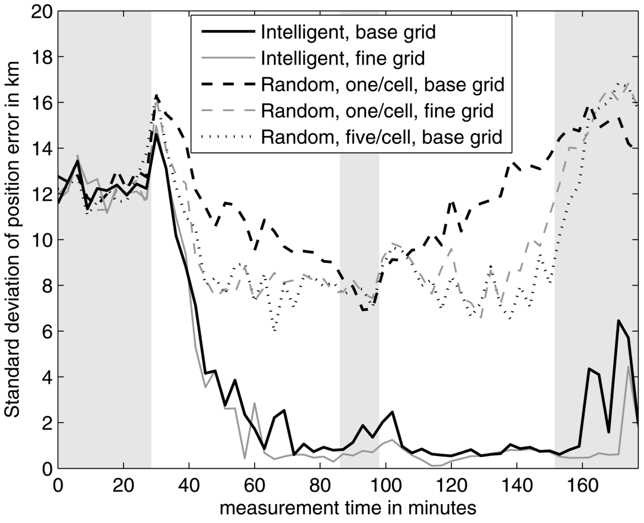

Fig. 5. Standard deviation of position error for FM data.

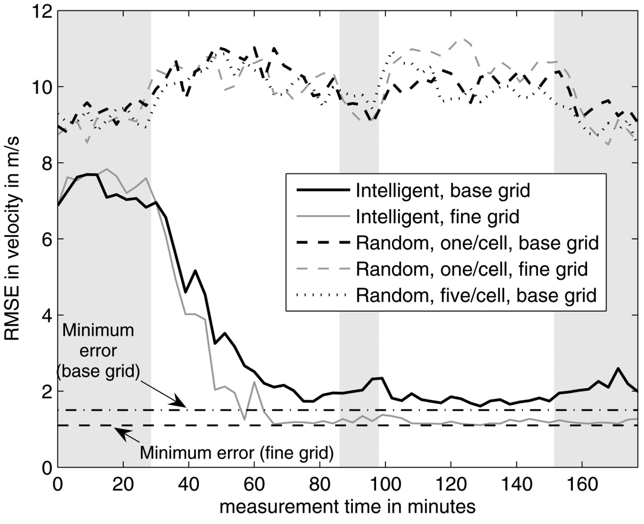

Fig. 6. Velocity error for FM data.

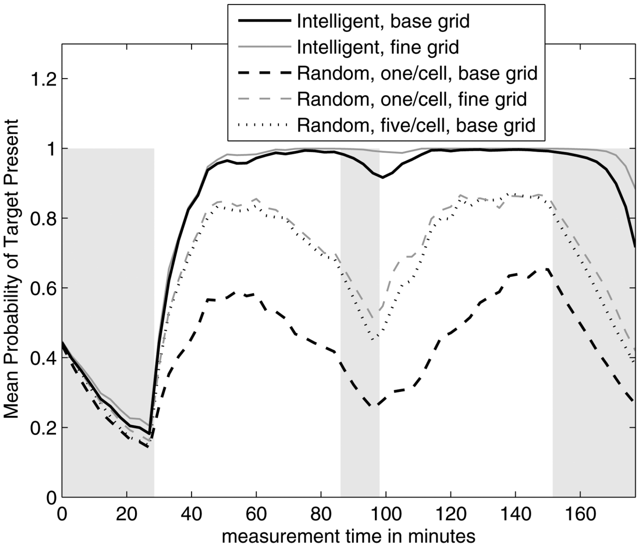

Fig. 7. Target probability for FM data.

into the position estimates via the motion updates. The intelligent sampling characterizes the likelihood much more accurately, and this accuracy helps to reinforce itself over successive updates. The velocity estimation is much better with the intelligent method, and approaches the actual minimum error, as labeled in Fig. 6. The probability of there being a target present in the modeled region is shown in Fig. 7. The prior probability of there being a target is set to 0.5 before the first measurement update. The results for the intelligent sampling method reflect what one would expect. Initially, the target is outside of the detection region, so the target probability falls. Once the target enters the detection region at measurement 11 (time 30 min), the target probability begins to increase. It increases fastest for the intelligent sampling method, whereas the random methods lead to a significantly lower probability of there being a target present, even though the target is present and theoretically detectable.

Even though the nominal mean SNR of 9.5 dB is below the threshold, the target is visible for two reasons. First, the actual SNR is random, so it fluctuates above the mean. Second, energy from the target shows up at multiple points in the state space due to the waveform ambiguity function and the beam response, leading to more opportunities for random threshold exceedances. The tracker expects to find this energy if a target is present. In the random sampling approaches, these regions are not sampled densely enough to capture all of the details of the functions, so some of the energy is missed. This leads to a lower probability of there being a target present and to a poorer estimate of where the target is.

The target is also tracked reasonably well through the blanking region with the intelligent method. The blanking region is incorporated into the predicted SNR model, so essentially the tracker "knows" that a target is undetectable in that region. With the intelligent method, the target is well localized just before it enters the blanking region, and the velocity is well characterized. The motion update projects the estimate into the blanking region. The tracker now estimates the target to be in the blanking region, where it is undetectable. The subsequent lack of received echoes from the target is consistent with the state estimate, so the target probability barely drops. In the random methods, the tracker is less certain where the target is, and therefore it expects to be receiving measurements from the target. Since there are no strong measurements when the target is in the blanking region and the tracker has not localized the target as inside the blanking region, the target probability drops.

There is evidence of another effect in Fig. 4 and especially in Fig. 7. For the localization, the performance of the method of randomly choosing five points per grid cell starts to degrade before the target enters the blanking region. The degradation in target probability is more obvious and applies to both random methods. The cause of this is the mapping between the grid cell (Cartesian) state space and the measurement space (range and bearing).

As the range to the target decreases, the measurement points are more densely packed in Cartesian space (refer back to Fig. 2). Consequently, there is more detail contained in a particular grid cell, and denser sampling is appropriate. Adaptive grid methods are designed to meet this sort of challenge indirectly by increasing the resolution in the state space (see for example [35]). While this may provide benefit in some applications, our approach is to base the sampling of the likelihood directly on the likelihood properties and the received measurements, while maintaining a reasonable grid size in state space. Consequently, the intelligent methods do not show this effect; the performance does not drop until the target actually enters the blanking region, and even there it is reasonably good due to the accurate modeling of the uncertainty and blanking region in the likelihood structure.

The intelligent method continues to track the target relatively well after it leaves the detection region. This is again due to the good localization and velocity estimates before it exits the region. The motion model projects the estimate forward and outside of the detection region, and the lack of measurements on the target is consistent with this estimate. Eventually the increase in uncertainty with successive motion updates (and no relevant measurement data) leads to a drop in target probability, and eventually the mass moves outside the modeled state space (and is appropriately reapportioned across the entire state model, in this case uniformly by the birth/death process).

We now consider the examples with the finer grid resolution. First, we note that the two methods in which the total number of samples is approximately equal—randomly sampling one point per grid cell with the fine grid and randomly sampling five points per grid cell

TABLE II
Comparison of Computational Costs of One Measurement Update for FM

	Total Samples (avg)	Initial Update avg. run time in sec.	Subsequent Update avg. run time in sec.
Intelligent Sampling	16,269	6.8	1.8
Intelligent Sampling, fine grid	20,270	13.3	6.9
Random sampling one/cell	127,449	6.0	5.8
Random sampling one/cell, fine grid	644,809	30.0	29.0
Random sampling five/cell	637,245	31.9	30.8

with the base grid—have very similar RMSE performance across all localization and detection metrics. This suggests that the additional stratification in the sampling in the finer grid and the additional resolution of the grid itself do not improve the overall estimation—the sampling is still just too sparse. These methods both yield results that are inferior to the intelligent sampling method.

For the intelligent sampling method, there is a slight improvement in localization using the finer grid, especially in velocity (Fig. 6), where the estimate follows the absolute minimum error very closely. In the next section, we address the computational costs of these issues and discuss whether this improvement is worth the additional cost of using a finer grid.

5.2.2. Computational Costs for FM Data

The total number of samples taken (on average across all trials, all measurements) by each method for the FM example is shown in Table II, as well as two different run times for the measurement update. The initial update run time is the average for the first measurement update of each run. This includes the overhead involved in setting up some reusable data structures, especially in the intelligent sampling methods. If the sensors move or are in some other way reconfigured between measurements, then this is the average time needed for measurement updates. If the data structures can be reused, there is substantial savings in computational load for subsequent updates. The second column shows the average time required for subsequent updates when the overhead data is reused, which for static sensors is every update except the first.

We first compare the sampling methods for the base grid resolution. For the first update, the intelligent sampling method is slightly slower than the randomly sampling one point per grid cell method, but when the sensors are static it is much faster in the long run. Even considering the initial cost, the large increase in tracking performance strongly favors the intelligent method. The intelligent sampling method is considerably faster than randomly sampling five points per grid cell, and the tracking performance is much better, too. Taken together, these results argue strongly for the superiority of the intelligent sampling method in this example.

Because run times are very implementation-dependent, we also display the actual number of samples taken. This performance criterion suggests the best method is the intelligent sampling, which yields the best performance in terms of localization and target detection with the fewest degrees of freedom in likelihood sampling. The run times do not align perfectly with the number of samples due to the overhead involved in different methods, and particularly the initial overhead in the intelligent sampling method. Nevertheless, the total samples reveal how much more efficient the intelligent method is at sampling the likelihood function—it provides significantly better tracking performance than randomly sampling five points per grid cell with less than three percent of the number of samples.

If the two approaches with the finer grid are considered, the intelligent sampling method is the all-around winner. Not only is the localization and detection performance superior to random sampling with the finer grid, but so is the likelihood update run time and the total number of samples.

An additional factor in the total run time is apparent when the motion update is considered. The average run time for the first motion update is about 6 s for the base grid and 105 s for the finer grid. The timing for the first update is used because subsequent updates depend on the observed measurements due to optimizations in the code that lead to faster run times with better localization. Here, the additional cost of using the finer grid resolution is obvious. Not only does each measurement update take longer, but the motion updates are considerably slower, and with little benefit for localization and detection. While there are scenarios in which this additional resolution may be useful, the trade-off is a large increase in computational costs. The intelligent sampling method appears to allow for significantly improved tracking performance at a much lower cost than refining the grid globally. Adaptive grid methods combined with the intelligent sampling of the likelihood would likely yield even larger gains in performance, but that is left as an item for future research.

5.3. CW Data Example

For the CW example, the normalized matched filter output is thresholded at 9.5 dB. In this manner, we are trying to track a relatively quiet target in a relatively loud environment, although the thresholding is not as extreme as in the FM example. We can use a lower threshold in the CW example than in the FM example because on average there are fewer threshold crossings for CW. Specifically, there are around 368,000 measurement points in the CW example, compared to 1,296,000 in the FM example. Assuming uniform background clutter with a mean SNR of 0 dB, this yields an average of 58 threshold crossings for FM (with a 10 dB threshold) and 50 for CW (with a 9.5 dB threshold). In other examples, there may be motivations for raising or lowering the threshold, such as computational considerations or the need to track low SNR targets.

5.3.1. Tracker Performance for CW Data

The RMSE in localization for the CW example is given in Fig. 8. The shaded gray regions indicate regions in which the target is not detectable by the sonar system. The MAP is again used for the point estimate of the state. We first discuss the cases with the base grid. The intelligent sampling method appears to outperforms the other two methods, but not nearly by as much as in the FM example. As expected, sampling five points randomly in each grid cell is superior to sampling a single point in each grid cell. The standard deviations on the localization error (shown in Fig. 9) suggest that there is so much variation in the random methods that the results are not statistically significant. However, further analysis reveals that the large standard deviations are due to outliers in which the performance is exceptionally poor. The standard deviation of the intelligent method is much smaller, reflecting that it does not perform too badly, even in the extreme cases.

The performance of the three methods in estimating the velocity of the target, shown in Fig. 10, reveals a greater separation between the methods. The intelligent sampling across bearing, range, and Doppler provides more information about the velocity than random sampling. This is in part due to the low resolution in velocity of the grid in state space.

Because not many velocity grid cells are needed to adequately model the target dynamics, not many were defined in the grid. Consequently, they are relatively coarse, and the example results suggest that this resolution is not sufficient for sampling the velocity dependence of the likelihood function.

The error in velocity estimates for the random methods for CW is much smaller than it was for FM. This is partially because CW provides some direct velocity information via the Doppler shift, and partially due to a sampling issue alluded to previously. Specifically, the CW measurements of time delay are spaced more widely than for the FM case, so an equivalent density of sampling in state space is actually a finer sampling in CW measurements than in FM measurements. This sampling effect and the improved velocity estimate also improve the localization in the CW results compared to the FM results.

The target probability is shown in Fig. 11. The intelligent method appears to be the most accurate,

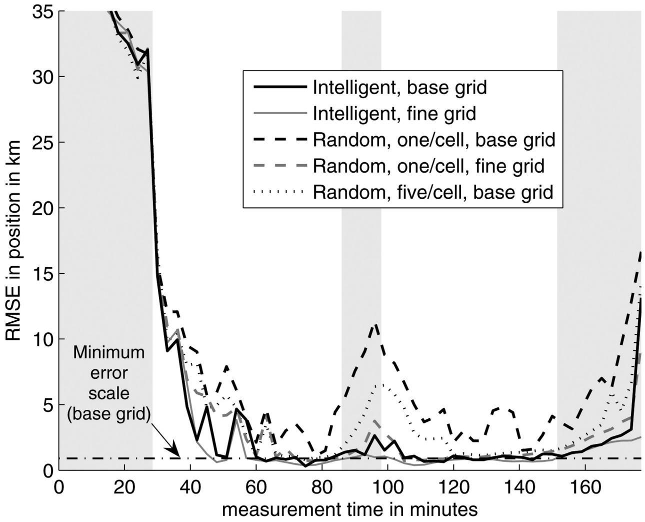

Fig. 8. Localization error for CW data. Gray regions indicate where the target is inherently undetectable.

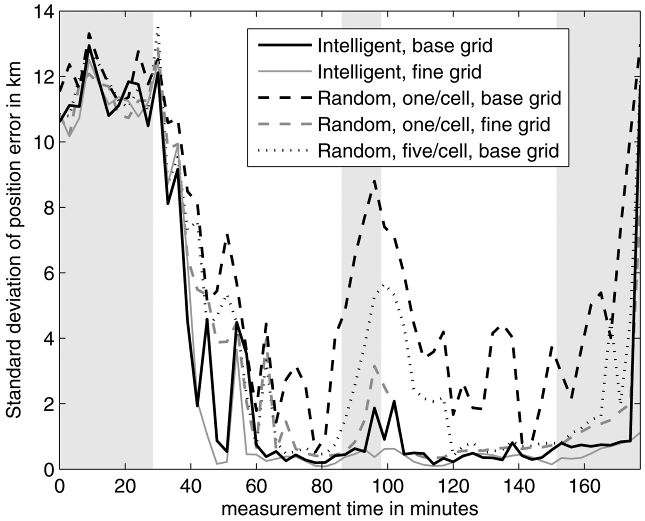

Fig. 9. Standard deviation of position error for CW data.

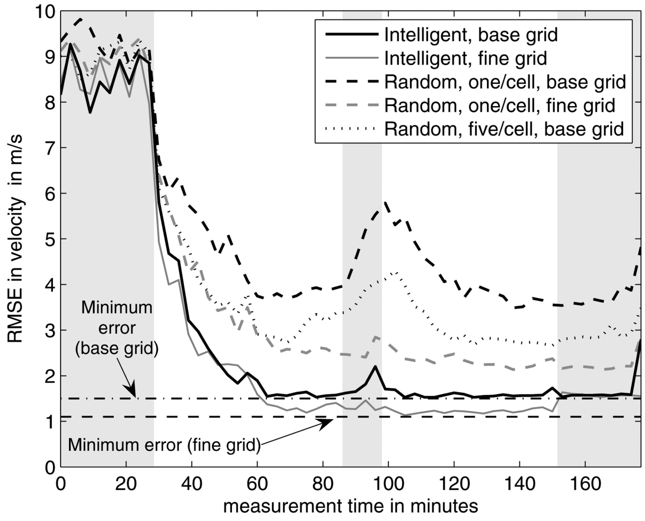

Fig. 10. Velocity error for CW data.

TABLE III
Comparison of Computational Costs of One Measurement Update for CW

	Total Samples (avg)	Initial Update avg. run time in sec.	Subsequent Update avg. run time in sec.
Intelligent Sampling	58,530	22.3	10.2
Intelligent Sampling, fine grid	72,570	79.4	15.4
Random sampling one/cell	127,449	4.3	4.1
Random sampling one/cell, fine grid	644,809	21.5	20.7
Random sampling five/cell	637,245	27.9	27.1

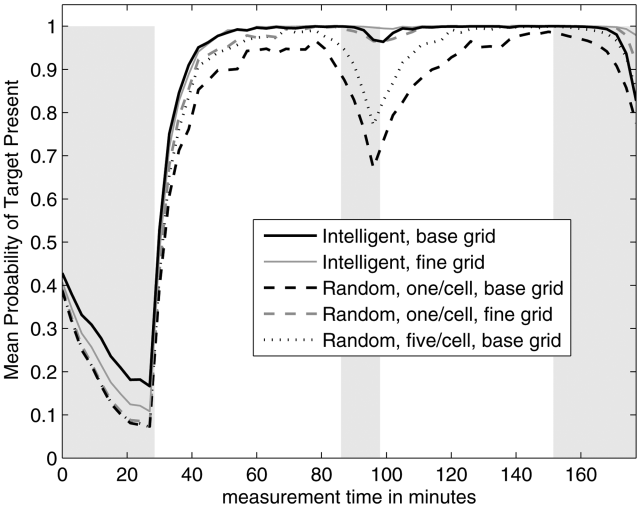

Fig. 11. Target probability for CW data.

followed by the method sampling five points per grid cell. However, the differences are much less than were seen in the FM case shown in Fig. 7. Across all of the figures, there is less evidence in the degradation of performance as the range to the target decreases. Some degradation is visible in the velocity plot in Fig. 10 as the range to the target decreases, but much less in the other plots compared to the FM case. This is related to the relationship between the grid cells size and the likelihood function structure.

For the finer grid with CW, the results show a different trend than in the FM examples. Again, the intelligent sampling with the fine grid yields the best results in RMSE for position and velocity, and the best target detection performance. However, randomly sampling one point per grid cell with the fine grid performs nearly as well as the intelligent sampling methods, and considerably better than randomly sampling five points per grid cell with the base grid, whereas in the FM case, there was little difference. It appears that for the CW case, there is an advantage to sampling the state space more evenly. The reason for this appears to be the relative scales of the likelihoods and the grid.

As described in Section 5.1, the measurement space is finer in range for FM than for CW ($\Delta\tau_{\mathrm{FM}} = 0.0017$ s and $\Delta\tau_{\mathrm{CW}} = 0.4186$ s). Graphically, this would translate to the measurement dots in Fig. 2 being farther apart in position space for CW than for FM. Consequently, fewer samples would be needed per grid cell to adequately sample the position. Therefore, the random sampling methods are sampling the likelihood across time delay and bearing at a finer scale (relative to its features) in the CW case compared to the FM case, and thus, are better capturing the details in the observed acoustics. However, there is also the effect of the necessary Doppler sampling to consider when the trade-offs between performance and computational are examined.

In the FM case, even with one random sample per grid cell and the fine grid, the time delay structure is too fine for the sampling to capture. For example, a 0.0017 s time delay (two-way) corresponds to a range distance (one-way) of just over 1 meter.

In contrast, the fine grid cells have sides of length 900 meters (in position). The two-way time delay of 0.4186 s for CW corresponds to about 300 meters in range (one-way). This is not adequate to fully capture the details of the likelihood in one grid cell with one sample, but given the sampling across the velocity dimensions (of which there are 121 cells for each position grid cell), the marginalized values for position and for velocity lead to a relatively good estimate of the state. The use of five samples per grid cell improves the average sampling density, but not in a stratified manner. While on average the density increases, the sampling can still often be quite bad due to its randomness. The fine grid with one sample per grid achieves a similar average density, but in a much more systematic manner.

By some standards, the localization and detection performance of the intelligent method (with either grid) and the randomly sampling one per grid cell with the fine grid are comparable. However, the computational costs are quite different, as discussed in the following.

5.3.2. Computational Costs for CW Data

The total number of samples taken (on average across all trials, all measurements) by each method for CW is shown in Table III, as well as the initial and subsequent run times of the measurement update. Several additional optimizations and approximations could reduce the overhead involved in the intelligent method, but the current implementation does not contain these. As with the FM example, the first motion update takes around 6 s for the base grid and 105 s for the fine grid when the probability is not well localized.

In a scenario in which the sensors are moving, the run time for the intelligent method is five times that of randomly sampling one point per grid cell, and slightly less than sampling five per grid cell. The run time for the measurement update for the intelligent method on the base grid is about equal to the run time of randomly sampling one point per grid cell on the fine grid. The intelligent sampling on the fine grid is much slower (for the initial update) than any other example, but this run time drops significantly in subsequent updates.

Given the performance results in Section 5.3.1 and these run times, the intelligent sampling method with the base grid is superior to the random methods with either grid size. For example, in the blanking region the error for the intelligent method is on the order of the width of one grid cell (about 1 km, which is nearing the limit of the possible resolution), whereas the error for randomly sampling one point per grid cell is on the order of eight grid cells (or 10 km). In many applications, this difference will be significant and the extra run time of the intelligent sampling is likely justified, as the method still easily runs in real time (since the pings are at least 30 s apart).

If the sensors are approximately stationary, then the intelligent method with the fine grid appears to be the best option, as the tracking performance improves with only a 5 s increase in run time. However, if the sensors are moving, the increase in run time is probably prohibitive, and the intelligent sampling with the base grid is the best approach.

All of this neglects the costs of the motion update, which is very high for the fine grid (105 s) compared to the base grid (6 s). Once this is considered, the superiority of the intelligent method with the base grid compared to any method with the fine grid is obvious.

The actual number of samples used suggests that the best method is intelligent sampling, which yields the best performance in terms of localization and target detection with the fewest degrees of freedom in likelihood sampling. This suggests, but does not guarantee, that improved methods for implementing the intelligent sampling scheme may lead to shorter run times than the random methods. Areas for such improvement include the mapping of Doppler shifts into lines in velocity, and computing the number of measurements in a given grid cell (used to facilitate the evaluation of (20)), which should be approximated reasonably well by geometric arguments rather than requiring explicit, brute force calculation.

5.4. Robustness of Intelligent Sampling

As a test of the method's sensitivity to the tuning parameters, an additional set of runs was performed in which the intelligent sampling method sampled a smaller region around each threshold exceedance for the base grid.

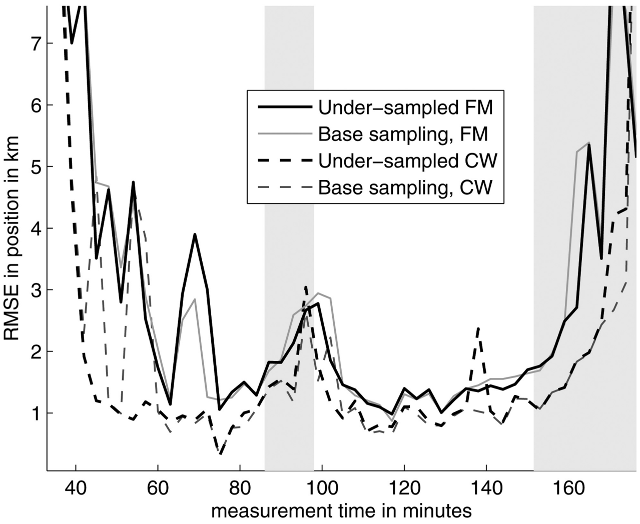

Fig. 12. Sensitivity analysis of RMSE for FM and CW.

For the FM example, we reduce the sampling swatch from $\pm 5\Delta\tau$ by $\pm 3\Delta\phi$ centered on (τ_k, ϕ_k) to $\pm 3\Delta\tau$ by $\pm 2\Delta\phi$ centered on (τ_k, ϕ_k). This leads to roughly 7,000 total samples per measurement update (compared to 16,000 in the original case). The comparison of localization performance is shown in Fig. 12. There are some regions in which the base sampling appears to work better, and there are some regions in which the under-sampled example works better, but in either case the differences are relatively small, on the order of the grid cell dimensions. Such small differences are considered minor by the assumption that the grid cell size represents acceptable error. Consequently, it appears that the results shown in Section 5.2 are not very sensitive to under-sampling, and the gains in computational performance may even be larger than shown in Table II.

For the CW case, an example was considered in which the number of beams, time delay steps, and Doppler steps sampled on either side of each measurement is reduced, leading to a sample swatch that is $\pm 2\Delta\tau$ by $\pm 2\Delta\phi$ by $\pm 2\Delta\nu$, centered on (τ_k, ϕ_k, ν_k) (instead of the original $\pm 3\Delta\tau$ by $\pm 3\Delta\phi$ by $\pm 3\Delta\nu$). The total number of samples decreases to one-third of its value using the original sampling swatch (from 58,530 samples down to 20,000 samples). The resulting change in error was not significant, as shown in Fig. 12. For the most part, the results are indistinguishable, with the reduced sampling outperforming the original sampling at two updates early in the run, and the original sampling outperforming the reduced at one later update.

These results reveal the robustness of the method to the specifics of the sampling scheme. They reemphasize that the intelligent sampling method yields better localization than the random sampling methods even if the likelihoods are slightly under-sampled. This supports the general result that intelligent sampling based on the observed measurements and the scale of the likelihood functions provides improved tracking performance over methods that do not explicitly consider these characteristics.

6. DISCUSSION

In Bayesian tracking, the measurement information is incorporated into the probability density via the likelihood function. In a grid-based method, the proper value to use when updating the probability mass in a grid cell is the integral of the likelihood function across all states in that grid cell. In this paper, a method for adaptively sampling the likelihood function for this integration has been introduced. This method scales to the actual measurements received and can be customized for the likelihood model that is relevant for a particular application.

In certain applications, such as finding a single quiet target in a very loud and cluttered environment, it may be necessary to base tracking on a level of detail that preserves information from sensor outputs. When the sensors provide very detailed information, then accurate likelihood functions will have a very detailed structure, too, in order to reflect the associated uncertainties correctly. Consequently, the implementation of the tracker should evaluate these likelihoods accurately. In other applications, smooth and broad likelihood models may be appropriate, in which case existing methods such as standard particle filters or grid-based methods with random likelihood sampling are likely adequate.

The intelligent sampling approach presented in this article allows computational resources to be invested in sampling the areas where there are the most relevant details. At the same time, a relatively coarse grid can be used to represent the probabilities over the state space, which further reduces the computational effort compared to methods that refine the state space to indirectly improve likelihood sampling.

The example problems reflect these qualities, showing that the intelligent method can lead to significant performance gains at reasonable computational costs. Specifically, the trade-offs between improving the likelihood sampling and refining the grid cell size reveal that large gains can be made by carefully sampling the likelihood sampling.

The value of this method extends beyond the specific case of a fixed grid-based Bayesian tracker because all Bayesian methods require accurate modeling and evaluation of the likelihood function. In an adaptive grid, one must still adequately integrate the likelihood function over each grid cell, so the method is directly applicable. By extension, the likelihood sampling may also indicate the regions in which a refined grid is useful. The combination of a locally adaptive grid method with the intelligent sampling presented in this article should further improve the overall efficiency of the modeling, although this is not examined in this article.

The lessons regarding sampling presented in this article can also be applied to particle filtering, because those methods still require adequate sampling of the likelihood to capture the subtleties of a likelihood function such as described in Section 3. This fine sampling of the likelihood model requires the development of new, advanced importance sampling methods for particle filters. Such adaptations are left for future work.

7. SUMMARY

In certain applications, small details in the received signals are important for distinguishing a target from clutter and effectively tracking it. By adopting a track-before-detect paradigm and using unclustered normalizer output data, these details can be extracted. However, the likelihood function in a Bayesian tracker must then reflect these details, leading to a function that varies at a relatively small scale compared to the state space. In this paper, an adaptive likelihood sampling scheme is presented that appropriately samples the likelihood according its structure and the measurements actually received. Two example problems reveal that this method can lead to significantly improved detection and localization performance while realizing a computational savings over more traditional grid-based methods. Although it was applied to a fixed grid for the sake of comparison, the basic sampling concepts can be extended to adaptive grid and particle filter methods.

APPENDIX

Consider a sequence of independent, univariate random variables $Y_1,\ldots,Y_K$ with corresponding PDFs $f_1,\ldots,f_K$ and CDFs $F_1,\ldots,F_K$. In the context of this paper, each Y_k corresponds to, say, the measured SNR at a given point (i.e., time delay, bearing, and Doppler shift) in measurement space. The distribution of each Y_k will also depend upon whether we assume a target is present or not.

Now suppose we are interested in the subset of values which exceed a given threshold $\eta > 0$. Let $K_1 < \cdots < K_I \in \{1,\ldots,K\}$ denote the (random) indices of those exceedances. If these threshold exceedances constitute our definition of a measurement, then the corresponding likelihood function is the probability density of obtaining the particular set of indices (i.e., points in measurement space) $\mathbf{k} = (k_1,\ldots,k_I)$ with a corresponding set of values $\mathbf{y} = (y_1,\ldots,y_K)$. We shall denote this likelihood function $L(\mathbf{k},\mathbf{y})$.

For $I = 0$, this takes the simple form

$$L(\emptyset,\mathbf{y}) = \prod_{k=1}^{K} F_k(\eta) \qquad (25)$$

while, for $I = K$, we have

$$L(1,\ldots,K,\mathbf{y}) = \prod_{k=1}^{K} [1 - F_k(\eta)] f_k(y_k \mid \eta) \qquad (26)$$

where the conditional PDF is

$$f_k(y_k \mid \eta) := \frac{f_k(y_k)}{1 - F_k(\eta)} 1_{[\eta,\infty)}(y_k). \tag{27}$$

If, however, $I = 1$, then

$$\begin{aligned} L(k_1, \mathbf{y}) &= P[Y_{k_1} = y_{k_1} \mid I = 1, K_1 = k_1]P[I = 1, K_1 = k_1]] \\ &= f_{k_1}(y_{k_1} \mid \eta) \prod_{k=1}^{k_1 - 1} F_k(\eta)[1 - F_{k_1}(\eta)] \prod_{k'=k_1+1}^{K} F_{k'}(\eta) \\ &= L(\emptyset, \mathbf{y}) \frac{1 - F_{k_1}(\eta)}{F_{k_1}(\eta)} f_{k_1}(y_{k_1} \mid \eta). \end{aligned} \tag{28}$$

More generally, for $I \in \{1, \ldots, K\}$ and $1 \leq k_1 < \cdots < k_I \leq K$, we have

$$L(\mathbf{k}, \mathbf{y}) = L(\emptyset, \mathbf{y}) \prod_{i=1}^{I} \frac{1 - F_{k_i}(\eta)}{F_{k_i}(\eta)} f_{k_i}(y_{k_i} \mid \eta). \tag{29}$$

REFERENCES

[1] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp
A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking.
IEEE Transactions on Signal Processing, **50**, 2 (Feb. 2002), 174–188.

[2] J. M. Aughenbaugh, B. A. Yocom, and B. R. La Cour
Active clutter reduction through fusion with passive data.
In *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, Scotland, July 26–29, 2010.

[3] J. M. Aughenbaugh and B. R. La Cour
Use of prior information in active sonar tracking.
In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, July 2009.

[4] Y. Barniv
Dynamic programming solution for detecting dim moving targets.
IEEE Transactions on Aerospace and Electronic Systems, **21** (1985), 144–156.

[5] Y. Barniv and O. Kella
Dynamic programming solution for detecting dim moving targets, Part II: Analysis.
IEEE Transactions on Aerospace and Electronic Systems, **23** (1987), 776–788.

[6] Y. Boers, F. Ehlers, W. Koch, T. Luginbuhl, L. D. Stone, and R. L. Streit
Track before detect algorithms.
EURASIP Journal on Advances in Signal Processing, vol. 2008, 2008, 1–2.

[7] R. S. Bucy and K. D. Senne
Digital synthesis of nonlinear filters.
Automatica, **7** (1971), 287–289.

[8] K-C. Chang, S. Mori, and C-Y. Chong
Evaluating a multiple-hypothesis multitarget tracking algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **30**, 2 (Apr. 1994), 578–590.

[9] H. Cox
Fundamentals of bistatic active sonar.
In *Underwater Acoustic Data Processing*, Y. T. Chan, Ed. Kluwer Academic Publishers, 1989, 3–24.

[10] S. J. Davey, M. G. Rutten, and B. Cheung
A comparison of detection performance for several track-before-detect algorithms.
EURASIP Journal on Advances in Signal Processing, vol. 2008, Article ID 428036, 10 pages, 2008.

[11] R. J. P. De Figueiredo and Y. G. Jan
Spline filters.
In *Proceedings of the 2nd Symposium on Nonlinear Estimation Theory and Its Applications*, San Diego, 1971, 127–141.

[12] A. Doucet
On sequential Monte Carlo sampling methods for Bayesian filtering.
Statistical Computation, **10**, 3 (2000), 197–208.

[13] N. J. Gordon, D. J. Salmond, and A. F. M. Smith
Novel approach to nonlinear/non-Gaussian Bayesian state estimation.
IEE Proceedings of Radar and Signal Processing, **140**, 2 (Apr. 1993), 107–113.

[14] Y. C. Ho and R. C. K. Lee
A Bayesian approach to problems in stochastic estimation and control.
IEEE Transactions on Automatic Control, **9** (1964), 333–339.

[15] C. Jemmott, R. L. Culver, and N. Bose
Passive sonar target localization using a histogram filter with model derived priors.
In *IEEE Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, 2008.

[16] L. Johnston and V. Krishnamurthy
Performance analysis of a dynamic programming track before detect algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 1 (Jan. 2002), 228–242.

[17] S. J. Julier and J. K. Uhlmann
A new extension of the Kalman filter to nonlinear systems.
In *SPIE AeroSense Symposium*, Orlando, FL, Apr. 21–24 1997, 182–193.

[18] K. Kastella and C. Kreucher
Multiple model nonlinear filtering for low signal ground target applications.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 2 (Apr. 2005), 549–564.

[19] G. Kitagawa
Non-Gaussian state-space modeling of nonstationary time series.
Journal of the American Statistical Association, **82**, 400 (1987), 1032–1041.

[20] S. C. Kramer and H. W. Sorensen
Bayesian parameter estimation.
IEEE Transactions on Automatic Control, **33**, 2 (Feb. 1988), 217–222.

[21] S. C. Kramer and H. W. Sorensen
Recursive Bayesian estimation using piece-wise constant approximations.
Automatica, **24**, 6 (1988), 789–801.

[22] C. M. Kreucher, K. Kastella, and A. O. Hero, III
Multitarget tracking using a particle filter representation of the joint multitarget density.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 4 (2003), 1396–1414.

[23] C. M. Kreucher, B. Shapo, and R. Bethel
Multitarget detection and tracking using multi-sensor passive acoustic data.
Proceedings of the 2009 IEEE Aerospace Conference, Mar. 2009, 1–16.

[24] B. R. La Cour
Ensemble-based Bayesian detection and tracking.
In *Proceedings of Meetings on Acoustics*, vol. 1, 2008.

[25] B. R. La Cour
Stationary priors for Bayesian target tracking.
In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, June 30–July 3, 2008.

[26] E. A. Lehmann and R. C. Williamson
Particle filter design using importance sampling for acoustic source localization and tracking in reverberant environments.
EURASIP Journal on Applied Signal Processing, 2006, 1–9.

[27] M. Morelande and S. Challa
Manoeuvring target tracking in clutter using particle filters.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 1 (Jan. 2005), 252–270.

[28] L. Rabiner and B. Juang
An introduction to hidden Markov models.
IEEE Acoustics, Speech and Signal Processing Magazine, **3**, 1 (Jan. 1986), 4–16.

[29] D. B. Reid
An algorithm for tracking multiple targets.
IEEE Transactions on Automatic Control, **24**, 6 (1979), 843–854.

[30] M. G. Rutten, B. Ristic, and N. J. Gordon
A comparison of particle filters for recursive track-before-detect.
In *Proceedings of the 8th International Conference on Information Fusion*, Philadelphia, PA, July 2005, 169–175.

[31] D. J. Salmond and H. Birch
A particle filter for track-before-detect.
In *Proceedings of the American Control Conference*, Arlington, VA, June 25–27, 2001, 3755–3760.

[32] U. Scheunert, N. Mattern, P. Lindner, and G. Wanielik
Generalized grid framework for multi sensor data fusion.
In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, June 2008, 814–820.

[33] H. W. Sorenson, Ed.
Kalman Filtering: Theory and Application.
IEEE Press, 1985.

[34] L. D. Stone, T. L. Corwin, and J. B. Hofmann
Technical documentation of Nodestar.
Naval Research Laboratory Technical Report, Tech. Rep. NRL/FR/5580-95-9788, 1995.

[35] L. D. Stone, C. A. Barlow, and T. L. Corwin
Bayesian Multiple Target Tracking.
Boston: Artech House, 1999.

[36] R. L. Streit, M. L. Graham, and M. J. Walsh
Multitarget tracking of distributed targets using Histogram-PMHT.
Digital Signal Processing, **12**, 2–3 (2002), 394–404.

[37] R. L. Streit and T. E. Luginbuhl
A probabilistic multi-hypothesis tracking algorithm without enumeration and pruning.
In *Proceedings of the Sixth Joint Service Data Fusion Symposium*, Laurel, MD, June 1993, 1015–1024.

[38] R. Streit and R. Barrett
Frequency line tracking using hidden Markov models.
IEEE Transactions on Acoustics, Speech and Signal Processing, **38**, 4 (Apr. 1990), 586–598.

[39] S. Thrun, W. Burgard, and D. Fox
Probabilistic Robotics.
Cambridge, MA: The MIT Press, 2005.

[40] A. J. Viterbi
Error bounds for convolutional codes and an asymptotically optimum decoding algorithm.
IEEE Transactions on Information Theory, **13**, 2 (1967), 260–269.

[41] L. J. Ziomek
Fundamentals of Acoustic Field Theory and Space-Time Signal Processing.
Boca Raton: CRC Press, 1995.

Jason M. Aughenbaugh received the B.S.E. degree in operations research and financial engineering from Princeton University with a certificate in engineering management systems. He received the M.S.M.E. degree in the area of controls and dynamic systems from the George W. Woodruff School of Mechanical Engineering at the Georgia Institute of Technology, where he then earned the Ph.D. degree while focusing on uncertainty modeling and the efficient management of information in the engineering design process.

Dr. Aughenbaugh has been with Applied Research Laboratories, The University of Texas at Austin, since 2006. His research interests include uncertainty modeling, decision making, and multi-sensor tracking and data fusion.

Brian R. La Cour received the B.S. degree in physics and the M.S. degrees in physics and mathematics from the University of New Orleans, New Orleans, LA, in 1991 and 1995, respectively. He received the Ph.D. degree in physics from The University of Texas at Austin in 2000, while specializing in statistical mechanics.

He has been with Applied Research Laboratories, The University of Texas at Austin, since 2001 and has been the Head of the Signal Physics Division's Research and Development group since 2004. His research interests include active sonar signal processing and multisensor tracking and data fusion.