December 2010 Volume 5 Number 2 ISSN 1557-6418

JAIF

Journal of Advances in Information Fusion

A semi-annual archival publication of the International Society of Information Fusion

Regular Papers Page

From the Administrative Editor

Final Stages of the Production Process for JAIF Manuscripts

ISIF

INTERNATIONAL SOCIETY OF INFORMATION FUSION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied INFORMATION FUSION technologies. Technical areas of interest include target tracking, detection theory, applications for information fusion methods, image fusion, fusion systems architectures and management issues, classification, learning, data mining, Bayesian and reasoning methods.

Journal of Advances in Information Fusion (ISSN 1557-6418) is published semi-annually by the International Society of Information Fusion. The responsibility for the contents rests upon the authors and not upon ISIF, the Society, or its members. ISIF is a California Nonprofit Public Benefit Corporation at P.O. Box 4631, Mountain View, California 94040.

From the Administrative Editor:

December 2010

Final Stages of the Production Process for JAIF Manuscripts

When a JAIF manuscript has been accepted for publication, the next stage in the editorial procedure is for the authors to provide a folder containing a complete set of manuscript production files. The correct format for the production files is specified at http://www.isif.org/prepforpublication. Upon receipt by the Administrative Editor, the production files go through a complete audit. For more detail on the auditing process, see the editorial in the December 2007, Volume 2, Number 2, issue of JAIF.

After the production files are determined to be acceptable, the next step in the editorial process is to assign a copy editor to proofread the final manuscript. JAIF currently utilizes two copy editors, Associate Editor-in-Chief Uwe Hanebeck of the University of Karlsruhe and Associate Administrative Editor Ruixin Niu of Virginia Commonwealth University, to handle the current production load of accepted manuscripts.

The role of a JAIF copy editor is to thoroughly proofread each manuscript making sure that it meets the standards established by the editorial board of the journal. For example, typical standards used by JAIF for publishing academic papers can be found at http://en.wikipedia.org/wiki/Academic_publishing. In proofreading, the most common errors that a JAIF copy editor looks for are of the typographical, punctuation, and grammatical type. Additionally, other common errors to look for involve mistakes in referencing sections, equations, and figures or tables. With that, it is necessary to check that these referenced items are not missing labels, erroneously labeled, or formatted incorrectly. Finally, an important function of the JAIF copy edit process is to look for issues with incorrect equations, and improper usage of scientific terms.

When the copy editor has completed the proofing of a manuscript, all recommended modifications are scanned and forwarded to the corresponding author as a marked up PDF manuscript file. The author then reviews the changes, and makes any requested changes of his/her own by marking up the same PDF file. These

changes are forwarded to the copy editor for review and approval. When the copy editor and author are satisfied that no further modifications are required, the author then updates the manuscript production files and sends these to the copy editor. At that point, the copy editor uploads the files to the JAIF system and releases the paper for production, which notifies the typesetter to begin preparing a first set of proofs.

The entire copy edit process typically takes four weeks, and the typesetter takes another two weeks beyond that to produce first proofs (see the editorial in the June 2010, Volume 5, Number 1, issue of JAIF). When the proofs are ready, the authors are notified to give the manuscript one last check for modifications. If no modifications are required, the proofs are posted on JAIF's web site as a first version of the paper.

Robert Lynch
Administrative Editor

A Decision-Centric Framework for Density Forecasting

GABRIEL TEREJANU
PUNEET SINGLA
TARUNRAJ SINGH
PETER D. SCOTT

In general, the uncertainty propagation problem, in which the uncertain initial condition evolves through a dynamic system driven by noise, is seen strictly from the producer's perspective. This means that uncertainty propagation algorithms are derived and evaluated based on statistical measures independent of the user's decision needs. However accurate the uncertainty evolution given by a particular method, it may be less than optimal to the user or the decision maker, who takes decisions based on an implicit or explicit utility function. While in a static environment, one may be able to select an appropriate method for uncertainty propagation, in a dynamic environment with an ever-changing utility function this becomes a challenging task.

The goal of the present work is to reconcile the two views into a decision-centric framework which provides both a more accurate approximation to the relevant probability density function and a more precise expected utility value for the decision maker. A numerical example using a puff-based dispersion model, for forecasting downwind concentrations of toxic materials, demonstrates the capacity of this approach to focus computational resources on regions of particular interest such as high population density. A second example shows improvement over alternative methods as measured by a variety of utility-weighted metrics.

Manuscript received October 6, 2009; revised February 3, 2010, May 7, 2010; released for publication May 11, 2010.

Refereeing of this contribution was handled by Peter Willett.

Authors' address: G. Terejanu and P. D. Scott, Department of Computer Science & Engineering, University at Buffalo, Buffalo, NY 14260, E-mail: (terejanu@buffalo.edu, peter@buffalo.edu); P. Singla and T. Singh, Department of Mechanical & Aerospace Engineering, University at Buffalo, Buffalo, NY 14260, E-mail: (psingla@buffalo.edu, tsingh@buffalo.edu).

1. INTRODUCTION

Decision makers increasingly rely on mathematical models in choosing the right set of actions in critical situations. The accuracy of mathematical models in predicting the physical state of the system directly affects the accuracy of the decision making process. Such situations are often encountered in deployment of emergency responders in response to extreme events such as covert release of hazardous material, storm surge due to a hurricane, wild fire, etc. Disaster response managers routinely use numerical modeling to assist in hazard response and mitigation. However, any numerical model used to forecast physical state variables and assist in decision making is a reflection of numerous assumptions and simplifications to permit the determination of a tractable model. The error inherent in any model is a result of model truncation, errors in model parameters, and errors in initial and boundary conditions. Together these factors cause overall prediction model accuracy to degrade as the simulation evolves. Hence, it is important to forecast the evolution of a physical state variable with its attendant uncertainty given the uncertainties in the inputs to the numerical model. Based on the forecast of physical state and associated uncertainty, decisions can be made on deploying emergency responders, evacuating cities, sheltering or medical gear caching.

The optimal decision under uncertainty corresponds to maximizing the expected value of a utility function or minimize the expected value of a loss function [40]. The utility or its complement, the loss function, are defined to measure the consequences of the decision making process. The accurate computation of the expected loss requires the knowledge of the probability distribution of the physical state variable due to model and input uncertainties. The exact time evolution of state probability density function (pdf) is given by the Fokker-Planck-Kolmogorov Equation (FPKE) [27].

If FPKE could be solved for the state pdf, it would be possible to calculate statistical moments like the mean state and the error covariance at different times as well as different expectations such as the expected loss. Analytical solutions for the FPKE exist only for a stationary pdf and are restricted to a limited class of dynamical systems [9, 27]. Thus researchers are actively looking at numerical approximations to solve the FPKE [15–17, 20, 23], generally using the variational formulation of the problem. However, these methods are severely handicapped for even low dimensions because the discretization of the space over which the pdf lives is, computationally impractical.

To emulate the exact methods, many approximate techniques exist in the literature to approximate the uncertainty evolution problem, the most popular being Monte Carlo (MC) methods [8], Gaussian closure [12], Equivalent Linearization [28], and Stochastic Averaging [18, 19]. All of these algorithms except Monte Carlo methods are similar in several respects, and are suitable

only for linear or moderately nonlinear systems, because the effect of higher order terms can lead to significant errors. The Markov Chain Monte Carlo (MCMC) or sequential Monte Carlo methods [31] are other attractive alternatives in the case of low-order nonlinear systems to solve the FPKE. Their applicability to higher-order systems, particularly in "plain-vanilla" forms, is limited by their high computational complexity and sensitivity to properties such as the rate of decay of the conditional pdf. As noted by Daum [7], sequential Monte Carlo methods are not immune to the "curse of dimensionality," and their effective use should take into account the smoothness constraint implied by the FPKE.

Recently Terejanu et al. [36] have proposed the Gaussian mixture model for accurately solving the FPKE in a computationally effective manner. The key idea is to approximate the state pdf by a finite sum of Gaussian density functions whose mean and covariance are propagated using linear theory. The weights corresponding to different Gaussian kernels are updated by requiring the mixture to satisfy the FPKE [36]. With this formulation, the mixture problem can be solved efficiently and accurately using convex optimization solvers, even if the mixture model includes many terms. Another advantage of the proposed method is that it decouples a large uncertainty characterization problem into many small scale problems. As a consequence, the algorithm can be parallelized on today's high performance computing systems. Although Gaussian mixture idea has been successfully applied to low and moderate dimension systems ($n = O(10)$), including the uncertainty propagation through two-body system and toxic cloud transported by wind [10, 36, 37], like any other method to solve the FPKE it only provides an approximate description of the uncertainty propagation problem by restricting the FPKE solution space to a small number of parameters.

In general, the uncertainty evolution process does not take into account the knowledge about the decision making process. Evaluation of the approximate state pdf provided by different methods is based on statistical measures, such as minimization of FPKE error, integral square error between true pdf and its approximation [36], mean square error [14] or expected exponential of estimation error [4, 29]. This process is independent of the user's decision needs and is referred here as the producer's perspective where the accuracy of the forecast is the main driver in the algorithm evaluation [14]. These assumptions make the problem tractable and computationally efficient, which satisfies the requirement of minimizing decision latency, but the approximations may be of little use when computing the expected loss, since they are not sensitive to the decision maker's loss function [37]. For example, an approximation which underestimates a tail of the forecast pdf where the main support of the loss function resides.

Ideally the uncertainty evolution should be performed from the user's perspective [25], i.e., it should take into account the structure of the utility or loss function. While in a static environment, one may be able to select an appropriate method for uncertainty propagation, in a dynamic environment with an ever-changing utility function this becomes a challenging task. The main objective of this work is to reconcile the two views into a decision-centric framework which provides both a more accurate approximation to the relevant state probability density function and a more precise expected utility value for the decision maker. This is achieved by incorporating contextual loss information held by the decision maker into the density forecasting process.

We use a Gaussian mixture approximation to the state pdf and propose a "non-intrusive" way of computing an approximate pdf that addresses the region of interest and is closer to the true pdf in the sense of minimizing FPKE error. Non-intrusive refers here to the fact the we do not require a new uncertainty propagation method when incorporating the loss function into the derivation. The interaction level between the Decision Maker (DM) and Density Forecasting (DF) is acting at the process refinement level which manages the resources of the density forecasting method, in this case the location of the Gaussian components.

A progressive selection method is designed to add new Gaussian components to the initial Gaussian mixture, such that probabilistic support is reaching the region of interest at the decision time. The initial weights of the added Gaussian components are set to zero and they are modified when propagated throughout the nonlinear dynamic system to minimize the error in the FPKE [36]. Therefore, if there is any probability density mass in the region of interest it will be represented by the non-zero weight of the new Gaussian components at the decision time.

We mention that the similar ideas have been explored in risk sensitive particle filters [39], which are not to be confused with risk sensitive filters [4, 29]. The risk sensitive particle filter modifies the sampling density of the standard particle filter so that more samples are generated in high risk regions of the state space. This is achieved with a risk function obtained using a Markov decision process to approximate the future risk of decisions from a particular state.

The structure of the paper is as follows: first the decision making problem is stated in Section 2 and the Gaussian Sum approximation to the forecast pdf is presented in Section 3. The progressive selection of Gaussian components is derived in Section 4 followed by two numerical examples in Section 5 to motivate and to illustrate the performance of the method. The conclusions and future work are discussed in Section 6.

2. PROBLEM STATEMENT

Consider a general n-dimensional continuous-time noise driven nonlinear dynamic system with uncertain

initial conditions given by the following equations:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(t,\mathbf{x}(t)) + \mathbf{\Gamma}(t)\\ \mathbf{x}(t_0) &\sim p(t_0,\mathbf{x}_0)\end{aligned} \tag{1}$$

where $\mathbf{\Gamma}(t)$ represents a Gaussian white noise process with the correlation function $\mathbf{Q}\delta(t-\tau)$, and uncorrelated with the initial condition.

We are interested in finding the forecast probability density function $p(t,\mathbf{x}(t))$ whose time-evolution is given by the following partial differential equation known as the Fokker-Planck-Kolmogorov Equation (FPKE) [27]:

$$\begin{aligned}\frac{\partial}{\partial t}p(t,\mathbf{x}) = &-\frac{\partial p(t,\mathbf{x})^T}{\partial \mathbf{x}}\mathbf{f}(t,\mathbf{x}) - p(t,\mathbf{x})\mathrm{Tr}\left[\frac{\partial \mathbf{f}(t,\mathbf{x})}{\partial \mathbf{x}}\right]\\ &+\frac{1}{2}\mathrm{Tr}\left[\mathbf{Q}\frac{\partial^2 p(t,\mathbf{x})}{\partial \mathbf{x}\partial \mathbf{x}^T}\right].\end{aligned} \tag{2}$$

Given a state space region of interest at a particular decision time, t_d, which may be represented as a *loss function* by the decision maker, $L(\mathbf{x}_d,a_d)$, the expected loss of an action a_d is calculated as follows:

$$L(a_d) = \int L(\mathbf{x}_d,a_d)p(t_d,\mathbf{x}_d)d\mathbf{x}_d \tag{3}$$

where $\mathbf{x}_d$ is the state of the system at decision time, $t = t_d$.

If the FPKE in (2) can be solved exactly for the forecast pdf, $p(t_d,\mathbf{x}_d)$, it would be possible to obtain the expected loss and also find the optimal Bayesian decision [24], if a set of decisions exists. Although analytical steady state solutions for the FPKE exist for a limited class of dynamical systems, finding the solution for the generic nonlinear system in (1) is not a trivial task. In practice, we only know a numerical approximation to the state pdf $p(t_d,\mathbf{x}_d)$, denoted by $\hat{p}(t_d,\mathbf{x}_d)$. As a consequence of this, we can only compute an approximated value of the expected loss and hence optimal decision:

$$\hat{L}(a_d) = \int L(\mathbf{x}_d,a_d)\hat{p}(t_d,\mathbf{x}_d)d\mathbf{x}_d \tag{4}$$

$$\hat{a}_d = \arg\min_{a_d}\int L(\mathbf{x}_d,a_d)\hat{p}(t_d,\mathbf{x}_d)d\mathbf{x}_d. \tag{5}$$

The decision making process in the density forecasting context is presented in Fig. 1 (left). Obviously if we have a good approximation for the forecast pdf in the region of interest the same can be said for the expected loss. This situation becomes more dramatic when a large deviation exists between the actual and the estimated forecast pdf in the region of interest. In the case of evaluation of a single decision, the algorithm may underestimate the actual expected loss, $\hat{L}(a_d) \ll L(a_d)$, or overestimate it, misguiding the decision maker with respect to the magnitude of the situation. In the case when a optimal decision has to be chosen, the large

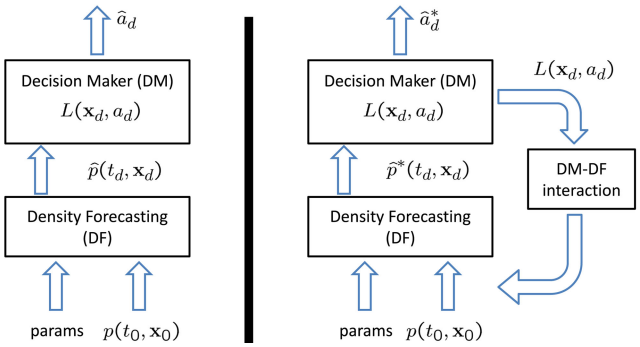


Fig. 1. Left figure represents the classic approach to decision making in the density forecasting context. The right figure shows the proposed model.

difference between forecast pdfs may result in picking not only a suboptimal decision but a very consequential one.

While one can derive a new method to approximate the forecast pdf by including the loss function in the derivation and overweighting errors in the region of interest to better approximate the expected loss, it will accomplish this at the expense of worsening the global approximation of the pdf. This will coarsen other estimates based on the forecast pdf, independent of the utility function, such as the mean of the pdf, the modes of the pdf, etc. The loss in global accuracy in estimating these statistics may end up misleading the decision maker with respect to the dominant behavior of the system.

In other words, if we name the computation of the expected loss of a given action as *impact assessment* and the computation of the moments and other quantities based on the pdf as *situation assessment*, one will require that both to be as accurate as possible. At the limit, if we can compute exactly the forecast pdf we accurately obtain both *impact assessment* and *situation assessment* since we can quantify exactly the probability of all the outcomes. The proposed decision-centric framework for density forecasting is in agreement with the information flow across the fusion levels of the JDL model proposed in [32]. Both an upward flow and a downward flow is necessary to obtain relevant inferences.

Since the decision maker holds important information regarding the use of the pdf obtained from the density forecasting method, we can incorporate this information in the uncertainty propagation process in a non-intrusive manner (do not have to derive a new method), by supplementing the inputs into the density forecasting module. The proposed method is shown in Fig. 1 (right), where a new interaction level is introduced between the decision maker and the uncertainty propagation, that uses the contextual information provided by the decision maker to supplement the inputs of the density forecasting process. In other words the proposed method changes the environment in which the density forecasting method is running.

Therefore, we want to find an approximation to the forecast pdf, $\hat{p}^*(t_d, \mathbf{x}_d)$, that addresses the interest held by the decision maker and provides both a better impact and situation assessment than $\hat{p}(t_d, \mathbf{x}_d)$. These objectives can be captured by the following two relations:

$$\int |p(t_d, \mathbf{x}_d) - \hat{p}^*(t_d, \mathbf{x}_d)|^2 d\mathbf{x}_d \leq \int |p(t_d, \mathbf{x}_d) - \hat{p}(t_d, \mathbf{x}_d)|^2 d\mathbf{x}_d \tag{6}$$

$$|\hat{L}^*(a_d) - L(a_d)| \leq |\hat{L}(a_d) - L(a_d)|. \tag{7}$$

In the present paper, we will design an interaction level between the decision maker and the uncertainty propagation module that approximates the pdf using a Gaussian mixture. The interaction level is adding new Gaussian components to the initial uncertainty, such that they will be positioned near the region of interest at the decision time. Their initial weights will be set to zero, thus the initial uncertainty is not changed, but the evolution of the weights is dictated by the error in the FPKE as in the Adaptive Gaussian Sum algorithm used to propagate the uncertainty in [36]. Thus if any probability density mass is moving naturally towards the region of interest, the weights of the new Gaussian components will become greater than zero. Therefore the method will find if there is any probability density mass in the region of interest.

In this paper we will consider only the forecast of the pdf when no measurements are available between the current time and the decision time. A suggestion, on how this can be used in the case when we have observations to assimilate between the current time and the decision time, is given in Section 4. In the following section we present the uncertainty propagation method for DF and in Section 4, the algorithm in the DM-DF interaction level is derived.

3. APPROXIMATION OF THE FORECAST PROBABILITY DENSITY FUNCTION

In this section, we briefly summarize the Gaussian mixture model approach to solve the FPKE; more details can be found in our prior work [30, 36]. The main idea of this approach is to approximate the state pdf by a finite sum of Gaussian density functions whose mean and covariance are propagated using linear theory. The weights corresponding to different Gaussian kernels are updated by requiring the mixture to satisfy the FPKE [36].

Let us consider the following equation depicting the Gaussian mixture model approximation for the forecast density function, $p(t, \mathbf{x})$:

$$\begin{aligned} \hat{p}(t, \mathbf{x}) &= \sum_{i=1}^{N} w_t^i \underbrace{\mathcal{N}(\mathbf{x}(t); \boldsymbol{\mu}_t^i, \mathbf{P}_t^i)}_{p_{g_i}} \\ \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_t^i, \mathbf{P}_t^i) &= |2\pi \mathbf{P}_t^i|^{-1/2} \\ &\quad \times \exp[-\tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_t^i)^T (\mathbf{P}_t^i)^{-1} (\mathbf{x} - \boldsymbol{\mu}_t^i)] \end{aligned} \tag{8}$$

where $\boldsymbol{\mu}_t^i$ and $\mathbf{P}_t^i$ represent the mean and covariance of the ith component of the Gaussian pdf, and w_t^i denotes the amplitude of ith Gaussian in the mixture. The positivity and normalization constraint on the mixture pdf, $\hat{p}(t, \mathbf{x})$, leads to following constraints on the amplitude vector:

$$\sum_{i=1}^{N} w_t^i = 1, \qquad w_t^i \geq 0, \qquad \forall t \tag{9}$$

In [2], it is shown that since all the components of the mixture pdf of (8) are Gaussian and thus, only estimates of their mean and covariance need to be maintained, they can be propagated between t and $t' = t + \Delta t$ using the linear system propagation methods such as the Extended Kalman Filter (EKF):

$$\dot{\boldsymbol{\mu}}_t^i = \mathbf{f}(t, \boldsymbol{\mu}_t^i) \tag{10}$$

$$\dot{\mathbf{P}}_t^i = \mathbf{A}_t^i \mathbf{P}_t^i + \mathbf{P}_t^i (\mathbf{A}_t^i)^T + \mathbf{Q} \tag{11}$$

$$\mathbf{A}_t^i = \left. \frac{\partial \mathbf{f}(t, \mathbf{x}(t))}{\partial \mathbf{x}(t)} \right|_{\mathbf{x}(t) = \boldsymbol{\mu}_t^i}. \tag{12}$$

Although, in this paper we present only the EKF model to propagate the mean and covariance of each of the Gaussian component, one can easily use some advanced linear propagation methods like unscented Kalman filter [13] or quasi-Gaussian Kalman filter [5] to propagate the mean and covariance more accurately.

The weights of the Gaussian components are not known and must be computed as part of the solution process. Using the following approximation for the total derivative of the weights, $\dot{w}_t^i = (1/\Delta t)(w_{t'}^i - w_t^i)$, the unknown weights $w_{t'}^i$ are found by minimizing the integral square FPKE error as discussed in [35, 36]. Substituting (8) in (2) leads to,

$$\begin{aligned} e(t, \mathbf{x}) &= \frac{\partial}{\partial t} \hat{p}(t, \mathbf{x}) + \frac{\partial \hat{p}(t, \mathbf{x})^T}{\partial \mathbf{x}} \mathbf{f}(t, \mathbf{x}) + \hat{p}(t, \mathbf{x}) \mathrm{Tr}\left[\frac{\partial \mathbf{f}(t, \mathbf{x})}{\partial \mathbf{x}} \right] \\ &\quad - \frac{1}{2} \mathrm{Tr}\left[\mathbf{Q} \frac{\partial^2 \hat{p}(t, \mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} \right] \\ &= \frac{1}{\Delta t} \sum_{i=1}^{N} p_{g_i} w_{t'}^i \\ &\quad + \sum_{i=1}^{N} \left(\frac{\partial p_{g_i}^T}{\partial \boldsymbol{\mu}_t^i} \dot{\boldsymbol{\mu}}_t^i + \mathrm{Tr}\left[\frac{\partial p_{g_i}}{\partial \mathbf{P}_t^i} \dot{\mathbf{P}}_t^i \right] - \frac{1}{\Delta t} p_{g_i} \right. \\ &\qquad + \frac{\partial p_{g_i}^T}{\partial \mathbf{x}} \mathbf{f}(t, \mathbf{x}) + p_{g_i} \mathrm{Tr}\left[\frac{\partial \mathbf{f}(t, \mathbf{x})}{\partial \mathbf{x}} \right] \\ &\qquad \left. - \frac{1}{2} \mathrm{Tr}\left[\mathbf{Q} \frac{\partial^2 p_{g_i}}{\partial \mathbf{x} \partial \mathbf{x}^T} \right] \right) w_t^i. \end{aligned} \tag{13}$$

Since the FPKE error of (13) is linear in Gaussian weights, the integral square FPKE error minimization problem can be written as the following quadratic pro-

gramming problem:

$$\begin{aligned} \min_{w^i_{t'}} \quad & \tfrac{1}{2}\mathbf{w}_{t'}^T\mathbf{M}_\mathbf{c}\mathbf{w}_{t'} + \mathbf{w}_{t'}^T\mathbf{N}_\mathbf{c}\mathbf{w}_t \\ \text{s.t} \quad & \mathbf{1}_{N\times 1}^T\mathbf{w}_{t'} = 1 \\ & \mathbf{w}_{t'} \geq \mathbf{0}_{N\times 1} \end{aligned} \tag{14}$$

where $\mathbf{w}_t \in \mathbb{R}^{N\times 1}$ is the vector of weights at time t, $\mathbf{w}_{t'} \in \mathbb{R}^{N\times 1}$ is the vector of unknown weights at time t', $\mathbf{1}_{N\times 1} \in \mathbb{R}^{N\times 1}$ is a vector of ones, $\mathbf{0}_{N\times 1} \in \mathbb{R}^{N\times 1}$ is a vector of zeros and the components of the two matrices $\mathbf{M}_\mathbf{c} \in \mathbb{R}^{N\times N}$ and $\mathbf{N}_\mathbf{c} \in \mathbb{R}^{N\times N}$ are given by

$$\begin{aligned} m_{c_{ij}} = & \frac{1}{\Delta t^2}|2\pi(\mathbf{P}_t^i + \mathbf{P}_t^j)|^{-1/2} \\ & \times \exp\left[-\frac{1}{2}(\boldsymbol{\mu}_t^i - \boldsymbol{\mu}_t^j)^T(\mathbf{P}_t^i + \mathbf{P}_t^j)^{-1}(\boldsymbol{\mu}_t^i - \boldsymbol{\mu}_t^j)\right] \\ & \qquad\qquad \text{for} \quad i \neq j \end{aligned} \tag{15}$$

$$m_{c_{ii}} = \frac{1}{\Delta t^2}|4\pi\mathbf{P}_t^i|^{-1/2} \qquad \text{for} \quad i = j \tag{16}$$

and,

$$\begin{aligned} n_{c_{ij}} = \frac{1}{\Delta t}p_{g_i}\int_V \Bigg(& \frac{\partial p_{g_j}^T}{\partial \boldsymbol{\mu}_t^j}\dot{\boldsymbol{\mu}}_t^j + \mathrm{Tr}\left[\frac{\partial p_{g_j}}{\partial \mathbf{P}_t^j}\dot{\mathbf{P}}_t^j\right] - \frac{1}{\Delta t}p_{g_j} \\ & + \frac{\partial p_{g_j}^T}{\partial \mathbf{x}}\mathbf{f}(t,\mathbf{x}) + p_{g_j}\mathrm{Tr}\left[\frac{\partial \mathbf{f}(t,\mathbf{x})}{\partial \mathbf{x}}\right] \\ & - \frac{1}{2}\mathrm{Tr}\left[\mathbf{Q}\frac{\partial^2 p_{g_j}}{\partial \mathbf{x}\partial \mathbf{x}^T}\right]\Bigg) d\mathbf{x}. \end{aligned} \tag{17}$$

Details on the derivation of the above relations can be found in [36, 38]. Notice that to carry out this minimization, we need to evaluate integrals involving Gaussian pdfs over volume V which can be computed exactly for polynomial nonlinearity and in general can be approximated by the Gaussian quadrature method. By updating the forecast weights, not only can we obtain a more accurate estimate but also a better approximation to the forecast probability density function [35].

The estimated pdf is used to compute the expected loss. We require that the loss function provided is positive, finite everywhere and it is able to distinguish the important states from the unimportant ones. For simplicity the loss function used in this work has the following form:

$$L(\mathbf{x}_d, a_d) = \mathcal{N}(\mathbf{x}_d; \boldsymbol{\mu}_L, \boldsymbol{\Sigma}_L). \tag{18}$$

Due to the approximations used in propagating the pdf it may happen that no or very little probability density mass exists in the region of interest at the decision time, depicted here by the loss function. In the following section we present an algorithm which adds new Gaussian kernels to the initial mixture such that they will be positioned in the region of interest defined by the loss function at the decision time, increasing the accuracy of the expected loss.

4. DECISION MAKER–DENSITY FORECASTING INTERACTION LEVEL

The iterative method proposed here, is adding a set of Gaussian components to the initial pdf that are sensitive to the loss function at the decision time. After propagation, these Gaussian components will be located near the center of support of the loss function at the decision time. Initially the weights of these components are set to zero, and they will be updated in the propagation step, using the method in Section 3, if any probability density mass is moving in their direction. The weights at the decision time will give their relative contributions in computing the expected loss with respect to the entire pdf.

An algorithm, called the Progressive Selection of Gaussian Components (PSGC), that bears similarity to the simulated annealing and the progressive correction used in particle filters [21], is proposed in selecting the initial Gaussian components sensitive to the loss function. The means of the new Gaussian components will be sampled from a proposed distribution, $p_{\mathrm{Smp}}(t_0, \mathbf{x}_0)$, which is recursively constructed to be sensitive to the contextual loss function. The support of the proposal distribution or sampling pdf is gradually mapped into a region that covers the support of the loss function at decision time.

The main idea in constructing the sampling pdf is as follows: initially set the sampling pdf equal to the uncertain initial condition in (1), select the means and covariances of a set of Gaussian components based on this distribution, propagate each one of them using the time update equations in the Extended Kalman Filter, Eqs. (10)–(11), until the decision time is reached, and based on the contributions to the expected loss find their corresponding weights. The new sampling pdf is just the weighted sum of the of the initially selected Gaussian components. The sampling process is repeated until all the Gaussian components are located in the support region of the loss function at the decision time. The remainder of the section details the derivation of this procedure.

Let initially the sample pdf, $p_{\mathrm{Smp}}(t_0, \mathbf{x}_0)$, to be equal to the initial uncertainty given by $p(t_0, \mathbf{x}_0)$, which is modeled using a Gaussian sum as in (8). Compute the mean and the variance of the sample pdf:

$$\boldsymbol{\mu}_0 = E[\mathbf{x}_0] = \int \mathbf{x}_0 p_{\mathrm{Smp}}(t_0, \mathbf{x}_0) d\mathbf{x}_0 \tag{19}$$

$$\begin{aligned} \mathbf{P}_0 &= E[(\mathbf{x}_0 - \boldsymbol{\mu}_0)(\mathbf{x}_0 - \boldsymbol{\mu}_0)^T] \\ &= \int (\mathbf{x}_0 - \boldsymbol{\mu}_0)(\mathbf{x}_0 - \boldsymbol{\mu}_0)^T p_{\mathrm{Smp}}(t_0, \mathbf{x}_0) d\mathbf{x}_0. \end{aligned} \tag{20}$$

For the first iteration the above two moments are computed as follows:

$$\boldsymbol{\mu}_0 = \sum_{i=1}^{N} w_0^i \boldsymbol{\mu}_0^i \tag{21}$$

$$\mathbf{P}_0 = \sum_{i=1}^{N} w_0^i [\mathbf{P}_0^i + (\boldsymbol{\mu}_0^i - \boldsymbol{\mu}_0)(\boldsymbol{\mu}_0^i - \boldsymbol{\mu}_0)^T]. \tag{22}$$

Assume that we want to add another M new Gaussian components to the initial pdf with zero weights and sensitive to the loss function. We sample the means of these Gaussian components from the proposal distribution such that their equally weighted sum gives the mean in (21).

$$\boldsymbol{\mu}^i \sim p_{\text{Smp}}(t_0, \mathbf{x}_0) \qquad \text{for} \quad i = 1 \ldots M-1 \tag{23}$$

$$\boldsymbol{\mu}^M = M\boldsymbol{\mu}_0 - \sum_{i=1}^{M-1} \boldsymbol{\mu}^i. \tag{24}$$

The default covariance of the Gaussian components is $\mathbf{D}$. We want to find the new covariance $\mathbf{D}^*$ such that the covariance of the new Gaussian components matches the covariance of the sample pdf, $\mathbf{P}_0$. Let $\mathbf{D}^* = \gamma\mathbf{D}$. Thus we want to find γ such that we minimize the following expression:

$$J_\gamma = \text{Tr}\left[\mathbf{P}_0 - \frac{1}{M}\sum_{i=1}^{M}(\gamma\mathbf{D} + (\boldsymbol{\mu}^i - \boldsymbol{\mu}_0)(\boldsymbol{\mu}^i - \boldsymbol{\mu}_0)^T)\right] \tag{25}$$

$$\gamma = \frac{1}{\text{Tr}[\mathbf{D}]}\text{Tr}\left[\mathbf{P}_0 - \frac{1}{M}\sum_{i=1}^{M}(\boldsymbol{\mu}^i - \boldsymbol{\mu}_0)(\boldsymbol{\mu}^i - \boldsymbol{\mu}_0)^T\right]. \tag{26}$$

Only solutions $\gamma > 0$ are accepted. Otherwise we repeat the sampling of the means, starting with (23). Once we have the first two moments of the new Gaussian components, we propagate them using the time update equations in the Extended Kalman Filter, Eqs. (10)–(11), until we reach the decision time. Let $\boldsymbol{\mu}_d^i$ and $\mathbf{P}_d^i$ be their means and covariances at the decision time, $t = t_d$. The Gaussian components will then be weighted based on their contribution to the expected loss. A larger contribution means a more sensitive component to the loss function, thus a larger weight.

To be able to compute the weights of the Gaussian components, make sure that all of them are fairly weighted, we are not running into numerical problems and also create an indicator to mark the end of the algorithm, we compute an inflation coefficient for the loss function. Let $\boldsymbol{\Sigma}_L^* = \alpha\boldsymbol{\Sigma}_L$ be the inflated covariance of the loss function.

The inflation coefficient α is found such that the expected loss computed using the most distant Gaussian component from the loss function is maximized. Let the mean and the covariance of the most distant component be denoted by $\boldsymbol{\mu}_d^{\max}$ and $\mathbf{P}_d^{\max}$ respectively.

$$\begin{aligned} J_{\max} &= \int \mathcal{N}(\mathbf{x}_d; \boldsymbol{\mu}_L, \alpha\boldsymbol{\Sigma}_L)\mathcal{N}(\mathbf{x}_d; \boldsymbol{\mu}_d^{\max}, \mathbf{P}_d^{\max})d\mathbf{x}_d \\ &= \mathcal{N}(\boldsymbol{\mu}_L; \boldsymbol{\mu}_d^{\max}, \alpha\boldsymbol{\Sigma}_L + \mathbf{P}_d^{\max}). \end{aligned} \tag{27}$$

An equivalent way to seek α is by minimizing the negative logarithm of the above expectation.

$$\begin{aligned} J_{\min} &= \log[\det(\alpha\boldsymbol{\Sigma}_L + \mathbf{P}_d^{\max})] \\ &\quad + (\boldsymbol{\mu}_L - \boldsymbol{\mu}_d^{\max})^T(\alpha\boldsymbol{\Sigma}_L + \mathbf{P}_d^{\max})^{-1}(\boldsymbol{\mu}_L - \boldsymbol{\mu}_d^{\max}) \end{aligned} \tag{28}$$

Let us denote $\mathbf{K} = \alpha\boldsymbol{\Sigma}_L + \mathbf{P}_d^{\max}$ and $\mathbf{U} = (\boldsymbol{\mu}_L - \boldsymbol{\mu}_d^{\max}) \cdot (\boldsymbol{\mu}_L - \boldsymbol{\mu}_d^{\max})^T$. We seek $\alpha > 0$ such that

$$\frac{\partial J_{\min}}{\partial \alpha} = 0 \tag{29}$$

$$\text{Tr}[\mathbf{K}^{-1}\boldsymbol{\Sigma}_L - \mathbf{K}^{-1}\mathbf{U}\mathbf{K}^{-1}\boldsymbol{\Sigma}_L] = 0. \tag{30}$$

After a few mathematical manipulations, (30) can be written in the following format:

$$\text{Tr}[\mathbf{K}^{-1}\boldsymbol{\Sigma}_L(\alpha\mathbf{I} + \mathbf{P}_d^{\max}\boldsymbol{\Sigma}_L^{-1} - \mathbf{U}\boldsymbol{\Sigma}_L^{-1})\mathbf{K}^{-1}\boldsymbol{\Sigma}_L] = 0. \tag{31}$$

Using the following notation, $\mathbf{A} = \mathbf{K}^{-1}\boldsymbol{\Sigma}_L$ and $\mathbf{B} = \alpha\mathbf{I} + \mathbf{P}_d^{\max}\boldsymbol{\Sigma}_L^{-1} - \mathbf{U}\boldsymbol{\Sigma}_L^{-1}$, (31) can be written as Tr[**ABA**] = 0. Observe that for $\alpha > 0$ the matrix $\mathbf{A}$ is symmetric and positive definite. Hence, by applying Lemma 1 from Appendix A to (31) we get,

$$\text{Tr}[\alpha\mathbf{I} + \mathbf{P}_d^{\max}\boldsymbol{\Sigma}_L^{-1} - \mathbf{U}\boldsymbol{\Sigma}_L^{-1}] = 0. \tag{32}$$

Therefore we accept solutions $\alpha > 1$ that satisfy the following relation

$$\alpha = \frac{1}{n}\text{Tr}[(\mathbf{U} - \mathbf{P}_d^{\max})\boldsymbol{\Sigma}_L^{-1}]. \tag{33}$$

For $\alpha \leq 1$ we stop the algorithm, because all the Gaussian components, including the most distant one, are located near the center of support of the loss function. Otherwise, α is used to compute the inflated covariance $\boldsymbol{\Sigma}_L^* = \alpha\boldsymbol{\Sigma}_L$, and the weights of the Gaussian components are obtained based on their approximation to the loss function by solving the following optimization problem:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \frac{1}{2}\int \left(\mathcal{N}(\mathbf{x}_d; \boldsymbol{\mu}_L, \boldsymbol{\Sigma}_L^*) - \sum_{i=1}^{M} w^i \mathcal{N}(\mathbf{x}_d; \boldsymbol{\mu}_d^i, \mathbf{P}_d^i)\right)^2 d\mathbf{x}_d. \tag{34}$$

The optimization in (34) is equivalent to solving the following quadratic programming problem:

$$\begin{aligned} \mathbf{w} &= \arg\min_{\mathbf{w}} \tfrac{1}{2}\mathbf{w}^T\mathbf{M}\mathbf{w} - \mathbf{w}^T\mathbf{N} \\ &\text{subject to} \quad \mathbf{1}_{M\times 1}^T\mathbf{w} = 1 \\ &\mathbf{w} \geq \mathbf{0}_{M\times 1} \end{aligned} \tag{35}$$

where $\mathbf{w} \in \mathbb{R}^{M\times 1}$ is the vector of weights and the entries of $\mathbf{M} \in \mathbb{R}^{M\times M}$ and $\mathbf{N} \in \mathbb{R}^{M\times 1}$ are given by:

$$m_{ij} = \mathcal{N}\{\boldsymbol{\mu}_d^j; \boldsymbol{\mu}_d^i, \mathbf{P}_d^i + \mathbf{P}_d^j\} \tag{36}$$

$$n_i = \mathcal{N}\{\boldsymbol{\mu}_L; \boldsymbol{\mu}_d^i, \mathbf{P}_d^i + \boldsymbol{\Sigma}_L^*\}. \tag{37}$$

Given the weights found in (35), the new pdf used to sample the new means is given by,

$$p_{\text{Smp}}(t_0, \mathbf{x}_0) = \sum_{i=1}^{M} w^i \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}^i, \beta\mathbf{D}^*) \tag{38}$$

where $\beta \leq 1$ is a coefficient that controls the decrease of the initial variance. If α has decreased from the previous iteration this means that the Gaussian components are getting closer to the loss function and therefore we can decrease the variance of the initial distribution to finely tune the position of the Gaussian components, otherwise $\beta = 1$. The process is repeated starting with (19), only this time the first two moments in Eqs. (21)–(22), are computed using the M components that construct the sample pdf in (38):

$$\boldsymbol{\mu}_0 = \sum_{i=1}^{M} w^i \boldsymbol{\mu}^i \tag{39}$$

$$\mathbf{P}_0 = \sum_{i=1}^{M} w^i [\beta\mathbf{D}^* + (\boldsymbol{\mu}^i - \boldsymbol{\mu}_0)(\boldsymbol{\mu}^i - \boldsymbol{\mu}_0)^T]. \tag{40}$$

If $\alpha < 1$ or the maximum number of time steps has been reached, then the algorithm is stopped and the new initial Gaussian mixture is obtained as follows:

$$\begin{aligned} p_{\text{NEW}}(t_0, \mathbf{x}_0) &= p(t_0, \mathbf{x}_0) + \sum_{j=1}^{M} 0 \times \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}^j, \beta\mathbf{D}^*) \\ &= \sum_{i=1}^{N} w_0^i \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_0^j, \mathbf{P}_0^i) + \sum_{j=1}^{M} 0 \times \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}^j, \beta\mathbf{D}^*). \end{aligned} \tag{41}$$

The entire algorithm to select the Gaussian components is presented in Table I and graphical illustrations are presented in Fig. 2. In the case of multiple loss functions, the algorithm is run once for each one of the loss functions, creating sets of initial Gaussian components sensitive to their loss function.

While not the scope of this paper, the above method can also be applied when measurements are available between the current time and the decision time. The PSGC algorithm will be applied every time a measurement has been assimilated and the a posteriori pdf has been found. The drawback of this procedure is that the number of Gaussian components will increase linearly with the number of measurements. Instead of adding new Gaussian components, a better way to deal with this situation is to allocate from the total of N Gaussian components, M which are designated to be sensitive to the loss function and the rest $M - N$ to capture the dominant evolution of the pdf.

The Decision-Centric Density Forecasting is obtained by running first the PSGC algorithm, derived in this section, to supplement the initial uncertainty with M new Gaussian components sensitive to the loss function. The new initial Gaussian mixture obtained, (41), is then propagated using the Adaptive Gaussian Sum algorithm presented in Section 3. The following section presents the application of the Decision-Centric Density Forecasting method to a toxic cloud transported by wind and a low dimensional numerical example where a number of performance measures are computed.

ALGORITHM 1 *Progressive Selection of Gaussian Components*

Require: t_d—decision time
$p(t_0, \mathbf{x}_0) = \sum_{i=1}^{N} w_0^i \mathcal{N}(\mathbf{x}_0; \boldsymbol{\mu}_0^j, \mathbf{P}_0^i)$—initial probability density function
M—number of extra Gaussian components
$\mathbf{D}$—default Gaussian component covariance
w_{tol}—add only Gaussian components with weights greater than this threshold
$L(\mathbf{x}_d) = \mathcal{N}\{\mathbf{x}_d; \boldsymbol{\mu}_L, \boldsymbol{\Sigma}_L\}$—loss function
maxiter—maximum number of iterations

$p_{\text{Smp}}(t_0, \mathbf{x}_0) = p(t_0, \mathbf{x}_0)$, $\alpha = \infty$, $\gamma = -1$
while ($\alpha > 1$) & maxiter **do**
  The mean and the covariance of the sample pdf, if first iteration (21)–(22), otherwise (39)–(40)
  $\boldsymbol{\mu}_0 = E[\mathbf{x}_0] = \int \mathbf{x}_0 p_{\text{Smp}}(t_0, \mathbf{x}_0) d\mathbf{x}_0$
  $\mathbf{P}_0 = E[(\mathbf{x}_0 - \boldsymbol{\mu}_0)(\mathbf{x}_0 - \boldsymbol{\mu}_0)^T]$
    $= \int (\mathbf{x}_0 - \boldsymbol{\mu}_0)(\mathbf{x}_0 - \boldsymbol{\mu}_0)^T p_{\text{Smp}}(t_0, \mathbf{x}_0) d\mathbf{x}_0$
  while ($\gamma < 0$) **do**
    Get the means of the Gaussian components
    Draw $\boldsymbol{\mu}^i \sim p_{\text{Smp}}(t_0, \mathbf{x}_0)$ for $i = 1 \ldots M-1$
    Set $\boldsymbol{\mu}^M = M\boldsymbol{\mu}_0 - \sum_{i=1}^{M-1} \boldsymbol{\mu}^i$
    $\gamma = \dfrac{1}{\text{Tr}[\mathbf{D}]} \text{Tr}\left[\mathbf{P}_0 - \dfrac{1}{M} \sum_{i=1}^{M} (\boldsymbol{\mu}^i - \boldsymbol{\mu}_0)(\boldsymbol{\mu}^i - \boldsymbol{\mu}_0)^T\right]$
  end while
  Get the covariance of the Gaussian components
  $\mathbf{P}_0^i = \mathbf{D}^* = \gamma\mathbf{D}$
  Propagate the moments from $t = t_0$ to $t = t_d$
  $\dot{\boldsymbol{\mu}}_t^i = \mathbf{f}(t, \boldsymbol{\mu}_t^i)$
  $\dot{\mathbf{P}}_t^i = \mathbf{A}_t^i \mathbf{P}_t^i + \mathbf{P}_t^i (\mathbf{A}_t^i)^T + \mathbf{Q}$
  Get the most distant component by computing the Mahanalobis distance
  $d_i = (\boldsymbol{\mu}_L - \boldsymbol{\mu}_d^i)^T (\mathbf{P}_d^i + \boldsymbol{\Sigma}_L)^{-1} (\boldsymbol{\mu}_L - \boldsymbol{\mu}_d^i)$
  $\boldsymbol{\mu}_d^{\max}, \mathbf{P}_d^{\max} = \arg\max(d_i)$
  Compute optimal α and the inflated matrix Σ_L^*
  $\alpha = \dfrac{1}{n} \text{Tr}[((\boldsymbol{\mu}_d^{\max} - \boldsymbol{\mu}_L)(\boldsymbol{\mu}_d^{\max} - \boldsymbol{\mu}_L)^T - \mathbf{P}_d^{\max}) \Sigma_L^{-1}]$
  if $\alpha < 1$ **then** $\alpha = 1$ **end if**
  $\Sigma_L^* = \alpha \Sigma_L$
  Elements of $\mathbf{M} \in \mathbb{R}^{M\times M}$ and $\mathbf{N} \in \mathbb{R}^{M\times 1}$
  $m_{ij} = \mathcal{N}\{\boldsymbol{\mu}_d^j; \boldsymbol{\mu}_d^i, \mathbf{P}_d^i + \mathbf{P}_d^j\}$
  $n_i = \mathcal{N}\{\boldsymbol{\mu}_L; \boldsymbol{\mu}_d^i, \mathbf{P}_d^i + \boldsymbol{\Sigma}_L^*\}$
  Compute the weights
  $\mathbf{w} = \arg\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{M} \mathbf{w} - \mathbf{w}^T \mathbf{N}$
    subject to $\mathbf{1}_{M\times 1}^T \mathbf{w} = 1$ and $\mathbf{w} \geq \mathbf{0}_{M\times 1}$

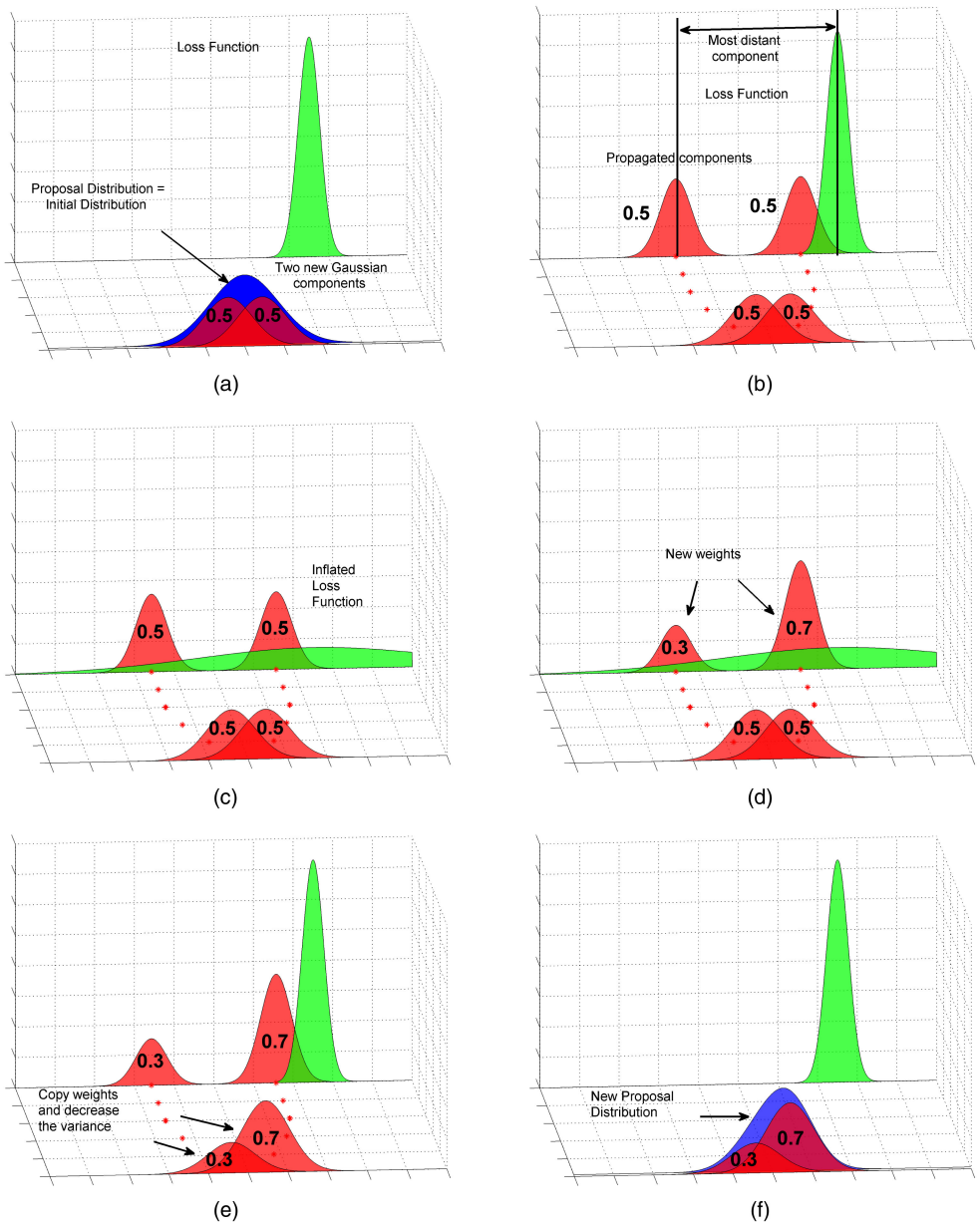


Fig. 2. Illustration for the PSGC algorithm. (a) Steps 1–8. (b) Steps 9–19. (c) Steps 11–12. (d) Steps 13–14. (e) Steps 15–16. (f) Goto Step 3 if $\sigma > 1$ and maximum number of iterations not reached.

15: **if** α is getting smaller **then** choose $\beta < 1$
else $\beta = 1$ **end if**
16: Set $p_{\text{Smp}}(t_0, \mathbf{x}_0) = \sum_{j=1}^{M} w^j \mathcal{N}\{\mathbf{x}_0; \boldsymbol{\mu}^j, \beta \mathbf{D}^*\}$
17: **end while**
18: Set $p_{\text{NEW}}(t_0, \mathbf{x}_0) = p(t_0, \mathbf{x}_0) + \sum_{j=1}^{M, w^j \geq w_{\text{tol}}} 0$
$\times \mathcal{N}\{\mathbf{x}_0; \boldsymbol{\mu}^j, \beta \mathbf{D}^*\}$
19: **return** $p_{\text{NEW}}(t_0, \mathbf{x}_0)$

5. NUMERICAL RESULTS

Chemical, Biological, Radiological, and Nuclear (CBRN) incidents are rare events but very consequential, which mandates extensive research and operational efforts in mitigating their outcomes. Many puff dispersion models, such as SCIPUFF [33] and RIMPUFF [22], try to model the atmospheric transport and diffusion of toxic plumes. Similarly, BIGFLOW [1] can be used to analyze the contaminant transport problem in the nonlinear porous media. While inherently stochastic and highly nonlinear, these mathematical models are able to capture just a part of the dynamics of the real phenomenon and the forward integration yields an uncertain prediction. The decision maker takes actions based on the expected loss computed using both the predicted uncertainty and the loss function, which here maps a region of interest in the state space into a threat level, such as the population density in a town. Thus the ability to propagate the uncertainty and errors throughout the dynamic system is of great importance. As mentioned previously, the present method can be also ap-

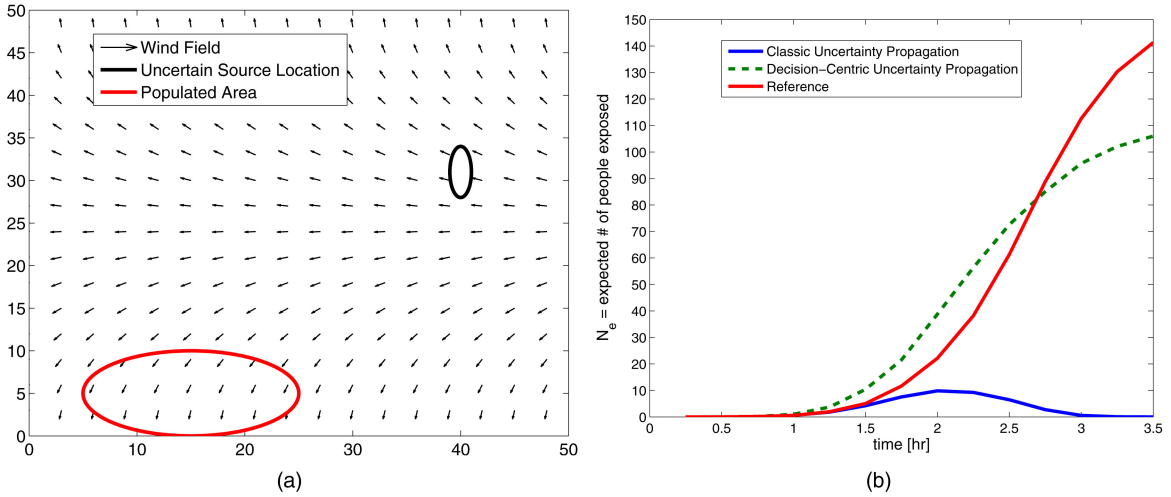


Fig. 3. (a) Chemical release scenario. (b) Evolution of number of people exposed.

plied for nonlinear filtering and smoothing problem which are relevant for source determination (localization and characterization) in the case of covert releases [3, 6, 34].

5.1. Example 1

To motivate the concept of incorporating contextual information into the uncertainty propagation algorithm, we consider the following noise driven nonlinear dynamic system that simulates the advection and the dispersion of a chemical material released from an uncertain location. A similar system has been previously used in [26, 34].

The instantaneous amount of material released is represented using a Gaussian-shaped puff, states of which evolve using the following equations, which describe a wind pattern as shown in Fig. 3(a).

$$\begin{aligned}\dot{x}(t) &= -a\sin(by(t)) + w_1(t)\\ \dot{y}(t) &= -a\cos(by(t)) + w_2(t)\\ \dot{s}(t) &= a\end{aligned} \tag{42}$$

where (x,y) is the position of the center of the puff, and the downwind distance from the source $s(t_k) = s_k$ is used to compute the puff radius at time $t = t_k$,

$$\sigma_k = p_y s_k^{q_y}. \tag{43}$$

Due to the simplicity of the model and the lack of knowledge about the initial conditions, release location in (44), the model forecast tends to become less accurate for longer simulations.

$$\begin{aligned}&p(t_0, x(t_0), y(t_0))\\ &\quad = \mathcal{N}([x,y]^T; [40,31]^T, \mathrm{diag}\{[1,9]^2\})\\ &s(t_0) = 0.\end{aligned} \tag{44}$$

The puff radius depends on meteorological conditions specified by the Karlsruhe-Jülich diffusion coefficients [22] which are set to $p_y = 0.466$, $q_y = 0.866$. The wind speed is considered to be $a = 10$ mph and the variable b depends on the boundaries of the domain and is set here to $\pi/50$. The process noise, $[w^1(t), w^2(t)]^T$, is a vector whose components are independent Gaussian white noise processes induced in the process model due to the uncertainty in the wind field. The autocorrelation function of the process noise is given by $\mathbf{Q} = 2\mathbf{I}_{2\times 2}\delta(t-\tau)$.

The concentration at each grid point, at time $t = t_k$, is computed using the following relation,

$$C_k(x_g, y_g) = \frac{M}{2\pi\sigma_k^2}\exp\left(-\frac{(x_k - x_g)^2 + (y_k - y_g)^2}{2\sigma_k^2}\right) \tag{45}$$

where $x_k = x(t_k)$, $y_k = y(t_k)$, and the instantaneous mass released is $M = 10$ kg.

These equations capture the main characteristics of puff-based dispersion models for a particular wind field. The weights of different Gaussian components have been updated every $\Delta t = 0.25$ hr using the error in the FPKE. The impulse chemical release is done in a region of 50×50 sqmi and the total simulation time is 3.5 hrs. The source location and its uncertainty as well as the decision region of interest, here the populated area with 10,000 residents, represented with a Gaussian function, $D(x,y)$ in (46), are shown in Fig. 3(a).

$$D(x,y) = 10{,}000 \times \mathcal{N}([x,y]^T; [15,5]^T, \mathrm{diag}\{[10,5]^2\}). \tag{46}$$

First, we propagate the uncertainty using the first-order Taylor expansion. We will call this method Classic Uncertainty Propagation. To evaluate the effect of the uncertainty propagation in the process of decision making, we compute the probability of the chemical concentration exceeding a critical value, $c_t = 0.0001$, which is assumed to be harmful. The decision maker may decide to evacuate or not the populated area, based on the evolution of this probability or hazard map and the number of people placed at risk due to exposure above

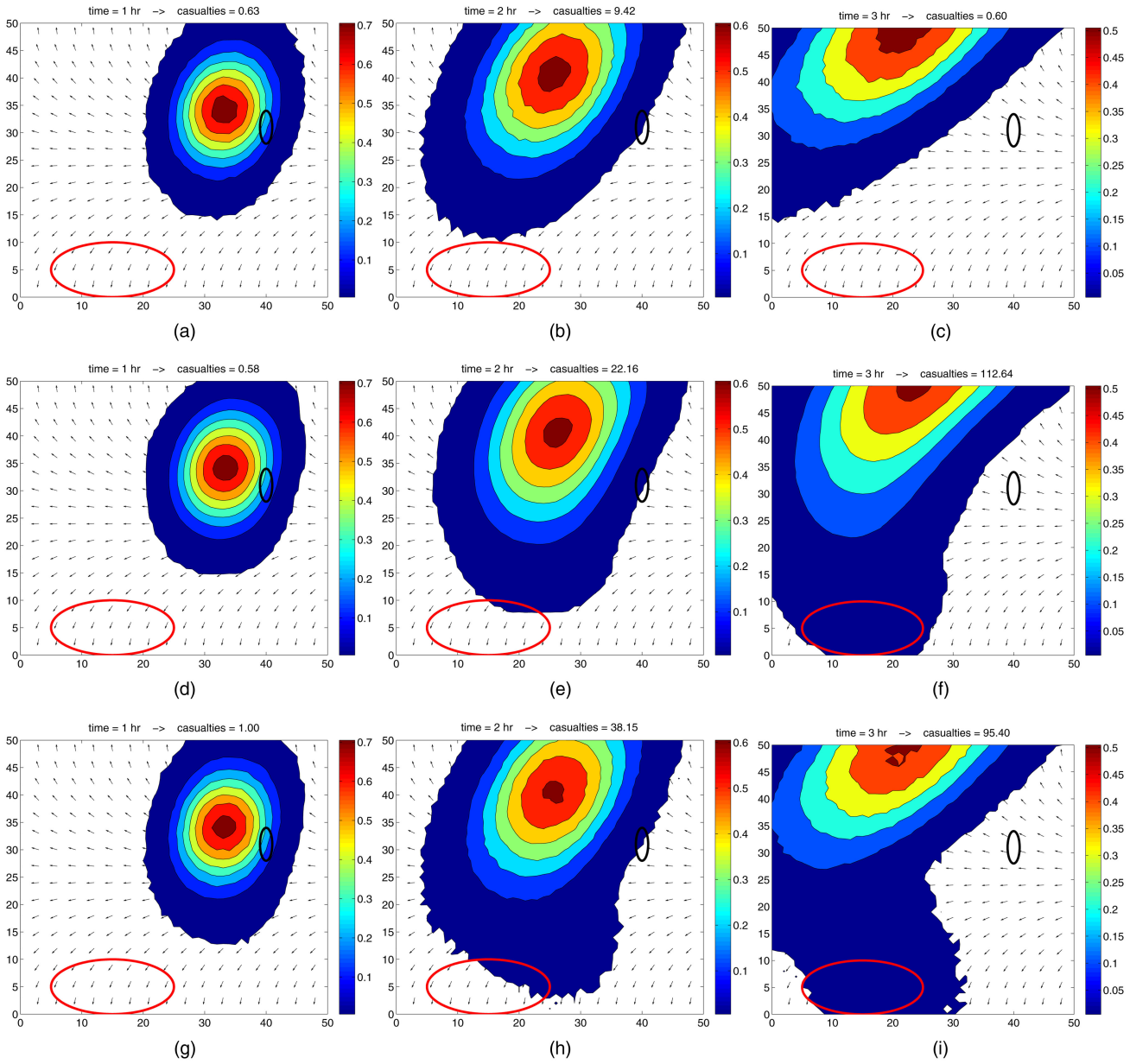


Fig. 4. Hazard maps: Probability evolution of chemical concentrations exceeding the critical value. (a) Classic uncertainty propagation after 1 hr. (b) Classic uncertainty propagation after 2 hrs. (c) Classic uncertainty propagation after 3 hrs. (d) Reference after 1 hr (Monte Carlo simulation). (e) Reference after 2 hrs (Monte Carlo simulation). (f) Reference after 3 hrs (Monte Carlo simulation). (g) Decision-centric uncertainty propagation after 1 hr. (h) Decision-centric uncertainty propagation after 2 hrs. (i) Decision-centric uncertainty propagation after 3 hrs.

safe concentration thresholds. Fig. 4(a), (b), (c) shows the probability evolution of the chemical concentration exceeding the consequential value for the Classic Uncertainty Propagation.

As reference, we use a Monte Carlo simulation, using 5,000 samples to evaluate as close as possible the probability of consequential concentrations. The evolution of the probability is presented in Fig. 4(d), (e), (f). We observe that in reality, consequential chemical concentrations are well into the populated area after 2 hrs. We refer to this method as Reference.

We apply the method presented in this paper to generate at most 5 new Gaussian components to be added to the initial condition. Their means and variances are returned by the PSGC algorithm, Algorithm 1. The initial weights of the new Gaussian components have been set to zero. The default value for the β coefficient is 0.9 and Gaussian components are included only if their weights are greater than $w_{\text{tol}} = 10^{-3}$. The label used for this method is Decision-Centric Uncertainty Propagation and its corresponding hazard map is presented in Fig. 4(g), (h), (i).

By accounting for the populated region, we are able to track the probability that consequential chemical concentrations are reaching that region. The expected number of people exposed to critical concentrations at dif-

ferent times are presented in Fig. 3(b) and is computed using (47). The classic uncertainty propagation is underestimating the magnitude of the situation, misguiding the decision maker to make a consequential decision such as not evacuating the region.

$$N_e = \int D(x,y)P(C_k(x,y) \geq c_t)dx\,dy. \tag{47}$$

Our method, while still using the same principles of first-order Taylor expansion, is able to capture the probability density mass in the region of interest by adding Gaussian components that are sensitive to this area and to estimate the expected number of people exposed in the same order of magnitude as the Reference.

While this example presents a particular wind pattern, the method can also be applied in more realistic scenarios using the Lagrangian puff atmospheric dispersion model RIMPUFF [22] in connection with any wind forecasting module such as WRF or MM5 [11].

The next example evaluates the performance of the decision centric forecasting method against a number of performance measures.

5.2. Example 2

To better illustrate the steps of the proposed method, as well as to evaluate its performance against a number of performance measures, we also consider the following low dimensional continuous-time dynamic system with uncertain initial condition given by:

$$\begin{aligned} &\dot{x} = \sin(x) + \Gamma(t) \qquad \text{where} \quad Q = 1 \\ &p(t_0,x_0) = \mathcal{N}(x_0; -0.3, 0.3^2). \end{aligned} \tag{48}$$

The noise free dynamic system in (48) is of particular interest since it exhibits chaotic behavior caused by multiple equilibrium states. The state space region of interest is depicted by the following loss function, and the time of decision is at $t_d = 8$ sec.

$$L(x_d) = \mathcal{N}\left(x_d; \frac{\pi}{2}, 0.1^2\right). \tag{49}$$

First we compute an accurate numerical solution based on the discretization of the FPKE, and this will stand as the reference probability density function. The performance measures for this method will be labeled as REF. The evolution of the pdf using this method can be seen in Fig. 5(a).

Three other approximations for the pdf are provided including the method presented in this paper. The first approximation propagates the initial uncertainty using the Extended Kalman Filter time update equations, Eqs. (10)–(11), labeled later as EKF. The evolution of the pdf for this method is presented in Fig. 5(b).

For the next approximation method, we add 5 Gaussian components to the initial uncertainty, creating a Gaussian mixture with 6 components. The means of the new components are just the result of back propagation (from $t_d = 8$ sec to $t_0 = 0$ sec) of 5 equidistant samples taken in the 3 sigma bound of the loss function support. The variance of the new components is set to 10^{-10} and their initial weights are set to zero. The label used for this method is GS_BCK and the evolution of the pdf is shown in Fig. 5(c). While all the means of the new Gaussian components are positioned in the loss function support region, their variances get large and the probability density mass in that region is difficult to be visualized.

TABLE 1
Performance Measures—500 Monte Carlo Runs

	$\hat{L}_d$	$\hat{R}_{\text{err}}$	ISD	WISD
REF	0.0332	N/A	N/A	N/A
EKF	4.93E-09	1.0000	0.1840	0.0015
GS_BCK	0.0001	0.9968	0.0536	0.0015
GS_DEC (mean)	0.0256	0.2300	0.0470	0.0004
GS_DEC: Percentile Table—500 Observations				
Percent	$\hat{L}_d$	$\hat{R}_{\text{err}}$	ISD	WISD
0.0%	0.0010	0.0151	0.0368	0.0002
5.0%	0.0142	0.0230	0.0378	0.0003
10.0%	0.0177	0.0271	0.0380	0.0003
25.0%	0.0229	0.0566	0.0387	0.0003
50.0%	0.0257	0.2270	0.0491	0.0003
75.0%	0.0313	0.3090	0.0514	0.0004
90.0%	0.0323	0.4670	0.0574	0.0006
95.0%	0.0324	0.5710	0.0601	0.0007
100.0%	0.0327	0.9700	0.0705	0.0014

We apply the method presented in this paper to generate at most 5 new Gaussian components to be added to the initial condition. Their means and variances are returned by the progressive selection algorithm, Algorithm 1. The initial weights of the new Gaussian components have been set to zero. The default value for the β coefficient is 0.9 and Gaussian components are included only if their weights are greater than $w_{\text{tol}} = 10^{-3}$. The label used for this method is GS_DEC and its corresponding pdf is presented in Fig. 5(d).

The evolution the Gaussian components for the last two methods is also achieved using the Extended Kalman Filter time update equations, but it is interrupted every $\Delta t = 0.5$ sec to adjust the weights of different Gaussian components using the optimization in (14).

The following performance measures have been computed for the methods used in the experiment:

$$\hat{L}_d = \int L(x)\hat{p}(t_d,x_d)dx_d \tag{50}$$

$$\hat{R}_{\text{err}} = \frac{1}{L_d}|L_d - \hat{L}_d| \tag{51}$$

$$\text{ISD} = \int |p(t_d,x_d) - \hat{p}(t_d,x_d)|^2 dx_d \tag{52}$$

$$\text{WISD} = \int L(x)|p(t_d,x_d) - \hat{p}(t_d,x_d)|^2 dx_d. \tag{53}$$

In Fig. 5(e) the forecast pdf is plotted at time t_d for all the methods, for a particular Monte Carlo run. Our method, GS_DEC, is able to better estimate the probability density mass in the region of interest.

In Table I, we present the performance measures after 500 Monte Carlo runs. The expected loss given

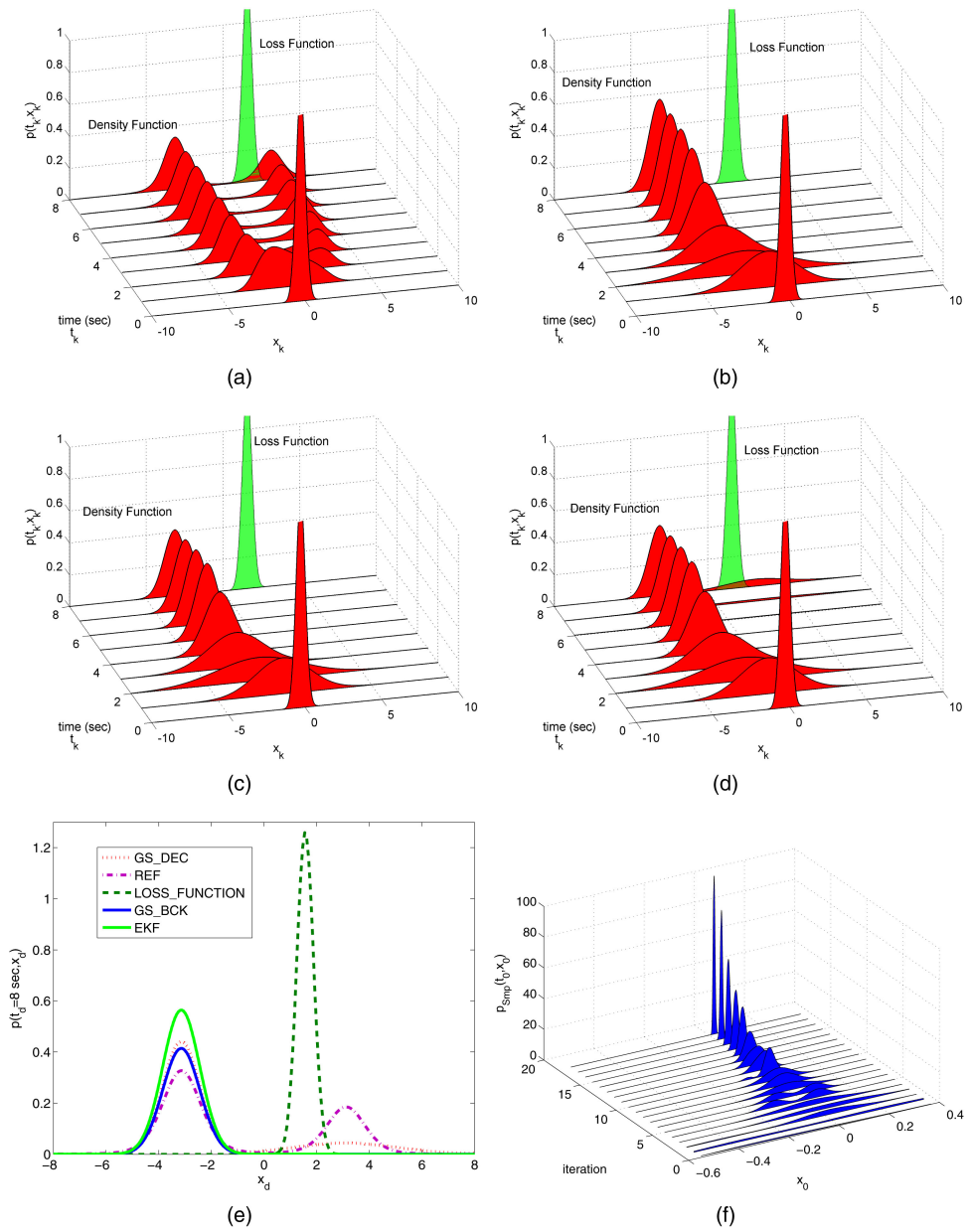


Fig. 5. The evolution of the forecast pdf and the sampling pdf. (a) REF: Numerical approximation FPKE. (b) EKF: first order Taylor expansion approximation. (c) GS_BCK: back propagated means. (d) GS_DEC: progressive selection of Gaussian components. (e) Probability density function at f_d = 8 sec. (f) The evolution of the pdf, $ps_{mp}(t_0, \mathbf{x}_0)$, used to sample the means of the Gaussian components.

by the GS_DEC method is consistently better over all the Monte Carlo runs than the EKF and the GS_BCK method. We are also able to consistently give an overall better approximation to the pdf and in the region of interest than the EKF method, which justifies the use of this method. Compared with the GS_BCK we do a better job on average in approximating the pdf which suggests that there is a trade off in selecting the Gaussian components regarding their means and variances.

In Fig. 5(f) it is plotted the evolution of the pdf, $p_{\text{Smp}}(t_0, \mathbf{x}_0)$, used to sample the means of the new Gaussian components for a particular Monte Carlo run. The pdf used in the first iteration is our initial uncertainty and we see how it converges, as the number of iterations increases, to a particular region in the state space that is sensitive to the loss function at the decision time.

6. CONCLUSIONS AND FUTURE WORK

A decision-centric view to create an interaction level between the decision maker and the density forecasting module has been designed, such that we can incorporate contextual information held by the decision maker into the uncertainty propagation process to better approximate the probability density function and the expected loss value.

The Progressive Selection of Gaussian Components algorithm is run once at the beginning of the simulation

to supplement the initial uncertainty with new Gaussian components that are sensitive to the loss function at the decision time. The weights of all the Gaussian components are then updated during the propagation based on the error in the Fokker-Planck-Kolmogorov Equation. This way we obtain not only a better approximation of the probability density function in the region of interest but also a better approximation overall. The cost of this overall improvement is an increase in the number of Gaussian components. The principal benefit is not the modest increase in accuracy overall, but the significantly enhanced accuracy within the decision maker's region of interest.

Although the novel method in this paper is presented only in the pure forecast context, it is equally relevant in solving nonlinear filtering problems when measurements are available. The implementation of the method in stochastic filtering context is briefly discussed in the paper, and its performance evaluation on numerical examples is set as future work.

APPENDIX

LEMMA 1 *If* $\mathrm{Tr}[\mathbf{ABA}] = 0$ *and* $\mathbf{A}$ *is symmetric and positive definite then* $\mathrm{Tr}[\mathbf{B}] = 0$.

PROOF Let $\mathbf{A} = \mathbf{VSV}^T$ be a singular value decomposition of matrix $\mathbf{A}$, where $\mathbf{V}$ is a unitary matrix and $\mathbf{S}$ is a diagonal matrix. Our trace can now be written as $\mathrm{Tr}[\mathbf{ABA}] = \mathrm{Tr}[\mathbf{VSV}^T\mathbf{BVSV}^T] = \mathrm{Tr}[\mathbf{S}^2\mathbf{B}]$.

If $\mathrm{Tr}[\mathbf{S}^2\mathbf{B}] = 0$ then $\mathbf{S}^2\mathbf{B}$ is a commutator. Thus there is $\mathbf{X}$ and $\mathbf{Y}$ such that $\mathbf{S}^2\mathbf{B} = \mathbf{XY} - \mathbf{YX}$. But $\mathbf{B} = \mathbf{S}^{-2}\mathbf{XY} - \mathbf{S}^{-2}\mathbf{YX} = \mathbf{X}^*\mathbf{Y} - \mathbf{YX}^*$, where $\mathbf{X}^* = \mathbf{S}^{-2}\mathbf{X}$. Therefore $\mathbf{B}$ is also a commutator, hence $\mathrm{Tr}[\mathbf{B}] = 0$.

ACKNOWLEDGMENT

This work was supported under Contract No. HM1582-08-1-0012 from ONR.

REFERENCES

[1] R. Ababou and A. C. Bagtzoglou
Bigflow: A numerical code for simulating flow in variably saturated, heterogeneous geologic media. Theory and user's manual, version 1.1.
Technical report, Nuclear Regulatory Commission, Washington, D.C. (United States). Div. of Regulatory Applications, NUREG/CR-6028, 1993.

[2] D. Alspach and H. Sorenson
Nonlinear Bayesian estimation using Gaussian sum approximations.
IEEE Transactions on Automatic Control, **17**, 4 (1972), 439–448.

[3] A. C. Bagtzoglou and S. A. Baun
Near real-time atmospheric contamination source identification by an optimization-based inverse method.
Inverse Problems in Science and Engineering, **13** (2005), 241–259.

[4] R. N. Banavar and J. L. Speyer
Properties of risk-sensitive filters/estimators.
IEE Proceedings of Control Theory and Applications, **145** (1998), 106–112.

[5] S. Chakravorty, M. Kumar, and P. Singla
A quasi-Gaussian Kalman filter.
American Control Conference, 2006, June 2006, 6 pp.

[6] Y. Cheng and T. Singh
Source term estimation using convex optimization.
In *The 11th International Conference on Information Fusion*, Cologne, Germany, July 1–3, 2008.

[7] F. Daum and J. Huang
Curse of dimensionality and particle filters.
In *Proceedings of IEEE Aerospace Conference*, vol. 4, 2003, 8–15.

[8] A. Doucet, N. de Freitas, and N. Gordon
Sequential Monte-Carlo methods in practice.
Springer-Verlag, Apr. 2001.

[9] A. T. Fuller
Analysis of nonlinear stochastic systems by means of the Fokker-Planck equation.
International Journal of Control, **9**, 6 (1969).

[10] D. Giza and P. Singla
An approach for nonlinear uncertainty propagation: Application to orbital mechanics.
In *AIAA Guidance Navigation and Control Conference*, 2009.

[11] G. A. Grell, J. Dudhia, and D. R. Stauffer
A description of the fifth generation Penn State/NCAR Mesoscale Model (MM5).
Technical report, NCAR/TN-398+STR, NCAR Technical Note, Mesoscale and Microscale Meteorology Division, National Center for Atmospheric Research, Boulder, CO, 1995.

[12] R. N. Iyengar and P. K. Dash
Study of the random vibration of nonlinear systems by the Gaussian closure technique.
Journal of Applied Mechanics, **45** (1978), 393–399.

[13] S. J. Julier and J. K. Uhlmann
Unscented filtering and nonlinear estimation.
Proceedings of the IEEE, **92**, 3 (2004), 401–422.

[14] R. W. Katz and A. H. Murphy, (eds.)
Economic value of weather and climate forecasts.
Cambridge University Press, 1997.

[15] M. Kumar, P. Singla, S. Chakravorty, and J. L. Junkins
A multi-resolution approach for steady state uncertainty determination in nonlinear dynamical systems.
In *38th Southeastern Symposium on System Theory*, 2006.

[16] M. Kumar, P. Singla, S. Chakravorty, and J. L. Junkins
The partition of unity finite element approach to the stationary Fokker-Planck equation.
In *2006 AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Keystone, CO, Aug. 21–24, 2006.

[17] H. C. Lambert, F. E. Daum, and J. L. Weatherwax
A split-step solution of the Fokker-Planck equation for the conditional density.
Fortieth Asilomar Conference on Signals, Systems and Computers, 2006., Oct.–Nov. 2006, 2014–2018.

[18] T. Lefebvre, H. Bruyninckx, and J. De Schutter
Comment on a new method for the nonlinear transformations of means and covariances in filters and estimators.
IEEE Transactions on Automatic Control, **47**, 8 (2002).

[19] T. Lefebvre, H. Bruyninckx, and J. De Schutter
Kalman filters of non-linear systems: A comparison of performance.
International Journal of Control, **77**, 7 (2004), 639–653.

[20] G. Muscolino, G. Ricciardi, and M. Vasta
Stationary and non-stationary probability density function for non-linear oscillators.
International Journal of Non-Linear Mechanics, **32** (1997), 1051–1064.

[21] C. Musso, N. Oudjane, and F. Legland
Improving regularized particle filters.
In A. Doucet, N. de Freitas, and N. Gordon, (Eds.), *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, New York, 12 (2001), 247–271.

[22] S. T. Nielsen, S. Deme, and T. Mikkelsen
Description of the atmospheric dispersion module RIMPUFF.
Technical Report RODOS(WG2)-TN(98)-02, Riso National Laboratory, P.O. Box 49, DK-4000 Roskilde, Denmark, 1999.

[23] M. D. Paola and A. Sofi
Approximate solution of the Fokker-Planck-Kolmogorov equation.
Probabilistic Engineering Mechanics, **17** (2002), 369–384.

[24] G. Parmigiani and L. Inoue
Decision Theory: Principles and Approaches.
John Wiley & Sons, 2009.

[25] M. H. Pesaran and S. Skouras
Decision-based methods for forecast evaluation.
In M. P. Clements and D. F. Hendry, (Eds.), *A Companion to Economic Forecasting*, Blackwell Publishers, 11 (2002), 241–267.

[26] K. V. U. Reddy, Y. Cheng, T. Singh, and P. D. Scott
Data assimilation in variable dimension dispersion models using particle filters.
In *The 10th International Conference on Information Fusion*, Quebec City, Canada, 2007.

[27] H. Risken
The Fokker-Planck Equation: Methods of Solution and Applications.
Springer, 1989.

[28] J. B. Roberts and P. D. Spanos
Random Vibration and Statistical Linearization.
Wiley, 1990.

[29] S. Sadhu, S. Bhaumik, A. Doucet, and T. K. Ghoshal
Particle-method-based formulation of risk-sensitive filter.
Signal Processing, **89**, 3 (2009), 314–319.

[30] P. Singla and T. Singh
A Gaussian function network for uncertainty propagation through nonlinear dynamical system.
18th AAS/AIAA Spaceflight Mechanics Meeting, Galveston, TX, Jan. 27–31, 2008.

[31] I. H. Sloan and H. Woniakowski
When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?
Journal of Complexity, 1998, 1–33.

[32] A. N. Steinberg and C. L. Bowman
Rethinking the JDL data fusion levels.
In *Proceedings of the National Symposium on Sensor and Data Fusion*, John Hopkins Applied Physics Laboratory, 2004.

[33] I. R. Sykes, S. F. Parker, D. S. Henn, and B. Chowdhury
SCIPUFF Version 2.2.
Technical report, L-3 Titan Corporation, NJ, Jan. 2006.

[34] G. Terejanu, T. Singh, and P. D. Scott
Unscented Kalman filter/smoother for a CBRN puff-based dispersion model.
In *The 10th International Conference on Information Fusion*, Quebec City, Canada, 2007.

[35] G. Terejanu, P. Singla, T. Singh, and P. D. Scott
A novel Gaussian sum filter method for accurate solution to nonlinear filtering problem.
In *The 11th International Conference on Information Fusion*, Cologne, Germany, 2008.

[36] G. Terejanu, P. Singla, T. Singh, and P. D. Scott
Uncertainty propagation for nonlinear dynamical systems using Gaussian mixture models.
Journal of Guidance, Control, and Dynamics, **31** (2008), 1623–1633.

[37] G. Terejanu, P. Singla, T. Singh, and P. D. Scott
Decision based uncertainty propagation using adaptive Gaussian mixtures.
In *The 12th International Conference on Information Fusion*, Seattle, WA, July 2009.

[38] G. Terejanu, P. Singla, T. Singh, and P. D. Scott
Adaptive Gaussian sum filter for nonlinear Bayesian estimation.
Under review, submitted to *IEEE Transactions on Automatic Control*, Sept. 2009.

[39] S. Thrun, J. Langford, and V. Verma
Risk sensitive particle filters.
In *Advances in Neural Information Processing Systems 14*, MIT Press, 2002.

[40] J. von Neumann and O. Morgenstern
Theory of Games and Economic Behavior.
Princeton University Press, 1944.

Gabriel Terejanu received his B.E. degree in automation, specialization robots, from University of Craiova, Romania in 2004 and the M.S. and Ph.D. degrees in computer science and engineering from University at Buffalo, in 2007 and 2010, respectively.

He is currently a postdoctoral fellow in the Center of Predictive Engineering and Computational Sciences (PECOS) at the Institute for Computational Engineering and Sciences (ICES), University of Texas at Austin. His research interests are in information fusion, model validation, uncertainty quantification, and decision making under epistemic and aleatory uncertainty.

Puneet Singla received the B.S. degree from the Indian Institute of Technology, Kanpur, India, in 2000 and the M.S. and Ph.D. degrees from Texas A&M University, in 2002 and 2006, respectively, all in aerospace engineering.

He is currently an assistant professor of mechanical and aerospace engineering with the University at Buffalo. His work in attitude estimation included algorithms supporting a successful experiment StarNav that won the STS-107. He has authored more than 60 papers to date, including ten journal papers covering a wide array of problems, including attitude estimation, dynamics and control, approximation theory, and uncertainty propagation. He is the principal author of a new textbook *Multi-Resolution Methods for Modeling and Control of Dynamical Systems* (Boca Raton, FL: CRC Press, 2008).

Tarunraj Singh received his B.E., M.E., and Ph.D. degrees in mechanical engineering from Bangalore University, Indian Institute of Science, and the University of Waterloo, respectively.

He was a postdoctoral fellow in the Aerospace Engineering Department of Texas A&M University prior to starting his tenure at the University at Buffalo in 1993, where he is currently a professor in the Department of Mechanical and Aerospace Engineering. He was a von Humboldt Fellow and spent his sabbatical at the Technische Universitat Darmstadt in Germany and at the IBM Almaden Research center in 2000–2001. He was a NASA Summer Faculty Fellow at the Goddard Space Flight Center in 2003. His research is supported by the National Science Foundation, AFOSR, NSA, Office of Naval Research and various industries including MOOG Inc., Praxair and Delphi Thermal Systems. His research interests are in robust vibration control, optimal control, nonlinear estimation and intelligent transportation. Dr. Singh has published over 100 refereed journal and conference papers and has presented over 30 invited seminars at various universities and research laboratories.

Peter D. Scott received the B.S., M.S., and Ph.D. degrees from the Phillips School of Electrical Engineering, Cornell University.

He has served as assistant and associate professor in the University at Buffalo with primary appointment in the Electrical and Computer Engineering Department, subsequently in the Computer Science and Engineering Department. He has held adjunct faculty positions in the Department of Biophysical Sciences and currently with the School of Medicine and Department of Electrical Engineering in the University at Buffalo. He held a sabbatical faculty position with the Department of Electrical Engineering at the University of Connecticut and was Director of the Surgical Computing Laboratory, The Buffalo General Hospital. He has published over a hundred books, articles and peer reviewed conference publications. His research has been supported by NSF, NIH Heart and Lung Institute, New York Science and Technology Institute, the Wendt Foundation, and various agencies of the DOD. His research interests include optimal control and estimation, bioelectric phenomena, neural networks, data fusion and management of uncertainty.

Feature-Aided Tracking of Ground Vehicles using Passive Acoustic Sensor Arrays

VISHAL CHOLAPADI RAVINDRA
YAAKOV BAR-SHALOM
THYAGARAJU DAMARLA

Tracking of a moving ground target using acoustic signals obtained from a passive sensor network is a difficult problem as the signals are contaminated by wind noise and are hampered by road conditions, terrain and multipath, etc., and are not deterministic. Multiple target tracking becomes even more challenging, especially when some of the vehicles are light (e.g., wheeled) and some are heavy (e.g., heavy wheeled vehicles like trucks, tracked vehicles like tanks, etc.). In such cases the stronger acoustic signals from the heavy vehicles can mask those from the light vehicles, leading to poor detection of such targets. The full position estimates of emitters (targets), obtained following the association of the DoA angle estimates from multiple sensor arrays at each time scan, are used for target tracking. However, because of the particular challenges encountered in multiple ground vehicle scenarios, this association using kinematic (DoA angle) measurements only is not always reliable and can lead to lost as well as false tracks.

In this paper we propose a new feature-augmented static association algorithm where feature augmented DoA angle measurements from multiple sensors are associated to localize targets and obtain composite measurements (position estimates) using a static multidimensional assignment (MDA) framework. We present a novel DoA detection scheme followed by a feature extraction technique designed from and for real data. Dynamic S-D and feature-aided S-D (multidimensional) assignment algorithms are presented to assign composite measurements and feature-augmented composite measurements, respectively, to tracks. The techniques are developed based on real data sets and tested on real data based on a field experiment.

Manuscript received August 06, 2009; revised February 03, 2010; released for publication August 17, 2010.

Refereeing of this contribution was handled by Dr. Huimin Chen and Dr. Stefano Coraluppi.

Authors' addresses: V. C. Ravindra and Y. Bar-Shalom, Department of Electrical & Computer Engineering, University of Connecticut, Storrs, CT 06269, E-mail: (vishalcr@nal.res.in, ybs@engr.uconn.edu); T. Damarla, Army Research Laboratories, Adelphi, MD 20783, E-mail: (rdamarla@arl.army.mil).

1. INTRODUCTION

Ground vehicle tracking using acoustic data obtained from passive sensor networks is a very challenging problem as the signals are contaminated by wind noise, and are hampered by road conditions, terrain, multipath, etc., and are not deterministic. Multiple vehicle tracking becomes even more challenging, especially when some of the vehicles are light vehicles, some are heavy wheeled vehicles and some are tracked like tanks, and are closely spaced. Passive acoustic sensors are gaining in popularity because of their low cost, ease of deployment, and the fact that they can be deployed on the ground. As passive sensors do not emit their own signals, unlike active sensors, there is no danger of being detected. Passive acoustic sensor networks are being used in battlefield monitoring as well as in civilian surveillance applications.

In single target scenarios with active sensors, kinematic measurements (such as range, bearing, etc.) are obtained which can be used to estimate the trajectory of targets [2]. When a network of passive sensors is used, however, the range of a target can be obtained only after associating the direction of arrival (DoA, or line of sight—LoS) angle estimates obtained by at least three sensors. This is because ghosting or false intersections occur with just two sensors with multiple targets in the same plane. Hence, in order to eliminate or reduce ghosting one has to use DoAs from at least three sensors (see [1, Sec. 8.8.2]). This makes the problem computationally expensive for a large number of measurements. There is added difficulty in data association when targets stay close together over an extended period of time (as acoustic signals from some targets can fade and then re-appear or can be masked by stronger signals from other targets), because one has to associate the DoA angle estimates from the *same target* to obtain its position.

Feature-aided tracking (FAT) is a rapidly developing research area [15, 16, 19, 33], as various FAT techniques exploit certain properties of the received signals at the sensor level to augment the kinematic measurements to alleviate the difficulties encountered in data association and tracking using kinematic measurements only. For scenarios such as the one considered in the present paper, traditional data association and target tracking algorithms can be enhanced by integrating features with the kinematic data. Typical features are estimated target dimensions, radar cross-section, signature data and received signal properties among others. Features have been typically used to enhance target identification, i.e., classification, discrimination or recognition [7] and to enhance data or track association [1, 15].

Classification aided tracking is presented in [15, 16] where it is offered as an alternative to the traditional approach (with the classifier being treated separately from the tracker). However, a unique mathematical model is needed to be constructed for each target type in order

to make such an algorithm robust. In [8] a multilevel feature-based association algorithm to simultaneously track and identify targets is presented, where features extracted from high range resolution (HRR) profiles are correlated with target signatures for identification. The simultaneous use of target classification information and target kinematic measurements for target tracking using a multidimensional assignment framework is presented in [4]. Another paper [28] studies the combining of identification and tracking, as each task can be enhanced by fusing it with the other; however, these methods require the availability of target signatures beforehand. In [22] kinematic measurements from ground moving target indicator (GMTI) and local motion features from HRR are combined in a probabilistic logic based tracker for dense multitarget scenarios. Features extracted from HRR profiles, by approximating it with a Gaussian mixture density using the expectation maximization (EM, see [17]) algorithm, are combined with kinematic measurements in a probabilistic framework for multiple target tracking in [33]. Another algorithm is presented in [21] to combine GMTI measurements with features extracted in the wavelet domain from HRR profiles for multitarget tracking in a joint probabilistic data association (JPDA, see [1, 3]) context. Various feature-aided classification and FAT techniques for single target as well as multiple targets have been proposed and implemented in the recent past for ground vehicle scenarios using acoustic signals from sensor arrays. Classification and identification of multiple targets using acoustic signatures is presented in [12]. In this approach, bearing tracking and data association is first performed. Once the bearings of the targets are established, beamforming is performed in each individual direction to extract harmonic features, and a multi-variate Gaussian classifier is applied on each feature set for identification and classification. Many target classification algorithms based on acoustic signatures in the literature [20, 24, 25, 34] assume that a single target is present and use statistical parameters, namely the mean and variance of several harmonics of the fundamental engine firing rate of each target for identification and classification. When multiple targets are present within the surveillance region of acoustic sensor arrays, the measurements no longer exhibit the same statistics as they did for the individual targets when alone, due to masking and interference. Hence, the algorithms developed with a single target assumption perform poorly [12]. In [11] an algorithm to track multiple ground vehicles was presented based on a template of the DoA angles for the leading target or the target closest to the sensor array and hence the loudest with the strongest signal to noise ratio (SNR). This template was used to predict the DoA angles of all the other targets. A distributed fusion algorithm is developed in [38] that extracts features in time as well as frequency domain from the acoustic signals and integrates classification results from different sensor arrays to increase the classification accuracy. Acoustic signatures from air and ground vehicles produced from their engine or propulsion mechanisms are used by neural network based pattern recognition algorithms for classification in [35]. Target tracking is easier if the identities of signal sources (targets) are known beforehand based on classification techniques [12, 13]. However, this is an unreasonable assumption in the scenario considered in the present paper due to the various challenges discussed earlier. Moreover, the extraction of reliable signatures is challenging because of the environment, interference from other moving parts and nearby vehicles and their nonstationarity. Further, a majority of the classification techniques require building reliable mathematical models and templates as well as an extensive library of training data for each target type.

In the present paper we present a feature-aided data association technique that employs a *feature-augmented* measurement set (as compared to just kinematic measurements) in a static as well as dynamic assignment framework for target localization and tracking. The major challenge is the development of reliable models to extract and characterize features. Section 2 describes the generation of the PSD of the signals received by the passive sensors using the minimum variance distortionless response (MVDR, see [36]) spectral estimation technique, and a novel DoA angle detection scheme using the PSD. Section 3 describes a novel algorithm used to extract features from the PSD. Section 4 introduces and describes the target localization problem, i.e., obtaining their full position estimates (composite measurements) using a multidimensional assignment (MDA[1]) framework. The composite measurements are assigned to tracks using both conventional as well as feature-aided dynamic MDA algorithms in Section 5. This multisensor information processing is configuration III from [1, Sec. 8.2]. Section 6 describes the real data scenario considered in the present paper and Section 7 provides the target localization and tracking performance comparison results between the conventional and feature-aided MDA algorithms. The techniques are developed based on real data sets and tested on real data based on a field experiment.

2. PSD GENERATION AND DOA DETECTION

Circular 2-D passive sensor arrays made up of M microphones arranged equidistantly, as shown in Fig. 1, are employed in a passive sensor network on the ground to listen to multiple ground vehicles. Assuming that the vehicles are at a sufficiently large distance from the sensor array, so that the received signals from the targets can be approximated by a planar wavefront, various wideband beamforming algorithms described in

[1]MDA is also known as S-D assignment. Note the use of S for the dimension of the assignment (the number of lists).

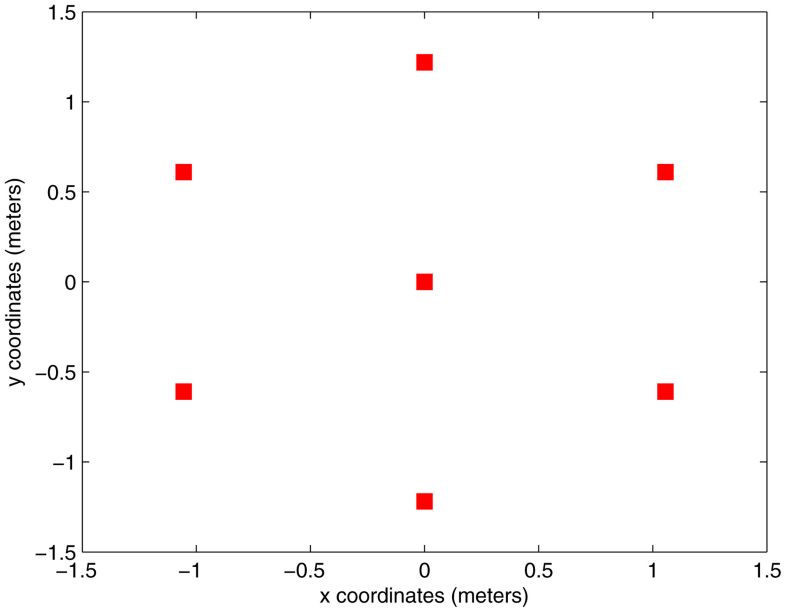


Fig. 1. Circular sensor array made up of 7 microphones.

[9, 10, 13, 23, 26, 27, 36, 37, 39, 41] can be used to detect signal sources and estimate their DoA angles with respect to the known sensor array positions.

At each sampling time, a fast Fourier transform (FFT) is performed on the raw acoustic data from each microphone in an array and a discrete frequency band from f_1 to f_{n_b}, where n_b is the number of bins with bin intervals of 1 Hz, is chosen for processing. The frequency bins are chosen to conform to the typical frequencies of the acoustic signals emitted by engines and other moving vehicle parts. As a result, we have a wideband processing algorithm that uses data from n_b frequency bins. We denote the FFT data used for processing at each sample by $X(m, f_b)$, where $m \in \{1, \ldots, M\}$ represents each microphone and $f_b \in \{f_1, \ldots, f_{n_b}\}$ indicates the frequency bins. The minimum variance distortionless response (MVDR) algorithm provides, for each sensor array, an estimate of the power spectral density (PSD) of the acoustic signals impinging the array at a particular scan

$$\hat{P}(\theta_d, f_b) = \frac{1}{\underline{a}(f_b)' \hat{R}(f_b)^{-1} \underline{a}(f_b)},$$

$$\theta_d \in \{\theta_1, \ldots, \theta_D\} \quad \text{and} \quad f_b \in \{f_1, \ldots, f_{n_b}\} \tag{1}$$

and

$$\underline{a}(f_b) = \{X\}'_{f_b} V \tag{2}$$

with $\{X\}_{f_b}$ denoting the column of the matrix X corresponding to column f_b, V denoting the steering matrix (of dimension $M \times 360$, representing all possible directions in degrees, see Ch. 2, [36]), and

$$\hat{R}(f_b) = \frac{1}{M} \sum_{m=1}^{M} X(m, f_b)^* X(m, f_b) \tag{3}$$

the estimated covariance matrix.

The DoA angles are estimated in the present paper by a novel DoA angle detection scheme from the PSD (illustrated in Fig. 2). DoA detection, at each angle θ, can be performed by applying a thresholding algorithm on the frequency averaged estimated power spectrum corresponding to direction θ

$$\hat{P}(\theta) = \frac{1}{n_b} \sum_{b=1}^{n_b} \hat{P}(\theta, f_b). \tag{4}$$

Fig. 2 shows a snapshot of the estimated PSD at a particular scan for a sensor array. Along each angle θ the frequency averaged PSD denoted by $\hat{P}(\theta)$ in (4) is illustrated in Fig. 3(a). In [13], thresholding on $\hat{P}(\theta)$ was used to detect DoA angles, i.e., a DoA angle detection was declared at θ_d if a peak was observed in the spectrum $\hat{P}(\theta)$ at θ_d. The first derivative $d\hat{P}(\theta)/d\theta$ is shown in Fig. 3(b). A DoA detection is made at angle θ_d if a positive peak is detected at θ_d in $-d^2\hat{P}(\theta)/d\theta^2$ shown in Fig. 3(c). This is because a thresholding algorithm based just on the frequency averaged power spectrum is likely to miss peaks which can be detected by a thresholding algorithm that is based on the second derivative of the spectrum. In the example shown in the figure, the DoA angles detected are 239°, 123°, 219° and 189°, arranged in decreasing order of their corresponding amplitudes.

In a typical ground vehicle tracking scenario where the targets are moving on road or off-road conditions, there are a variety of extraneous factors which affect the acoustic signal, such as road conditions, sound generated by moving parts, wind, etc. This results in numerous false alarms, i.e., power detections in frequency-angle bins not directly due to the engine or moving parts of the targets. There are also missing DoA angle estimates (missed detections) due to signal attenuation or the possible masking of signals from lighter vehicles

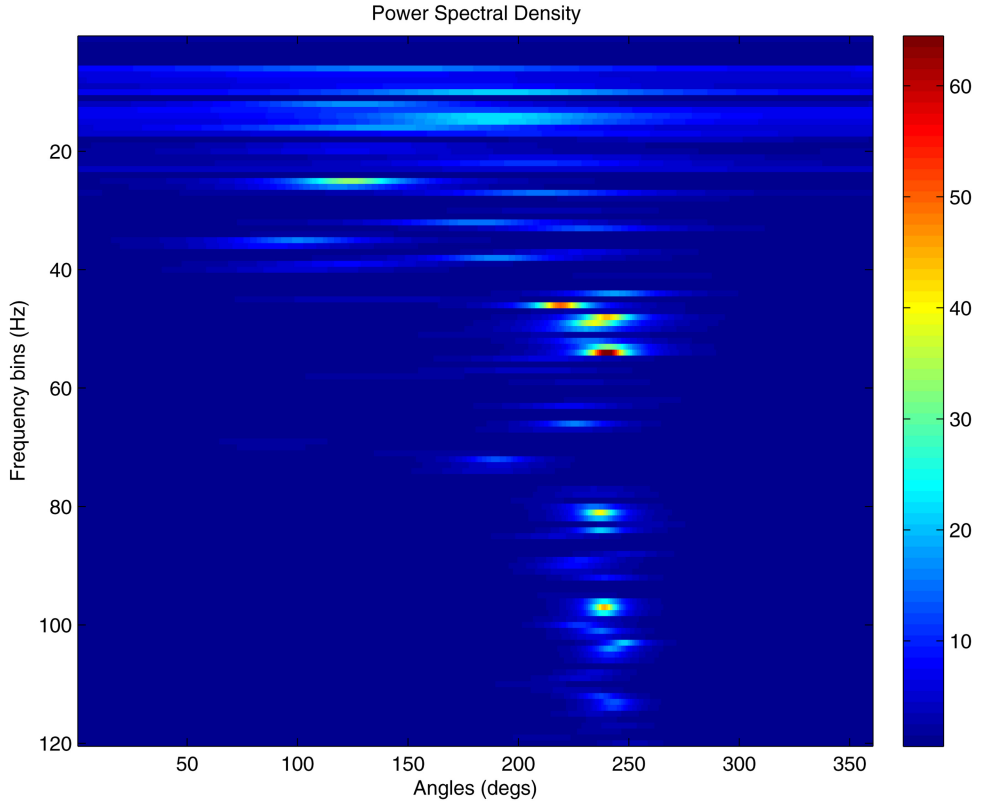


Fig. 2. Power spectral density as a function of DoA angles and frequency bins obtained from the MVDR algorithm for one sensor array at a particular scan (data set [12]).

by those from the heavier vehicles, especially when the targets are closely spaced. As a result, the quality of the data association is low if just the DoA angle estimates are used as measurements. Therefore, the power spectrum will be exploited to generate features which could enhance the accuracy of data association. Motivated by this, a new feature extraction procedure is described in the next section.

3. FEATURE EXTRACTION

In feature-aided ground vehicle tracking applications the peak amplitudes of the spectrum have been used as features [12, 35] in the past; however, due to signal attenuation and masking in the case of multiple vehicle road convoys, they are not reliable as features. The location and the spread of the peaks carry more useful information about the signal source, compared to the amplitude, because they are not as affected by signal attenuation and because a vehicle causes frequency peaks in similar locations of the spectrum across all the sensor arrays at any given time.[2] In this paper we propose a statistical modeling of the power distribution in the frequency bins along each detected DoA and use a Gaussian mixture model (GMM) to extract feature vectors instead of just scalar features. For the problem considered only scalar features were used in [12, 13].

[2]Doppler has been neglected as it amounts to less than 1–3(assuming 5–10 m/s vehicle speed).

3.1. Fitting of a Gaussian Mixture Model (GMM)

The observed data $d(x)$, i.e., the estimated power spectrum in a particular direction x, is modeled as

$$d(x) = y(x;\beta) + \epsilon(x) \tag{5}$$

where $y(x;\beta)$ is the fitted parametric model, β is the parameter vector and $\epsilon(x)$ is the fitting error. The objective is to estimate the parameters of the model such that the error (noise) is minimized in a statistical sense.

The nonlinear least squares (NLLS) method is used to estimate the parameters of a nonlinear model—a GMM—used to fit the data. The Gaussian mixture modeling, useful for peak finding applications, is given by

$$y(x;\beta) = \sum_{l=1}^{n} \alpha^l \exp\left[-\frac{1}{2}\left(\frac{x-\mu^l}{\sigma^l}\right)^2\right] \tag{6}$$

where $\beta = [(\alpha^1,\mu^1,\sigma^1),\ldots,(\alpha^n,\mu^n,\sigma^n)]'$ is the parameter vector of the GMM such that α^l is the weight (amplitude), μ^l is the location and σ^l is the width of the peak of component l, and n is the number of components of the GMM. The NLLS method can be used to estimate the parameters of the GMM which best fit the observed data in (5). Fig. 4 illustrates the GMM fitting results when applied to power spectrum data for certain DoA detections obtained by three sensor arrays at a particular time. A GMM with $n = 4$ components was used to ex-

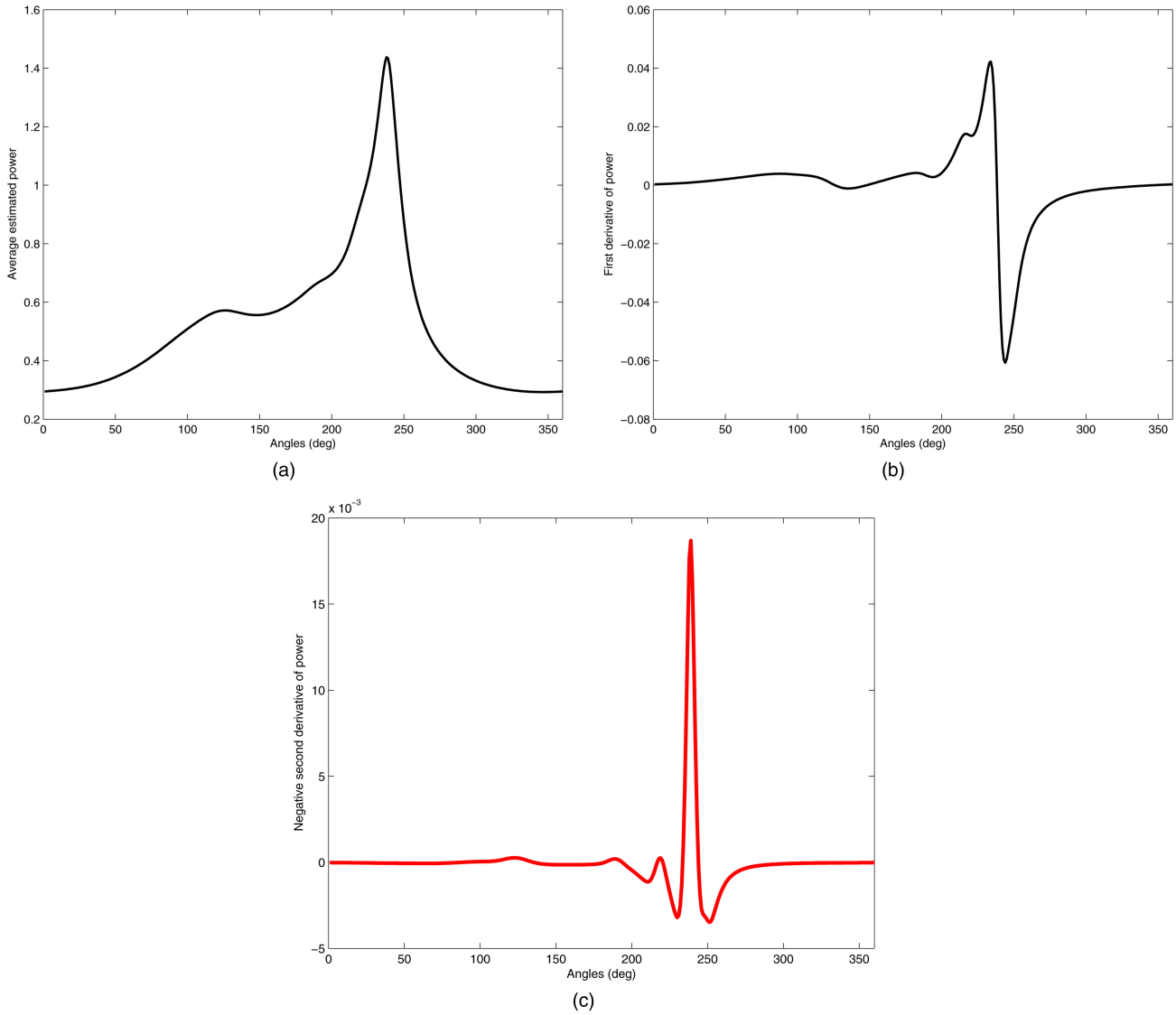


Fig. 3. The thresholding used to detect DoAs from a particular sensor array at a certain scan: (a) frequency averaged power spectrum as a function of angles only, (b) the first derivative of the function in (a), and (c) the negative of the second derivative of the function in (a).

tract the location, width and the amplitude of the peaks from the power spectrum data.[3]

From the example illustrated in Fig. 4, one can see that there are at most $n = 4$ frequency peaks from each of the $S = 3$ sensors, corresponding to three detected DoA angles θ_1, θ_2 and θ_3 (subscripted by the sensor index). These peak locations form a matrix of dimension $n \times S$ (with the column elements listed in decreasing amplitude order)

$$L = \begin{pmatrix} 40 & 78 & 40 \\ 15 & 40 & 18 \\ 36 & 117 & 37 \\ 49 & 15 & 14 \end{pmatrix}. \tag{7}$$

[3]The selection of n is taken as a design parameter, a larger n was found to lead to excessive uncertainty in the feature model for the application considered.

The peak location matrix given in (7) for the three sensor arrays in Fig. 4 shows that the peak locations are not necessarily matched across the S sensors, i.e., a peak location is not necessarily matched with the peak locations shown in the same row of the other $S - 1$ columns (lists). For example, in (7), the first row is [40 78 40]. The location of the peak in the second column (at 78 Hz) is not close to the location of the peaks in the first and third columns (at 40 Hz). Hence, in order to properly order peak locations across lists (sensors), each peak location should be matched to other peak locations (in the other lists) in such a way that each matched S-tuple of peak locations consists of peak locations which are close to each other. If a peak is located in such a way that it cannot be matched to any other peaks in the remaining lists, it is matched to a dummy element which indicates a missing peak detection. Each peak location is matched to at most one corresponding peak location from another list. This matching of peak locations is

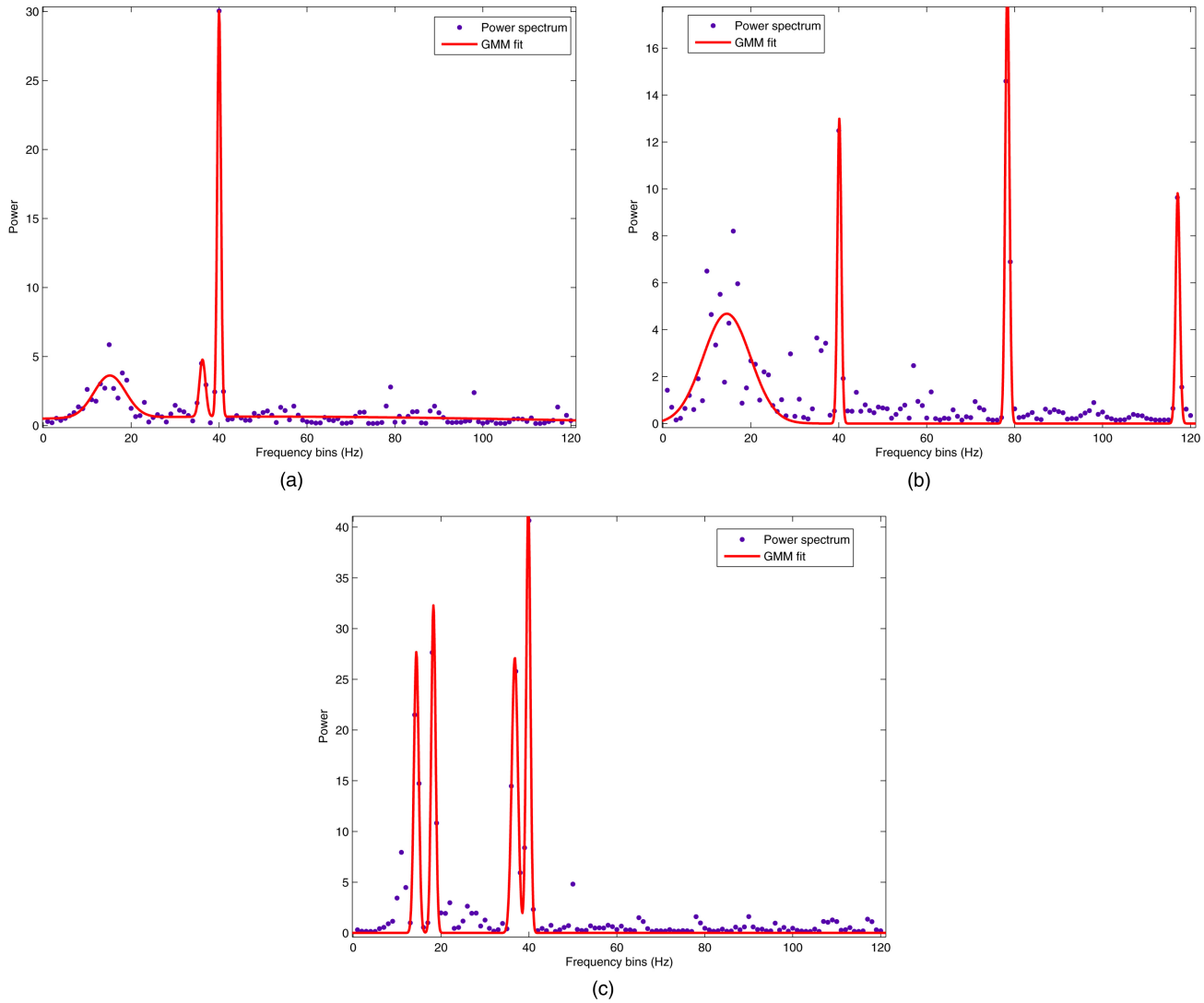


Fig. 4. Gaussian mixture model fitting to extract features: (a) feature data from sensor array 1 (DoA angle θ_1, peak locations: [40, 15, 36, 49]), (b) feature data from array 2 (DoA angle θ_2, peak locations: [78,40,117,15]), and (c) feature data from array 3 (DoA angle θ_3, peak locations: [40, 18, 37, 14]).

done by solving a generalized multidimensional assignment (MDA)[4] algorithm [1, 14, 29, 30, 31, 32] that is described in Sections 4 and 5.

3.2. Frequency Peak Location Matching to Obtain Feature Vectors

The peak location vectors (e.g., columns of the matrix in (7)), estimated by the GMM algorithm, corresponding to each of the detected DoA angles are $\Phi^p_{1i_1,\ldots,Si_S} = [\phi^p_{1i_1}, \phi^p_{2i_2}, \ldots, \phi^p_{Si_S}]$ where each peak location vector $\phi^p_{si_s}$ corresponding to DoA angle θ_{si_s}, $s = 1,\ldots,S$, is of the form

$$\phi^p_{si_s} = [\oslash \quad \mu^1_{si_s} \quad \mu^2_{si_s} \cdots \mu^n_{si_s}]' \tag{8}$$

where $\mu^l_{si_s}$ is the location of the lth component (peak) of the GMM (6) used to fit $\hat{P}(\theta_{si_s})$ and $\oslash$ indicates the dummy element[5] that signifies the missed detection of a peak.

The MDA algorithm is solved, for the purpose of matching of peak locations across lists, to obtain the following S-tuple of ***feature vectors*** corresponding to the S-tuple of detected DoA angle estimates $\theta_{1i_1}, \theta_{2i_2}, \ldots, \theta_{Si_S}$

$$\Phi_{1i_1,\ldots,Si_S} = [\phi_{1i_1}, \phi_{2i_2}, \ldots, \phi_{Si_S}] \tag{9}$$

where

$$\phi_{si_s} = [\oslash \quad \mu^{j_{s1}}_{si_s} \quad \mu^{j_{s2}}_{si_s} \cdots \mu^{j_{snm}}_{si_s}]' \tag{10}$$

where $\mu^{j_{sq}}_{si_s} \in \{\oslash, \mu^l_{si_s}\}$ and $q = 1,\ldots,n_m$. Each element $\mu^l_{si_s}$ from (8) appears exactly once in ϕ_{si_s} while $\oslash$ appears $n_m - n$ times. It has to be noted that n_m could vary with each S-tuple of DoA angle estimates being considered, while n remains the same as it is a GMM fitting design parameter and is fixed. This leads to "matched"

[4]The assignment problem is called generalized assignment if dummies are used.

[5]Indexed by zero, and shown as the top row.

feature vectors which is shown using the illustrative example in (7) and (11). The example presented corresponds to the same triplet ($S = 3$) of DoA angle estimates $\theta_1, \theta_2, \theta_3$, which gives rise to the peaks illustrated in Fig. 4. The triplet of ***peak location vectors*** of length $n = 4$ are as shown in (7). The triplet of feature vectors, each of length $n_m = 7$, obtained after performing the matching of peak locations as described above is

$$L^{\text{matched}} = \begin{pmatrix} 40 & 40 & 40 \\ 15 & 15 & 14 \\ 36 & \oslash & 37 \\ 49 & \oslash & \oslash \\ \oslash & 78 & \oslash \\ \oslash & 117 & \oslash \\ \oslash & \oslash & 18 \end{pmatrix}. \tag{11}$$

4. STATIC MDA PROBLEM: GENERATION OF FULL POSITION ESTIMATES

One of the most important issues in multisensor-multitarget tracking is data association [1]. Static data association, i.e., measurement to measurement association at each time scan is especially important in the case of passive sensor networks, where each sensor array obtains only DoA angle measurements at each scan. At each scan, the DoA angle measurements from the same target, from at least three sensor arrays (in order to reduce ghosting, see [1, Sec. 8.8.2]), have to be associated in order to obtain the full position estimate of its location. Recently, a class of algorithms called multidimensional assignment (MDA) algorithms have been developed to solve the data association problem using an assignment approach [6, 14, 29, 30, 31, 32]. The present paper uses the MDA approach to solving the data association problem. This approach, designated as Multisensor Information Configuration III in [1], requires first a **static association** of DoA angle measurements to DoA angle measurements across sensor arrays at each time k, resulting in full position estimates called "composite measurements" (CM). If the DoA angle measurements are augmented by features, the result will be feature-augmented composite measurements. This is to be followed by **dynamic association** of the CMs to tracks and filtering which then yields target tracks.

4.1. Conventional-Cost Based MDA

For the static problem considered, at any given time k, we are given S scans (lists) of measurements from S passive sensors gathering data in a surveillance region. Each list contains a certain number of detections, not necessarily equal to the number of targets. The objective is to obtain the full position estimates of an unknown number of targets using the S lists of DoA angle measurements. The sensors obtain measurements at discrete time samples,[6] $k = 1, \ldots, K$, with a time period of T s.

Each list consists of DoA angle measurements θ_{si_s}, where $i_s = 1, 2, \ldots, n_s$. Each measurement either originated from a true target t or from some spurious source of clutter ($t = 0$). If the measurement θ_{si_s} is originated from target t, it is modeled as

$$z_{si_s} = h(\mathbf{x}_t, \mathbf{x}_s) + w_{si_s} \tag{12}$$

where $\mathbf{x}_t$ is the true target position, $\mathbf{x}_s$ is the (fixed) sensor array position, h is the measurement function, $w_{si_s} \sim \mathcal{N}(0, \sigma^2_{si_s})$ is the measurement noise and σ_{si_s} represents the sensor error. If the measurement θ_{si_s} is originated from clutter, it is modeled as uniformly distributed within the field of view of the sensor array s, i.e.,

$$p(\theta_{si_s} \mid t = 0) = \frac{1}{V_s} \tag{13}$$

where V_s is the volume of the field of view of the sensor s. In addition, the probability of detection of a target is P_D.

The goal is to localize the targets by estimating their positions at time k. A generalized likelihood ratio[7] which uses estimated target positions instead of true target positions (which are unavailable) for candidate associations, is used to attach costs to each feasible S-tuple of measurements (or candidate associations) [1]. To account for missed detections that give rise to incomplete S-tuples a *dummy* measurement is added to each list which simplifies the notation for these incomplete measurement-to-target associations caused by missed detections. The MDA algorithm is then used to *globally minimize* the cost in order to obtain feasible associations.

The likelihood that an S-tuple of measurements Z_r (where r stands for the S-tuple index), originated from target t, with position $\mathbf{x}_t$ at some instant k is

$$\Lambda(t) = p(Z_r \mid t) = \prod_{s=1}^{S} [1 - P_{D_s}]^{1-\delta_{i_s}} [P_{D_s} p(\theta_{si_s} \mid \mathbf{x}_t)]^{\delta_{i_s}}, \quad i_s \in \{0, 1, \ldots, n_s\} \tag{14}$$

where δ_{i_s} is the measurement detection indicator function. The likelihood that the measurements Z_r are all spurious or *unrelated* to target t, i.e., ($t = 0$) is

$$\Lambda(t = 0) = p(Z_r \mid t = 0) = \prod_{s=1}^{S} \left[\frac{1}{V_s}\right]^{\delta_{i_s}}. \tag{15}$$

The cost of associating the S-tuple Z_r to target t is given by the negative log-likelihood ratio (NLLR), where the likelihoods that make up the numerator and the denominator are given in (14) and (15), respectively. However, since $\mathbf{x}_t$ in (14) is unknown, it will be replaced

[6] All the sensors are assumed synchronized with respect to time.

[7] An extensive treatment on the choice of a likelihood ratio as a cost function can be found in [5].

by its maximum likelihood estimate (MLE)

$$\hat{\mathbf{x}}_t = \arg\max_{\mathbf{x}_t} p(Z_r \mid t) \tag{16}$$

and the cost is

$$c_r = -\ln \frac{\hat{\Lambda}(t)}{\Lambda(t=0)} \tag{17}$$

where

$$\hat{\Lambda}(t) = \prod_{s=1}^{S} [1 - P_{D_s}]^{1-\delta_{i_s}} [P_{D_s} p(\theta_{si_s} \mid \hat{\mathbf{x}}_t)]^{\delta_{i_s}}. \tag{18}$$

Using (12), (15) and (18) in (17), the cost of the candidate association of the S-tuple of measurements Z_r to a target t is

$$c_r = -\sum_{s=1}^{S} [1 - \delta_{i_s}] \ln(1 - P_{D_s}) + \delta_{i_s} \ln[P_{D_s} V_s p(\theta_{si_s} \mid \hat{\mathbf{x}}_t)]. \tag{19}$$

4.2. Feature-Aided MDA

In the feature-aided MDA problem, at time k, we are given S lists of feature-augmented measurement vectors instead of S lists consisting only of DoA angle measurements. An S-tuple of DoA angle measurements Z_r can be augmented with an S-tuple of feature vectors $\Phi_r = \{\phi_{1i_1}, \ldots, \phi_{Si_S}\}$, where ϕ_{si_s} denotes the feature vector corresponding to a DoA angle measurement θ_{si_s}, to form an S-tuple of feature-augmented measurement vectors (denoted by boldface)

$$\mathbf{Z}_r = [Z_r, \Phi_r]'. \tag{20}$$

The generalized likelihood that the feature-augmented S-tuple $\mathbf{Z}_r$ consists of feature-augmented measurement vectors from a target t is[8]

$$\hat{\mathbf{\Lambda}}(t) = p(\mathbf{Z}_r \mid t) = p([Z_r, \Phi_r] \mid (\hat{\mathbf{x}}_t, \hat{\varphi}_t)) \tag{21}$$

where $\hat{\mathbf{x}}_t$ is the MLE of the target position given in (16) and $\hat{\varphi}_t$ is the estimated feature vector of target t. Assuming that the DoA angle measurement errors and the feature vectors are independent, we have

$$\hat{\mathbf{\Lambda}}(t) = p(Z_r \mid \hat{\mathbf{x}}_t) p(\Phi_r \mid \hat{\varphi}_t) = \hat{\Lambda}_K(t) \hat{\Lambda}_\phi(t) \tag{22}$$

where $\hat{\Lambda}_K(t)$ and $\hat{\Lambda}_\phi(t)$ represent the kinematic and feature generalized likelihood functions, respectively. Note that $\hat{\Lambda}_K(t)$ is the same as the generalized likelihood function given in (18).

The generalized likelihood that an S-tuple of feature-vector measurements, Φ_r, corresponding to an S-tuple of angle measurements Z_r are from a target t with feature estimate $\hat{\varphi}_t$ is given by

$$\hat{\Lambda}_\phi(t) = p(\phi_{1i_1}, \phi_{2i_2}, \ldots, \phi_{Si_S} \mid \hat{\varphi}_t) \tag{23}$$

where ϕ_{si_s} is the feature vector corresponding to the DoA angle measurement θ_{si_s} from the sth list

$$\phi_{si_s} = [\mu^1_{si_s}, \mu^2_{si_s}, \ldots, \mu^{n_r}_{si_s}]' \tag{24}$$

where $\mu^l_{si_s}$ ($l = 1, \ldots, n_r$) represents either a feature (detected matched peak location) or a dummy (missed detection of the peak in list s), as described in Section 3.2.[9]

Assuming independence between the measurement errors across lists, (23) can be simplified

$$\hat{\Lambda}_\phi(t) = \prod_{s=1}^{S} \{p(\phi_{si_s} \mid \hat{\varphi}_t)\}^{\delta_{i_s}}. \tag{25}$$

Substituting (24) in (25), and assuming that the components ϕ_{si_s} of the feature vector are uncorrelated, we have

$$\hat{\Lambda}_\phi(t) = \prod_{s=1}^{S} \left\{ \prod_{l=1}^{n_r} [1 - P_{D^l_s}]^{1-\delta_{s_l}} [P_{D^l_s} p(\mu^l_{si_s} \mid \hat{\varphi}_t)]^{\delta_{s_l}} \right\}^{\delta_{i_s}} \tag{26}$$

where $P_{D^l_s}$ is the (nonunity) probability of detection of the features in list s, while δ_{s_l} is the detection indicator function for the feature $\mu^l_{si_s}$. The feature $\mu^l_{si_s}$ is assumed to be distributed as follows

$$p(\mu^l_{si_s} \mid \hat{\varphi}_t) = \mathcal{N}(\mu^l_{si_s}; \hat{\mu}^l(t), (\sigma^l_{si_s})^2), \qquad i_s \in \{1, \ldots, n_s\} \tag{27}$$

where

$$\hat{\mu}^l(t) = \frac{\sum_{s=1}^{S} \mu^l_{si_s} \delta_{s_l}}{\sum_{s=1}^{S} \delta_{s_l}}. \tag{28}$$

The standard deviation $\sigma^l_{si_s}$ of the feature $\mu^l_{si_s}$ is obtained from the GMM fitting in (6).

The likelihood that the S-tuple $\mathbf{Z}_r$ consists of feature-augmented measurement vectors which are all spurious or are unrelated to a target t is

$$\begin{aligned} \mathbf{\Lambda}(t=0) = p(\mathbf{Z}_r \mid t=0) &= \prod_{s=1}^{S} \{p(\theta_{si_s} \mid t=0) p(\phi_{si_s} \mid \varphi(t)=0)\}^{\delta_{i_s}} \\ &= \prod_{s=1}^{S} \left\{ p(\theta_{si_s} \mid t=0) \cdot \left[\prod_{l=1}^{n_r} p(\mu^l_{si_s} \mid \varphi(t)=0) \right]^{\delta_{s_l}} \right\}^{\delta_{i_s}}. \end{aligned} \tag{29}$$

Assuming that the angle and feature clutter measurements are uniformly distributed, we have

$$\mathbf{\Lambda}(t=0) = \prod_{s=1}^{S} \left\{ \frac{1}{V_s} \left[\prod_{l=1}^{n_r} \frac{1}{V^f_s} \right]^{\delta^l_s} \right\}^{\delta_{i_s}} \tag{30}$$

where V_s and V^f_s are the surveillance volumes in angle and frequency, respectively, of array s.

[8] The likelihood based on the augmented (kinematic and feature) measurements is denoted by boldface ($\mathbf{\Lambda}$).

[9] The length of the feature vector n_r varies for each candidate S-tuple of feature-augmented measurement vectors, as explained in Section 3.2. Additional notation is omitted for simplicity.

The cost of assigning the S-tuple $\mathbf{Z}_r$ is given by the NLLR obtained from the likelihood functions (22) and (30), and substitution from (18) and (26)

$$\mathbf{c}_r = -\ln \frac{\hat{\mathbf{\Lambda}}(t)}{\mathbf{\Lambda}(t=0)} = -\ln \frac{\hat{\Lambda}_K(t)\hat{\Lambda}_\phi(t)}{\mathbf{\Lambda}(t=0)}. \tag{31}$$

This cost function can be simplified to the following form

$$\begin{aligned}\mathbf{c}_r = &-\sum_{s=1}^{S}[1-\delta_{i_s}]\ln(1-P_{D_s}) + \delta_{i_s}\ln[P_{D_s}V_s p(\theta_{si_s} \mid \hat{\mathbf{x}}_t)] \\ &+ \delta_{i_s}\left\{\sum_{l=1}^{n_r}(1-\delta_s^l)\ln(1-P_{D_s^l}) + \delta_s^l \ln[P_{D_s^l}V_s^f\, p(\mu_{si_s}^l \mid \hat{\varphi}(t))]\right\}.\end{aligned} \tag{32}$$

The most likely set of S-tuples such that each feature-augmented measurement vector in a list is assigned to either other measurement vectors, or declared false, with the constraint that each assigned S-tuple receives at most only one measurement vector from each list, is obtained by solving a global MDA optimization problem using the cost function (32). The assignment problem formulation can be found in [1, 14, 29, 30, 32]. Various Lagrangian relaxation techniques have been developed [14, 30, 31] to solve the MDA problem and the details are discussed in the following section. The solution to the MDA problem at each scan k results in a set of assigned feature-augmented measurement vectors (with at least 3 non-dummy DoA angles), and using triangulation (see Ch. 8, [1]) on the assigned S-tuples of DoA angle measurements the full position estimates (the composite measurements) are obtained.

5. TARGET TRACKING WITH DYNAMIC ASSIGNMENT

The static multidimensional assignment algorithm described in Section 4 yields composite measurements (CM) at each scan. The classical dynamic assignment problem [1, 29], i.e., the assignment of CMs from the current scan k or a window of scans up to scan $k-S+1$ to established tracks from the previous scan $k-1$, can be solved using an S-D[10] dynamic assignment framework. When the list of CMs at the current scan only is considered, the 2-D[11] dynamic assignment framework is used. Solving a 2-D problem is computationally less demanding than the case where $S>2$, as it can be solved optimally in polynomial time using a generalized auction algorithm [6, 18]. In a 2-D dynamic assignment framework, each CM at scan k is to be assigned to either an established track $T_u(k-1)$ from scan $k-1$, where $u \in \{1,\ldots,U(k-1)\}$, or is used to start a new track if assigned to $u=0$ (a dummy), based on the global solution of a 2-D generalized assignment algorithm using a likelihood ratio-based cost function. However, the information about track evolution which can be gained by using multiple scans of CMs, is lost. By using multiple scans it is also possible to modify previous assignments (excluding assignments from the scan at the tail of the sliding window, which cannot be modified in the next scan) which is not possible in 2-D dynamic assignment. Using a multidimensional algorithm (S-D) rather than a 2-D algorithm may yield better tracking results. In S-D dynamic assignment a sliding window of the latest $S-1$ scans of CMs is to be assigned to a list of established tracks. At scan k, a list of pre-existing tracks from scan $k-S+1$, is to be assigned to CMs from the window of scans with depth $S-1$, i.e., measurements from scan $k-S+2$ to the current scan k, using a likelihood ratio based cost function. It is well known that when $S \geq 3$, the MDA is an NP-hard problem, i.e., the complexity of an optimal algorithm increases exponentially with the size (number of dimensions) of the problem, as a result suboptimal algorithms with acceptable accuracy have to be solved that have polynomial complexity. In [30] a 3-D assignment problem was solved using a Lagrangian relaxation technique that successively solves a series of generalized 2-D assignment problems with the worst case complexity of $\mathcal{O}(3in^3)$ [30], where i is the number of relaxation iterations and n is the number of reports in each scan (or from each sensor in the case of static MDA discussed in Section 4). However, when $S \geq 4$ multiple sets of constraints have to be relaxed. In [31] the constraints are relaxed one set at a time corresponding to each list and the resulting $S-1$-D assignment problem is solved iteratively with a feasible solution to the original S-D problem being reconstructed subsequently. In the present paper the technique developed in [14] is used where all the S-2 lists are relaxed simultaneously. The problem then is a 2-D assignment problem and can be solved optimally in polynomial time. The Lagrangian multipliers associated with multiple constraint sets are updated simultaneously for faster convergence [14]. The computational complexity is $\mathcal{O}(i(S-1)Cn^3)$ where i is the maximum number of iterations, C is the range of the cost coefficient and n is the number of reports at each scan.

5.1. Conventional-Cost Based S-D Dynamic Assignment

The likelihood that the $S-1$-tuple of CMs $z_{m_{k-S+2}}(k-S+2),\ldots,z_{m_k}(k)$ is from track $T_u(k-S+1)$, where $m \in \{0,1,\ldots,M\}$ and $u \in \{1,\ldots,U(k-S+1)\}$ is given by

$$\begin{aligned}&\Lambda_{m_{k-S+2},\ldots,m_k,u}(k) \\ &\quad = p[z_{m_{k-S+2}}(k-S+2),\ldots,z_{m_k}(k) \mid T_u(k-S+1)] \\ &\quad = \prod_{i=k-S+2}^{k}(1-P_D)^{1-\delta_i}\cdot[P_D\Lambda_{m_i,u}(i)]^{\delta_i}\end{aligned} \tag{33}$$

[10] S denotes the number of scans in dynamic assignment, it has no relation to the number of sensor arrays used in the static case.

[11] The 2-D dynamic assignment is a special case of the S-D sliding window dynamic assignment algorithm where the size of the window is 1, i.e., $S=2$.

where

$$\Lambda_{m_i,u}(i) = p[z_{m_i}(i) \mid T_u(k-S+1), \{z_{m_j}(j)\}_{j=k-S+2}^{i-1}] \tag{34}$$

is the likelihood if the CM $z_{m_i}(i)$ is associated with track $T_u(k-S+1)$ continued with measurements $\{z_{m_j}(j)\}_{j=k-S+2}^{i-1}$ [4]. The kinematic CM prediction errors are assumed to be independent across lists, and $z_{m_i}(i)$ is the m_ith CM at scan i. The probability of detection and the indicator function of measurement $z_{m_i}(i)$ are P_D and δ_i, respectively, with the latter defined as

$$\delta_i = \begin{cases} 1 & \text{if} \quad m_i > 0 \\ 0 & \text{if} \quad m_i = 0 \end{cases}. \tag{35}$$

The likelihood that the measurements forming the $S-1$-tuple $z_{m_{k-S+2}}(k-S+2), \ldots, z_{m_k}(k)$ are from none of the established tracks (i.e., they are from false alarms)[12] is

$$\begin{aligned} &\Lambda_{m_{k-S+2},\ldots,m_k,0}(k) \\ &\quad = p[z_{m_{k-S+2}}(k-S+2), \ldots, z_{m_k}(k) \mid T_0(k-S+1)] \\ &\quad = \prod_{i=k-S+2}^{k} \left[\frac{1}{V}\right]^{\delta_i} \end{aligned} \tag{36}$$

where a uniform distribution is assumed, with V being the volume of the surveillance region.

The cost of assigning an $S-1$-tuple, $z_{m_{k-S+2}}(k-S+2), \ldots, z_{m_k}(k)$, of CMs to a track $T_u(k-S+1)$ where $u \in \{0, 1, \ldots, N(k-S+1)\}$ is

$$c_{m,u,S} = \begin{cases} 0 & \text{if} \quad u = 0 \\ -\ln\left[\dfrac{\Lambda_{m_{k-S+2},\ldots,m_k,u}(k)}{\Lambda_{m_{k-S+2},\ldots,m_k,0}(k)}\right] & \text{if} \quad u > 0 \end{cases} \tag{37}$$

where the two likelihood functions are defined in (33) and (36).

5.2. Feature-Aided S-D Dynamic Assignment

In the conventional-cost based S-D dynamic assignment approach, only the kinematic CMs obtained by the sensor network are used for target tracking and the information contained in the feature vectors is lost. The solution of the static MDA algorithm described in Section 4 results in $M(i)$ CMs, at scan i, that include the feature vectors corresponding to the assigned DoA angle measurements that give rise to each CM. A composite feature vector $\Omega_{m_i}(i)$, where i is the scan index $i \in \{k-S+2, \ldots, k\}$, can be obtained from the S-tuple of feature vectors corresponding to the S-tuple of assigned DoA angle measurements that give rise to $z_{m_i}(i)$. Appendix A describes the procedure to obtain a composite feature vector.

[12] All the measurements from time $k-S+2$ deemed false by the assignment are used to initialize new tracks.

In feature-aided S-D dynamic assignment, a sliding window of the latest $S-1$ scans (lists) of *feature-augmented CM vectors*, rather than just kinematic CMs, is assigned to an established track list. At scan k, one has a list of existing tracks at scan $k-S+1$. To this list one assigns the feature-augmented CM vectors from a window of scans with a window depth of $S-1$, i.e., measurement vectors from scan $k-S+2$ to current scan k, using a likelihood ratio based cost function. Each CM $z_{m_i}(i)$, at scan $i \in \{k-S+2, \ldots, k\}$, is augmented by its corresponding composite feature vector $\Omega_{m_i}(i)$ to form a feature-augmented CM vector (using bold face notation)

$$\mathbf{z}_{m_i}(i) = [z_{m_i}(i)' \quad \Omega_{m_i}(i)']'. \tag{38}$$

The likelihood that the $S-1$-tuple of feature-augmented CM vectors $\mathbf{z}_{m_{k-S+2}}(k-S+2), \ldots, \mathbf{z}_{m_k}(k)$ is from track $T_u(k-S+1)$, where $m \in \{0, 1, \ldots, M\}$ and $u \in \{1, \ldots, U(k-S+1)\}$ is given by (again using bold face notations)

$$\begin{aligned} &\boldsymbol{\Lambda}_{m_{k-S+2},\ldots,m_k,u}(k) \\ &\quad = p[\mathbf{z}_{m_{k-S+2}}(k-S+2), \ldots, \mathbf{z}_{m_k}(k) \mid T_u(k-S+1)] \\ &\quad = \prod_{i=k-S+2}^{k} (1-P_D)^{1-\delta_i} \cdot [P_D \boldsymbol{\Lambda}_{m_i,u}(i)]^{\delta_i} \end{aligned} \tag{39}$$

where

$$\boldsymbol{\Lambda}_{m_i,u}(i) = p[\mathbf{z}_{m_i}(i) \mid T_u(k-S+1), \{\mathbf{z}_{m_j}(j)\}_{j=k-S+2}^{i-1}] \tag{40}$$

where P_D and δ_i are the probability of detection and the indicator function, respectively, of $z_{m_i}(i)$. Assuming independence between the kinematic CM prediction errors and their corresponding composite feature vectors, we have using (38)

$$\begin{aligned} &\boldsymbol{\Lambda}_{m_i,u}(i) \\ &\quad = p[z_{m_i}(i) \mid T_u(k-S+1), \{z_{m_j}(j)\}_{j=k-S+2}^{i-1}] p[\Omega_{m_i,u}(i)] \\ &\quad = \Lambda_{m_i,u}(i) p[\Omega_{m_i,u}(i)] \end{aligned} \tag{41}$$

where $\Lambda_{m_i,u}(i)$ is the (kinematic) likelihood defined in (34) and $\Omega_{m_i,u}(i)$ is the composite feature vector at scan i matched to the features from track u whose distribution is derived in Appendix A.

The likelihood that the feature-augmented CM vectors forming the $S-1$-tuple $\mathbf{z}_{m_{k-S+2}}(k-S+2), \ldots, \mathbf{z}_{m_k}(k)$ are false alarms, assuming independence between the kinematic measurement errors and the features as before, is

$$\begin{aligned} &\boldsymbol{\Lambda}_{m_{k-S+2},\ldots,m_k,0}(k) \\ &\quad = p[\mathbf{z}_{m_{k-S+2}}(k-S+2), \ldots, \mathbf{z}_{m_k}(k) \mid T_0(k-S+1)] \\ &\quad = \prod_{i=k-S+2}^{k} \{p[z_{m_i}(i) \mid T_0(k-S+1)] p[\Omega_{m_i}(i) \mid T_0(k-S+1)]\}^{\delta_i}. \end{aligned} \tag{42}$$

Assuming uniform distributions like before, we have

$$\mathbf{\Lambda}_{m_{k-S+2},\ldots,m_k,0}(k) = \prod_{i=k-S+2}^{k} \left\{ \frac{1}{V} p[\Omega_{m_i,0}(i)] \right\}^{\delta_i} \quad (43)$$

where V is the volume of the surveillance region, $\Omega_{m_i,0}(i)$ represents the matched feature vector at scan i, if its elements are from clutter, and its distribution is given in Appendix A.

The cost of assigning an $S-1$-tuple of feature-augmented CM vectors $\mathbf{z}_{m_{k-S+2}}(k-S+2),\ldots,\mathbf{z}_{m_k}(k)$ to a track $T_u(k-S+1)$ where $u \in \{0,1,\ldots,N(k-S+1)\}$ is

$$\mathbf{c}_{m,u,S} = \begin{cases} 0 & \text{if} \quad u = 0 \\ -\ln\left[\dfrac{\mathbf{\Lambda}_{m_{k-S+2},\ldots,m_k,u}(k)}{\mathbf{\Lambda}_{m_{k-S+2},\ldots,m_k,0}(k)}\right] & \text{if} \quad u > 0 \end{cases} \quad (44)$$

where the two likelihood functions are defined in (39) and (43).

5.3. Track Initiation

A one-point initialization technique [40] is used to initialize tracks at the beginning of the scenario (i.e., $k = 0$),[13] where $M(0)$ tracks are initialized at scan 0 from $\{z_m(0)\}_{m=1}^{M(0)}$ measurements. The initial state estimate of each track is of the form $\hat{x}_u(0 \mid 0) = [\xi_m \;\; \eta_m \;\; \dot{\xi}_m \;\; \dot{\eta}_m]$ where u is the track index. The velocity state components $\dot{\xi}_m$ and $\dot{\eta}_m$ are initialized at 0 m/s, as the motion direction of the target is assumed unknown. The initial state estimate covariance of each track is of the form

$$P_u(0 \mid 0) = \begin{bmatrix} R_m & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \left(\frac{v_{\max}}{2}\right)^2 \mathbf{I}_{2\times 2} \end{bmatrix} \quad (45)$$

where $v_{\max}$ is the assumed maximum target speed and R_m is the measurement covariance matrix of the 2-dimensional Cartesian CM z_m. As it is not readily available, it is approximated by the Cramer-Rao lower bound (CRLB) [2, 29, 32] corresponding to the likelihood function of the S-tuple of DoA angle measurements that yielded it, and $\mathbf{0}_{n\times m}$ and $\mathbf{I}_{n\times m}$ are the all-zero matrix and the identity matrix of dimension $n \times m$, respectively. The maximum target speed $v_{\max}$ is assumed to be 9 m/s in the present paper.

An interacting multiple model (IMM) estimator with two second-order linear kinematic models (white noise acceleration, WNA) with two process noise levels is used (see [2], Ch. 11). The one with the lower noise level (with standard deviation 0.25 m/s^2) is used to model the uniform motion and the other one with standard deviation 5 m/s^2 for the maneuvers. The mode transition probability matrix

$$\pi_{ij} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \quad (46)$$

is used. A 5-sigma ($\gamma = 25$) validation region (see [1], Ch. 3) is used for gating, and in both 2-D and S-D dynamic assignment, only measurements that fall within the validation regions of tracks are candidates for assignment. A probability of detection of 0.85 is assumed for the CMs and 0.8 for the features used in feature-aided dynamic assignment. The volume of the entire surveillance region of the sensor network is assumed to be 100,000 m^2.

5.4. Track Confirmation and Deletion

After solving the S-D dynamic assignment problem (Sections 5.1 and 5.2) at each scan, each assigned measurement from scan $k-S+2$, i.e., the scan at the tail of the sliding window, is used to update the existing tracks, and the unassigned measurements are allowed to form new candidate tracks. The unassigned measurements from scans $k-S+3$ to k are retained as is in the sliding window for the S-D assignment at the next scan. Hence, they are given again a chance to be assigned to tracks. Candidate tracks are formed solely for track initiation and they can be thought of as "second class" members of the track list while the established tracks can be thought of as "first class" members. A candidate track is either confirmed as an updated track or rejected, after a maximum of 5 scans, using a Markov chain cascaded logic (2/2&2/3) (see [1, Sec. 2.6.3]). A confirmed track is deleted if it is not assigned any measurements in 3 consecutive scans. Fig. 5 is a flowchart describing the S-D dynamic assignment based tracker.

6. REAL DATA SCENARIO

The algorithm developed above was exercised on real data obtained from a field experiment conducted by the U.S. Army Research Laboratory [12] at Aberdeen Proving Ground, Maryland. A passive acoustic sensor network was placed within a path that was traveled by two targets, a heavy vehicle and a light vehicle. Fig. 6 shows the actual trajectories based on the instrumentation that was installed in the two vehicles. However, due to some alignment errors these could not be used as ground truth for our evaluation. They will, however, be used to compare the life of the tracks with the duration of these trajectories.

All the acoustic sensor arrays used in the experiment were circular arrays of microphones. The sampling rate for each array was set at 1024 Hz, with a cutoff frequency of 312 Hz and a gain of 100 selected for all microphones in the array. The acoustic sensor arrays that make up the passive sensor network were located at known fixed positions $\mathbf{x}_s$, $s = 1,\ldots,S$, where $S = 4$ is

[13] This is preferable to two-point initialization when, due to the small sampling interval (1 s) the variance of the two-point velocity estimate [1] is significantly larger than the maximum speed.

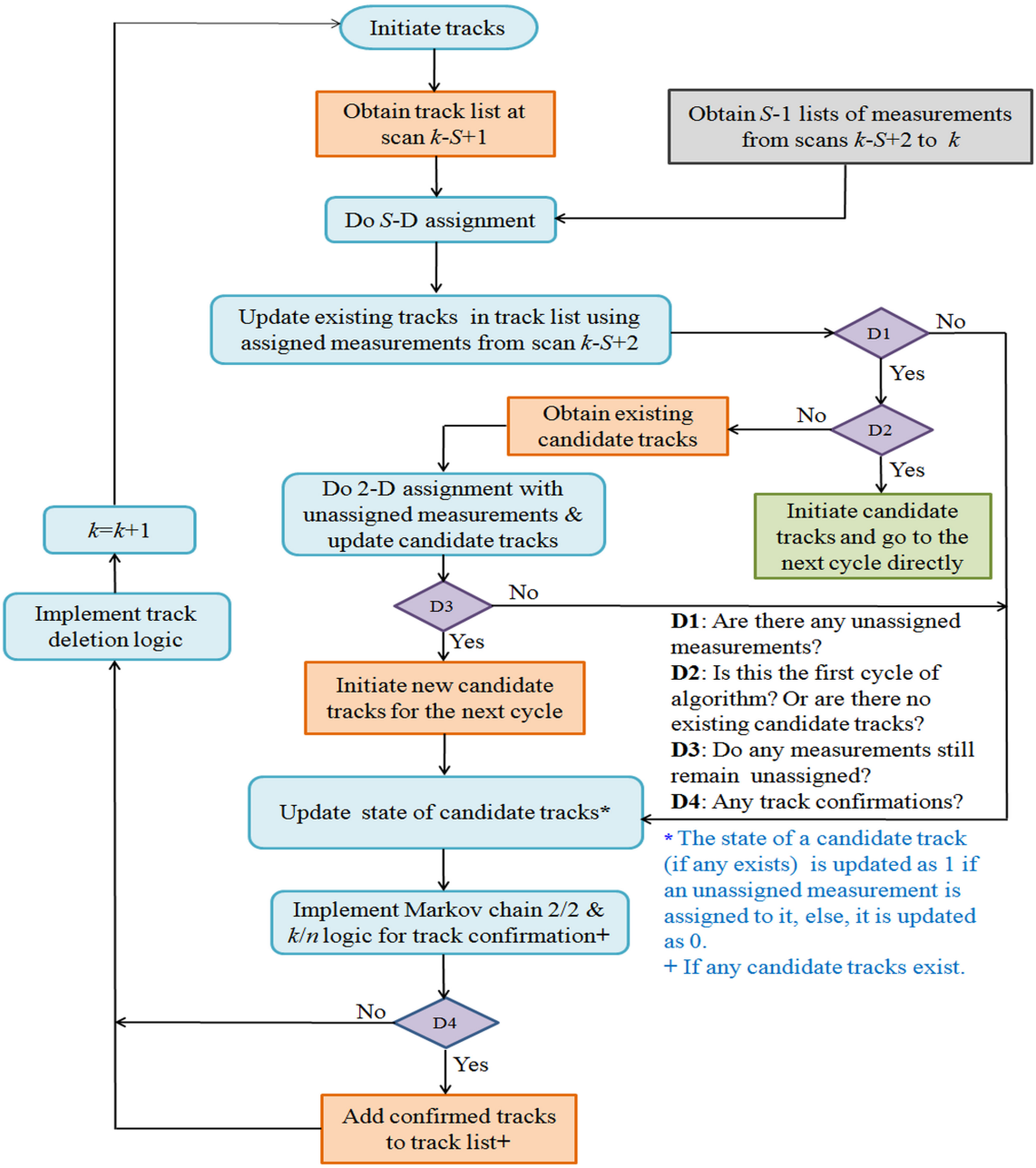


Fig. 5. Flowchart describing the dynamic S-D assignment-based tracker.

the number of sensor arrays. Each sensor array has a field of view of 360° and is made up of 7 equi-distant microphones as shown in Fig. 1. DoA angle measurements are estimated at each sampling time (1 s), from the PSD of the acoustic signals received by each sensor array, with an angular measurement error standard deviation of 1°, as described in Section 2. A low-pass spectrum of 1–120 Hz, divided into 120 bins of 1 Hz each, is used by the MVDR algorithm to estimate the PSD. The probability of detection of a DoA angle is assumed to be 0.8, the probability of detection of the features is assumed to be 0.8 and the surveillance region volume in frequency is assumed to be V_s^f = 120 Hz (the width of the frequency spectrum).

7. RESULTS

Fig. 7 shows the tracking results obtained using the 2-D dynamic assignment algorithm for a 2 target scenario: one target (a heavy vehicle) traveling counterclockwise on an oval gravel path and the other target (a light vehicle) on a southeast-northwest asphalt path. The scenario starts at time 50 s and ends at time 185 s.[14]

Fig. 7(a) shows the tracks obtained when the 2-D dynamic assignment tracker uses the CMs obtained from the conventional-cost based static (without fea-

[14]This is the time interval when there is a significant ambiguity due to the interference between the two vehicles—the second target appears shortly after time 50 s.

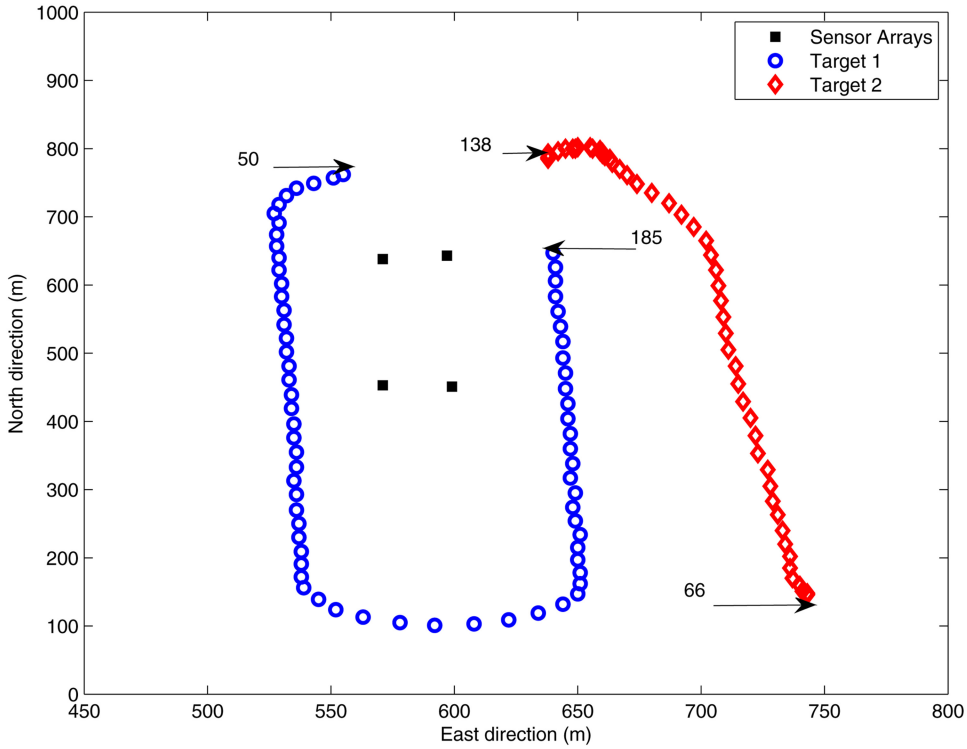


Fig. 6. True trajectories along with the start and end times (in seconds) for each target.

tures) MDA algorithm described in Section 4.1. One can see from comparing Fig. 7(a) with Fig. 6 that there are two false tracks at the bottom of the oval path when target 1 is turning. Track 2 is confirmed at scan 96 and continues towards the north, incorrectly, between scans 120 and 126 after which it gets back on track (compare with Fig. 6). In the case where the tracker uses CMs obtained from the feature-aided static MDA algorithm (Section 4.2), shown in Fig. 7(b), the tracker is able to follow the maneuver of target 1 at the bottom of the oval path better than the one in Fig. 7(a). Track 2 is confirmed at scan 89 (i.e., 7 s earlier) and is not lost until scan 117, after which it is reacquired at scan 125. It can also be seen from both Figs. 7(a) and 7(b) that track continuity is not as good at the bottom of the oval track as when the target is closer to the sensors, as the CMs obtained in that region are ill-conditioned because the DoA angle measurements from all 4 arrays are very closely spaced.

Fig. 8 shows the results obtained when the S-D assignment tracker with $S = 4$, i.e., when 2 additional lists of measurements are added. This improves over the performance when $S = 2$, as well as when $S = 3$ (the latter not shown). For $S > 4$, there is no further perceptible performance improvement. In the result shown in Fig. 8(a), when the CMs obtained from the conventional-cost based (without features) static MDA algorithm are used, target 1 is lost at scan 119 and there is a false track between scans 130–135. Target 1 is reacquired at scan 136 and continues till the end of the scenario (scan 185), along the oval path. The tracker has a false track between scans 83–89 and reacquires target 2 at scan 96 and keeps it until scan 127. In the result shown in Fig. 8(b), one can see that while track 1 is lost between scans 117–130, there are no false tracks like in Fig. 8(a). Track 2 is confirmed at scan 86 (10 s sooner than without features) and is not lost till it stops moving (130 s). The 4-D dynamic assignment based tracker performs much better at the bottom of the oval track when features are used to obtain CMs, when compared to the case where features are not used to obtain CMs.

Fig. 9 shows the results obtained when the feature-aided dynamic 2-D and feature-aided dynamic 4-D assignment based trackers are used. In the scenario shown in Fig. 9(a), when the feature-aided dynamic 2-D tracker is used, target 1 is lost at scan 120 (instead of 117 in the dynamic assignment without features) and is reacquired at scan 128 (vs. 130). Track 2 is confirmed at scan 86 and is not lost till target 2 stops moving. In the results shown in Fig. 9(b), target 1 is lost between scans 120–128. Track 2 is confirmed at scan 86 and is not lost till target 2 stops moving. From Fig. 9(a) it can be seen that when the 2-D feature-aided tracker is used, target 1 is lost between scans 65–69; this loss is eliminated when the 4-D feature aided tracker is used, as shown in Fig. 9(b). While there is a clear benefit from using features in the static association, the results in the dynamic part indicate that the benefits are that target 1 is lost at the bottom of the oval path for a shorter duration when compared to the dynamic assignment scenarios where features are not used.

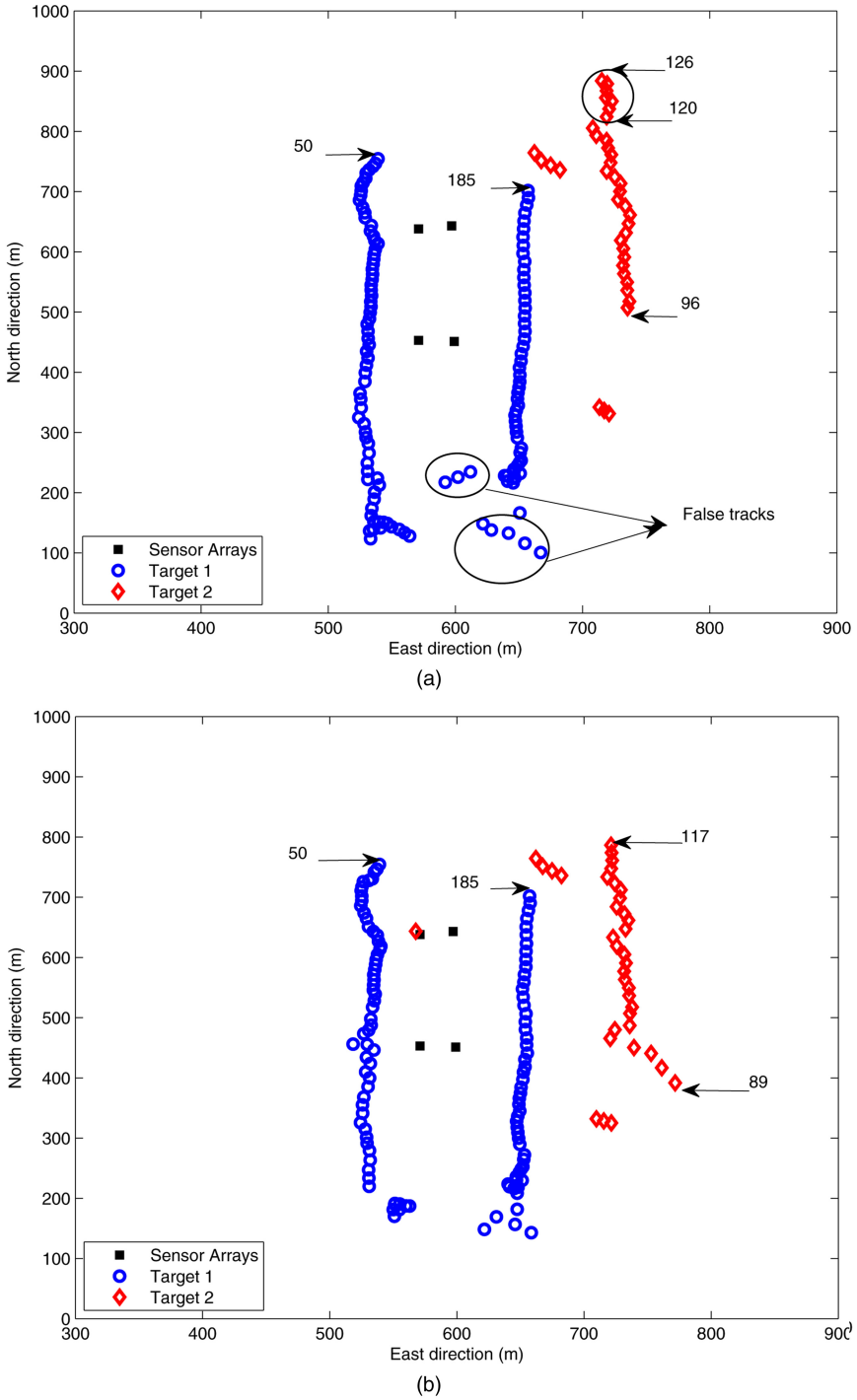


Fig. 7. Target tracking using conventional-cost based 2-D dynamic assignment: (a) using CMs obtained from ***conventional-cost based*** static MDA, and (b) using CMs obtained from ***feature-aided*** static MDA.

8. CONCLUSIONS

In this paper a static data association algorithm to obtain full position estimates (composite measurements —CMs) of multiple ground targets using real acoustic signal data, obtained by a passive sensor network, has been presented using a novel feature-aided static MDA (multidimensional assignment) framework. While the methodology developed is general, it is illustrated on real data collected by a sensor network comprised of four sensor arrays which listen to two vehicles, a heavy vehicle and a lighter vehicle. The CMs are assigned to tracks using dynamic 2-D and S-D (with $S = 4$) assignment.

A novel detection scheme has been presented to detect DoAs from real acoustic data and a new feature extraction technique based on GMM (Gaussian mixture model) based fitting and multidimensional matching has been presented to extract feature vectors which aug-

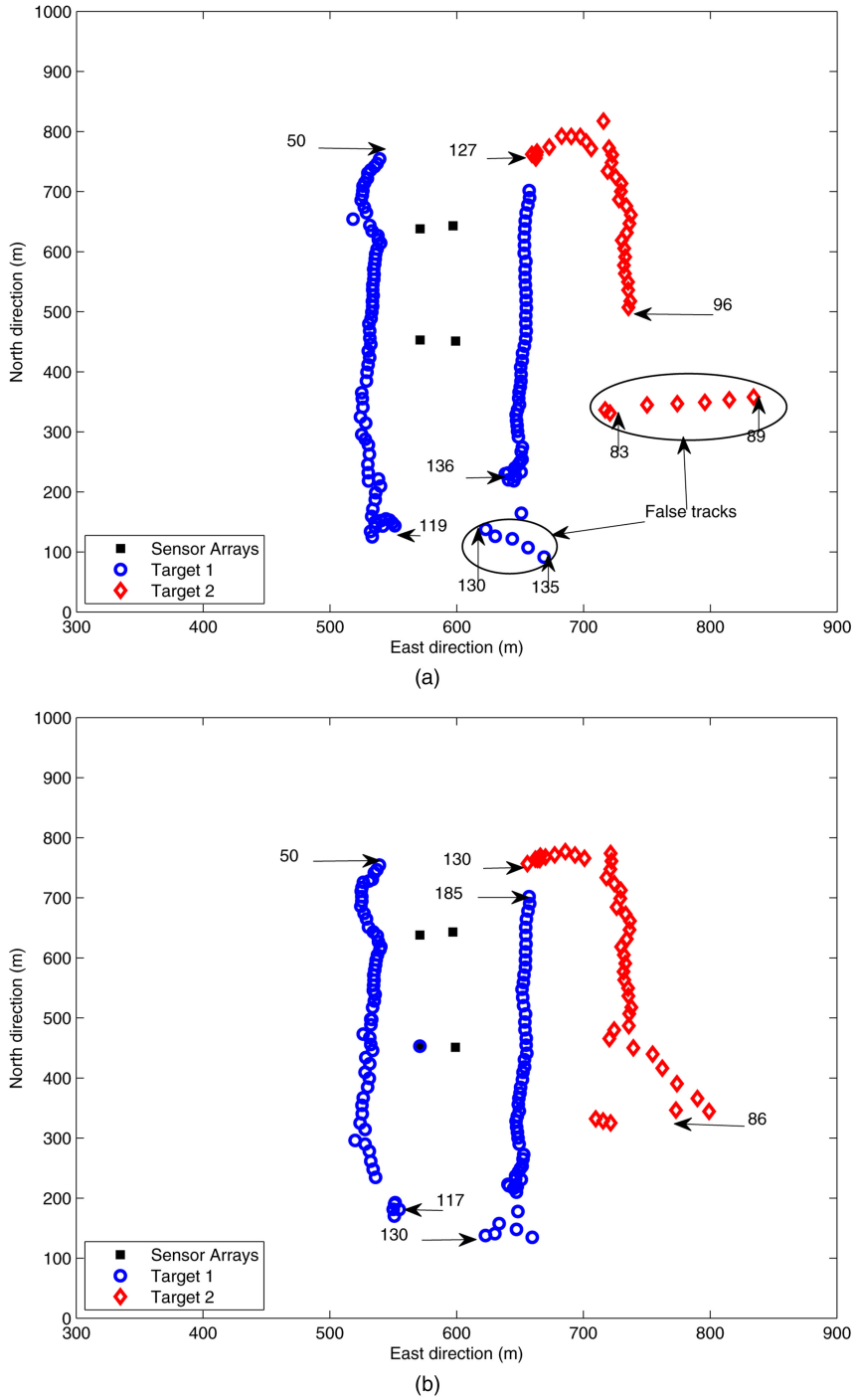


Fig. 8. Target tracking using conventional-cost based 4-D dynamic assignment: (a) using CMs obtained from ***conventional-cost based*** static MDA, and (b) using CMs obtained from ***feature-aided*** static MDA.

ment the corresponding DoA angle measurements. An MDA algorithm is solved at each scan, using feature-augmented likelihood ratio based cost functions, to obtain composite measurements that are the full position estimates of targets. Composite measurements are assigned to tracks using both conventional cost-based as well as feature-aided S-D dynamic assignment.

It is observed from running the dynamic 2-D and 4-D assignment algorithms on real data that the tracking performance is significantly better when features are used to generate the CMs. One track is initiated earlier, and the breakages in the other are reduced. It is also observed that the tracks using the feature-aided tracker have shorter breakages. Its performance could be improved if the shifts in frequency due to the motion of vehicles are accounted for. This is a topic for further research.

Modeling of the variation in time of the features and track segment association can be further future topics of investigation.

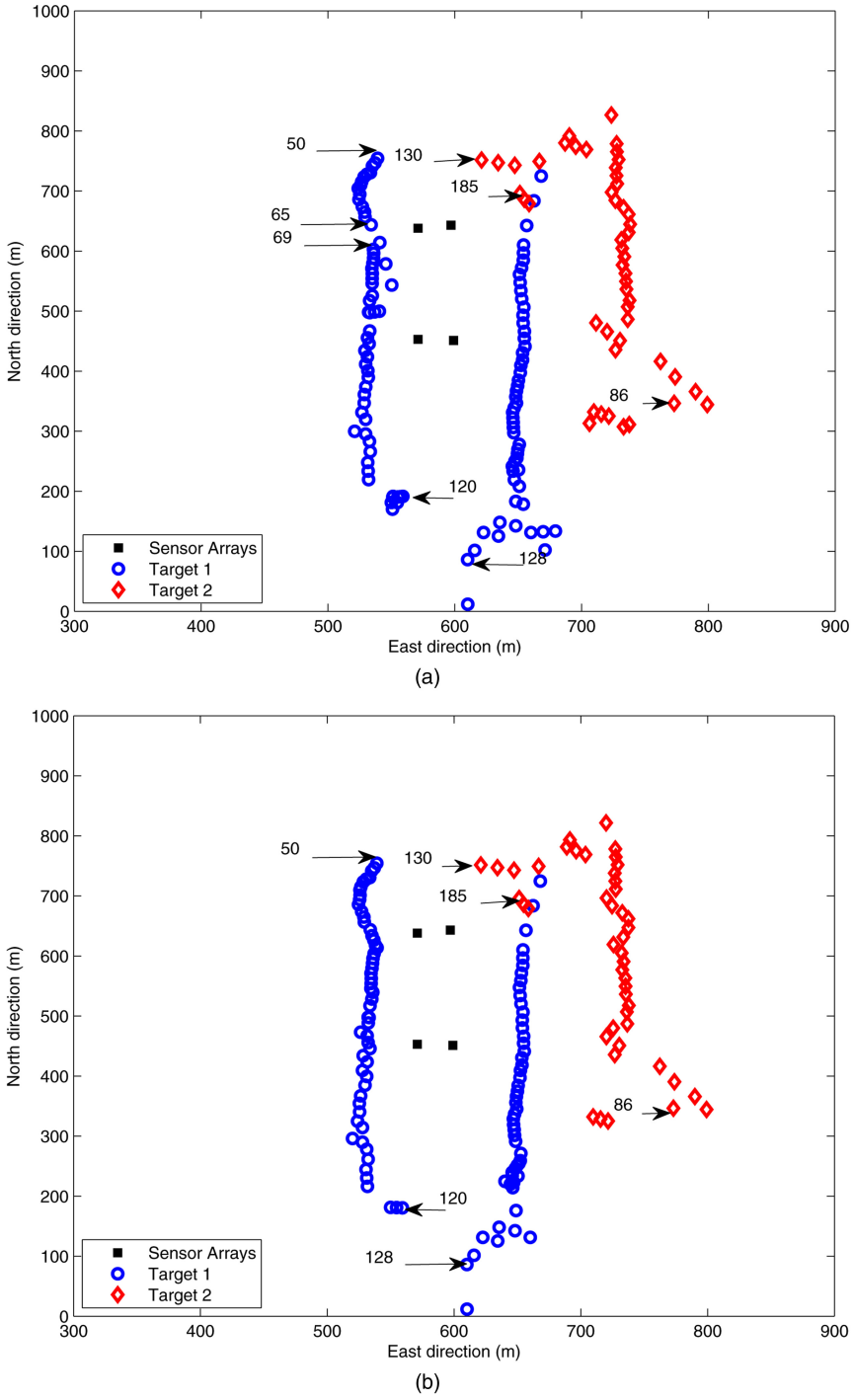


Fig. 9. Target tracking using feature-augmented cost based 2-D and 4-D dynamic assignment: (a) feature-aided 2-D dynamic assignment, and (b) feature-aided 4-D dynamic assignment.

9. ACKNOWLEDGMENT

This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant 51018-CI. The authors thank Dr. Javier Areta for his help in implementing the Lagrangian relaxation based MDA algorithm. The authors are also grateful to the Editor and the reviewers for their comments and suggestions to improve the paper.

APPENDIX A. THE COMPOSITE FEATURE VECTOR

The composite feature vector $\Omega_{m_i}(i)$ corresponding to the CM $z_{m_i}(i)$ is given by

$$\Omega_{m_i}(i) = [\omega^1_{m_i}(i), \omega^2_{m_i}(i), \ldots, \omega^{n_{m_i}}_{m_i}(i)]'. \qquad (47)$$

Each element $\omega^l_{m_i}(i)$ in (47) is the weighted average of the lth elements of the feature vectors corresponding to the assigned S-tuple of DoA angle measurements that give rise to $z_{m_i}(i)$ and is obtained using (28).

The composite feature vector corresponding to track T_u is the composite feature vector corresponding to the CM that has been assigned and used to update the track u and is given by

$$\Omega_u = [\omega_u^1, \omega_u^2, \ldots, \omega_u^{n_u}]'. \tag{48}$$

The composite feature vector in (47) is matched to the composite feature vector in (48) using the same matching technique described in Section 3.2. The matched composite feature vector at scan i, i.e., $\Omega_{m_i,u}(i)$, is given by

$$\Omega_{m_i,u}(i) = [\omega_{m_i}^{j_1}(i), \omega_{m_i}^{j_2}(i), \ldots, \omega_{m_i}^{j_{n_{m_i,u}}}(i)]' \tag{49}$$

where $\{\omega_{m_i}^{j_q}\}_{q=1}^{n_{m_i,u}} \in \{\oslash, \omega_{m_i}^{l}(k)\}$; the resulting dummy elements (missed detections) after matching are represented by $\oslash$ and $\omega_{m_i}^{l}(i)$ is an element of the composite feature vector in (47), with $l \in \{1, \ldots, n_{m_i}\}$ being the index of its elements.[15] Similarly, the matched composite feature vector of track T_u is given by

$$\Omega_u^{\dagger}(k-1) = [\omega_u^{j_1}, \omega_u^{j_2}, \ldots, \omega_u^{j_{n_{m_i,u}}}]' \tag{50}$$

where $\{\omega_u^{j_q}\}_{q=1}^{n_{m_i,u}} \in \{\oslash, \omega_u^{l}\}$, with $l \in \{1, \ldots, n_u\}$ being the index of the elements of the composite feature vector (48).

The likelihood that the matched composite feature vector $\Omega_{m_i,u}(i)$ at scan k is from track T_u is given by

$$p[\Omega_{m_i,u}(i)] = \prod_{q=1}^{n_{m_i,u}} (1 - P_{D_q})^{1-\delta(m_{iq})} \cdot [P_{D_q} p(\omega_{m_i}^{j_q}(i))]^{\delta(m_{iq})} \tag{51}$$

assuming that the individual matched composite feature vector elements are uncorrelated and where P_{D_q} and $\delta(m_{iq})$ represent the probability of detection and the indicator function for the elements of the matched composite feature vector (49). The elements of the matched composite feature vector are assumed to be distributed as

$$p(\omega_{m_i}^{j_q}(i)) = \mathcal{N}(\omega_{m_i}^{j_q}(i); \hat{\omega}^{j_q}(i), \sigma_{j_q}^2) \tag{52}$$

where

$$\hat{\omega}^{j_q}(i) = \frac{\omega_{m_i}^{j_q}(i)\delta(m_{iq}) + \omega_u^{j_q}(i)\delta(u_q)}{\delta(m_{iq}) + \delta(u_q)} \tag{53}$$

$\delta(u_q)$ is the indicator function of the elements of the matched composite feature vector (48), and σ_{j_q} is the standard deviation of the distribution of $\omega_{m_i}^{j_q}(k)$ and is assumed to be the minimum of all the standard deviations $\sigma_{si_s}^{l}$ (see Section 4.2) of the features that give rise to the composite feature vector.

The likelihood that the matched composite feature vector $\Omega_{m_i,u}(i)$ at scan i is from a source of clutter, assuming a uniform distribution with V^f being the volume of the surveillance region of the sensor network in frequency, is given by

$$p[\Omega_{m_i,0}(i)] = \prod_{q=1}^{n_{m_i,u}} \left[\frac{1}{V^f}\right]^{\delta(m_{iq})}. \tag{54}$$

[15] Each element of (47) appears only once in (49).

REFERENCES

[1] Y. Bar-Shalom and X. Li
Multitarget-Multisensor Tracking: Principles and Techniques.
Storrs, CT, YBS Publishing, 1995.

[2] Y. Bar-Shalom, X. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation.
Wiley, 2001.

[3] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe
Sonar tracking of multiple targets using joint probabilistic data association.
IEEE Journal of Oceanic Engineering, **8**, 3 (July 1983), 173–184.

[4] Y. Bar-Shalom, T. Kirubarajan, and C. Gokberk
Tracking with classification-aided multiframe data association.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 3 (July 2005), 868–879.

[5] Y. Bar-Shalom, S. S. Blackman, and R. J. Fitzgerald
Dimensionless score function for multiple hypothesis tracking.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 1 (Jan. 2007), 392–400.

[6] D. Bertsekas
The auction algorithm: A distributed relaxation method for the assignment problem.
Annals of Operations Research: Special Issue on Parallel Optimization, **14** (1988), 105–123.

[7] B. Bhanu
Automatic target recognition: State of the art survey.
IEEE Transactions on Aerospace and Electronic Systems, **22**, 4 (July 1986), 364–379.

[8] E. Blasch and L. Hong
Simultaneous feature-based identification and track fusion.
In *Proceedings of IEEE International Conference on Decision and Control*, Tampa, FL, Dec. 1998, 239–244.

[9] J. C. Chen, K. Yao, and R. E. Hudson
Source localization and beamforming.
IEEE Signal Processing Magazine, (Mar. 2002), 30–39.

[10] H-W. Chen and J-W. Zhao
Wideband MVDR beamforming for acoustic vector sensor linear array.
In *IEE Proceedings of Radar, Sonar and Navigation*, **151**, 3 (June 2004), 158–162.

[11] T. R. Damarla
Tracking a convoy of multiple targets using acoustic sensor data.
In M. K. Masten and L. A. Stockum (Eds.), *Proceedings of SPIE Acquisition, Tracking and Pointing XVII*, vol. 5082, Orlando, FL, Apr. 2003, 37–42.

[12] T. R. Damarla, T. Pham, and D. Lake
An algorithm for classifying multiple targets using acoustic signatures.
In *Proceedings of SPIE Sig. Proc., Sensor Fusion and Targ. Recognition*, 2004, 421–427.

[13] T. R. Damarla and G. Whipps
Multiple target tracking and classification improvement using data fusion at node level using acoustic signals.
In *Proceedings of SPIE Unattended Ground Sensor Technologies and Applications VII*, vol. 5796, Orlando, FL, Apr. 2005, 19–27.

[14] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom
A generalized S-D assignment algorithm for multisensor-multitarget state estimation.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 2 (Apr. 1997), 523–538.

[15] O. E. Drummond
Feature, attribute and classification aided target tracking.
In *Proceedings of SPIE Signal and Data Processing of Small Targets*, vol. 4473, San Diego, CA, Aug. 2001, 610–621.

[16] O. E. Drummond
On categorical feature aided tracking.
In *Proceedings of SPIE Signal and Data Processing of Small Targets*, vol. 5204, San Diego, CA, Aug. 2003, 544–558.

[17] N. M. Laird, A. P. Dempster, and D. B. Rubin
Maximum likelihood from incomplete data via the EM algorithm.
J. Roy. Soc. Statist., Series B, **39**, 1 (1977), 138.

[18] A. V. Goldberg, S. A. Plotkin, and E. Tardos
Combinatorial algorithms for the generalized auction problem.
Mathematics in Operations Research, **16** (1991), 351–381.

[19] B. Gu and L. Hong
Tracking 2-D rigid targets with invariant constraints.
Information Science, **138** (2001), 79–97.

[20] M. E. Holil, A. Rotolo, and J. Chang
A cylinder classification and target ID algorithm based on a reduced feature space representation for ground vehicles.
In *Proceedings of 2003 Meeting of the IRIS Speciality Group on Acoustic and Seismic Sensing*, Sept. 2003.

[21] L. Hong, S. Cong, M. T. Pronobis, and S. Scott
Wavelets feature aided tracking (WFAT) using GMTI/HRR data.
Signal Processing, **86** (2003), 2683–2690.

[22] L. Hong, N. Cui, M. Pronobis, and S. Scott
Local motion feature aided ground moving target tracking with GMTI and HRR measurements.
IEEE Transactions on Automatic Control, **50**, 1 (Jan. 2005).

[23] B-C. Kim and I-T. Lu
High resolution broadband beamforming based on the MVDR method.
OCEANS 2000 MTS/IEEE Conference and Exhibition, vol. 3, 2000, 1673–1676.

[24] D. Lake
Harmonic phase coupling for battlefield acoustic target identification.
In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 4, Seattle, WA, May 1998, 2049–2052.

[25] D. Lake
Tracking fundamental frequency for synchronous mechanical diagonostic signal processing.
In *Proceedings of 9th IEEE Signal Processing Workshop on Statistical Signal and Array Processing*, Portland, OR, Sept. 1998, 200–203.

[26] W. Liu, S. Weiss, J. G. McWhirter, and I. K. Proudler
Frequency invariant beamforming for two-dimensional and three-dimensional arrays.
Signal Processing (Elsevier), **87**, 11 (Nov. 2007), 2535–2543.

[27] W. Liu and S. Ding
An efficient method to determine the diagonal loading factor using the constant modulus feature.
IEEE Transactions on Signal Processing, **56**, 12 (Dec. 2008), 6102–6106.

[28] D. H. Nguyen, J. H. Kay, B. J. Orchard, and R. H. Whiting
Classification and tracking of moving ground vehicles.
Lincoln Lab. Journal, **13**, 2 (2002), 275–308.

[29] K. Pattipati, R. Popp, and T. Kirubarajan
Survey of assignment techniques for multitarget tracking.
In Y. Bar-Shalom and W. D. Blair (Eds.), *Multitarget-Multisensor Tracking: Applications and Advances*, vol. III, Artech House, 2001.

[30] K. Pattipati, S. Deb, Y. Bar-Shalom, and R. Washburn
A new relaxation algorithm and passive sensor network data association.
IEEE Transactions on Automatic Control, **37**, 2 (Feb. 1992), 197–213.

[31] A. Poore and N. Rijavec
A Lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking.
SIAM J. Optimization, **3**, 3 (Aug. 1993), 544–563.

[32] R. L. Popp, K. R. Pattipati, and Y. Bar-Shalom
m-best S-D assignment algorithm with application to multitarget tracking.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 1 (Jan. 2001).

[33] Y. Ruan and L. Hong
Feature-aided tracking with GMTI and HRR measurements via mixture density estimation.
IEE Proceedings—Control Theory Appl., **153** 3, 342–356.

[34] N. Srour, D. Lake, and M. Miller
Utilizing acoustic propagation models for robust battlefield target identification.
In *Proceedings of 1998 Meeting of the IRIS Speciality Group on Acoustic and Seismic Sensing*, Sept. 1998.

[35] G. Succi and T. K. Pedersen
Acoustic target tracking and target identification—Recent results.
In *Proceedings of SPIE Conf. on Unattended Ground Sensor Technologies and Applications*, vol. 3713, Orlando, FL, Apr. 1999, 3713-10–21.

[36] H. L. Van Trees
Detection, Estimation and Modulation Theory, Part IV, Optimum Array Processing.
New York: Wiley, 2002.

[37] B. V. Veen
Minimum Variance Beamforming, Adaptive Radar Detection and Estimation.
Edited by S. Haykin and A. Steinhardt, J. Wiley & Sons, Inc., 1992.

[38] X. Wang and H. Qi
Acoustic target classification using distributed sensor arrays.
In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, Orlando, FL, May 2002, IV-4186.

[39] S. Yan and Y. Ma
High-resolution broadband beamforming and detection methods with real data.
Acoustic Science and Technology, **25**, 1 (2004), 73–76.

[40] S-W. Yeom, T. Kirubarajan, and Y. Bar-Shalom
Track segment association, finite-step IMM and initialization with Doppler for improved track performance.
IEEE Transactions on Aerospace and Electronic Systems, **40**, 1 (Jan. 2004), 293–309.

[41] R. E. Zarnich
A fresh look at broadband passive sonar processing.
In *1999 Adaptive Sensor Array Processing Workshop* (ASAP '99), MIT Lincoln Laboratory, Lexington, MA, Mar. 1999, 99–104 .

Vishal Cholapadi Ravindra (S'03) was born in Manipal, Karnataka, India on May 14, 1980. He received the B.E. degree in electronics and comminications from the University of Mysore (P.E.S.C.E. Mandya), India in 2001. He received the M.S. degree in electrical engineering from the University of Colorado at Denver, in 2004. In August 2009, he received the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs.

Since December 2009, he has been a Research Fellow at the National Aerospace Laboratories (NAL) in Bangalore, India. His research interests lie in estimation theory, target tracking, data fusion, automatic target recognition, and guidance and navigation for unmanned/micro aerial vehicles.

Yaakov Bar-Shalom (S'63—M'66—SM'80—F'84) was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering.

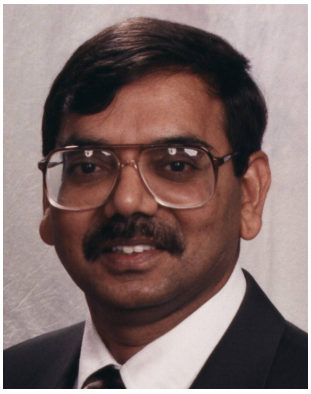
From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, CA. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab.

His current research interests are in estimation theory and target tracking and has published over 370 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate text *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000).

He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas.

During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of Automatica. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–08.

In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. He is the recipient of the 2008 IEEE Dennis J. Picard Medal for Radar Technologies and Applications.



Thyagaraju Damarla is working as an electronic engineer at the US Army Research Laboratory for the past 13 years. He received his B.S. and M.S. from Indian Institute of Technonology, Kharagpur, India and Ph.D. from Boston University.

He published more than 100 technical papers in various journals and conferences and has three US patents. His current interests include signal processing, and sensor fusion for situational awarenes.

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

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 least-

squares 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

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

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

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

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 piecewise 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

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

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

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

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.

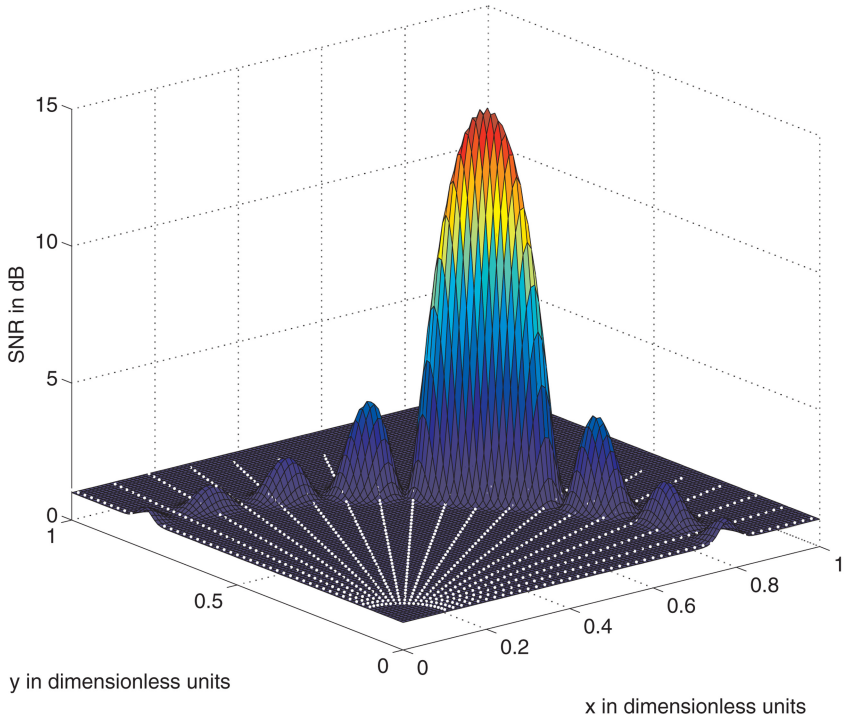


Fig. 1. Example response function for FM waveform.

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 $\times$. Two different sets of proposed sampling points are labeled, one with larger red $\times$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

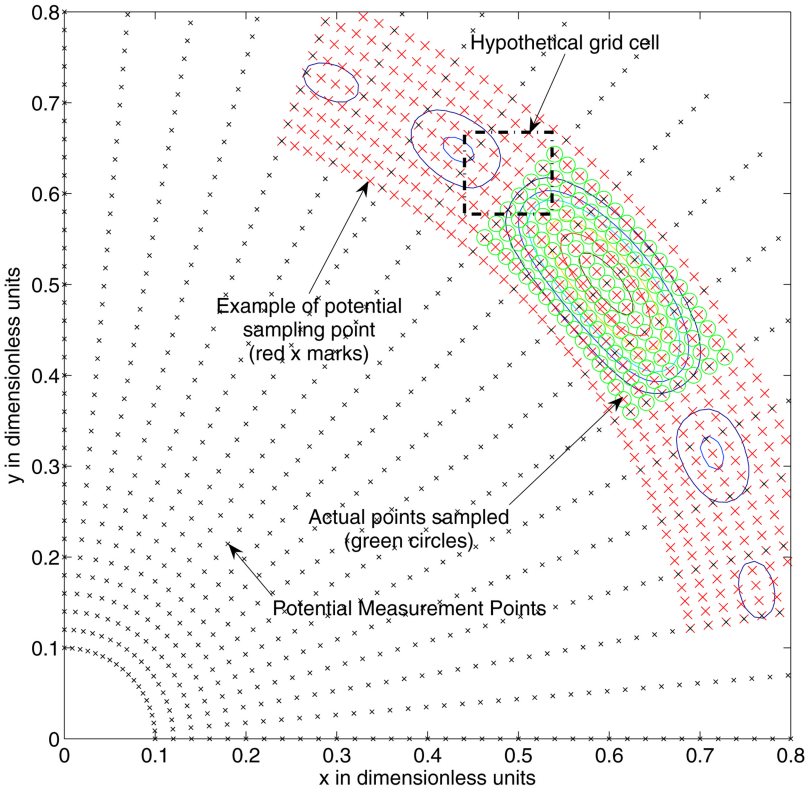


Fig. 2. Intelligent sampling example for FM waveform.

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 $\times$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

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

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

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

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

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

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

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

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

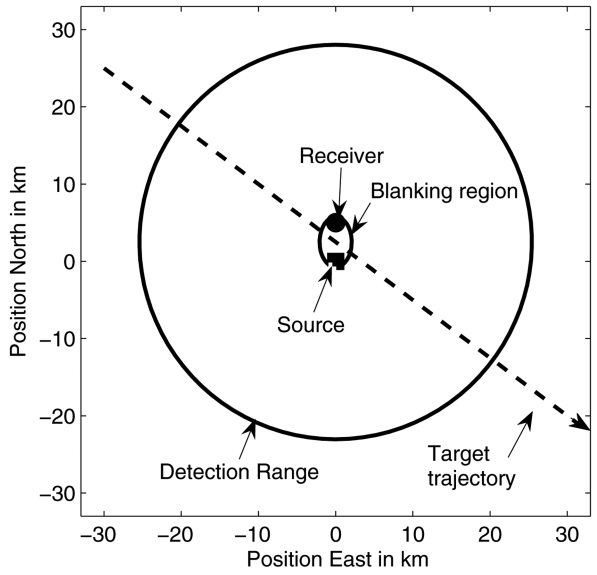


Fig. 3. Example problem geometry.

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

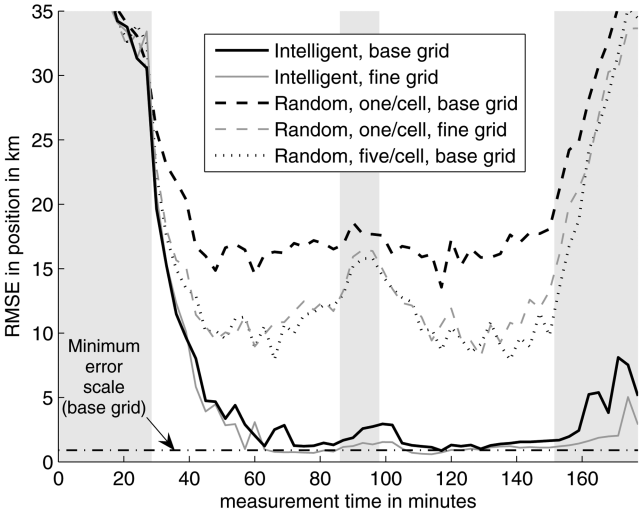


Fig. 4. Localization error for FM data. Gray regions indicate where the target is inherently undetectable.

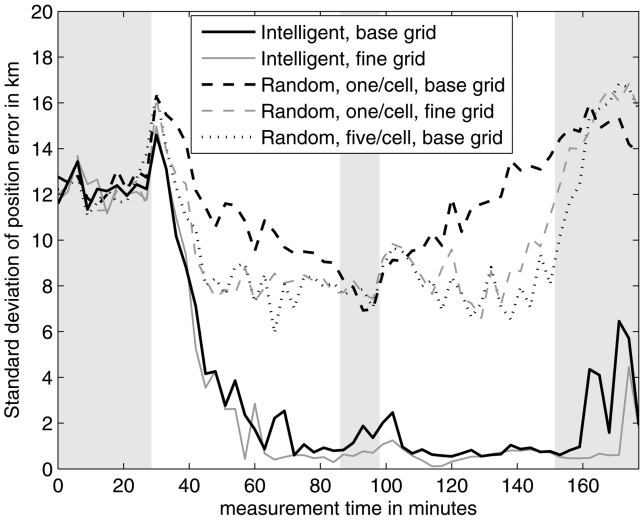


Fig. 5. Standard deviation of position error for FM data.

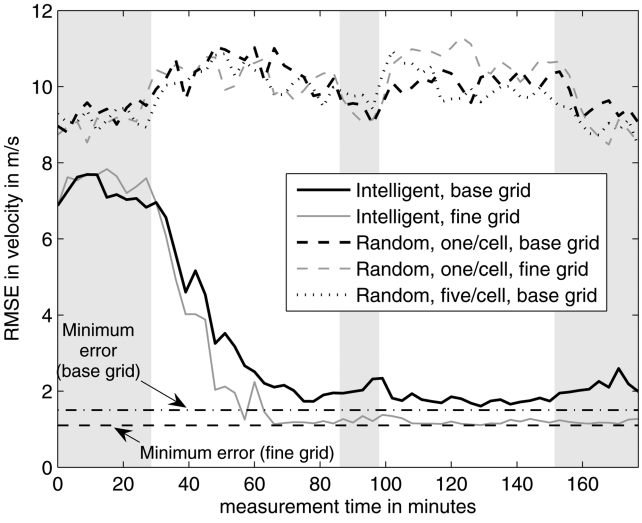


Fig. 6. Velocity error for FM data.

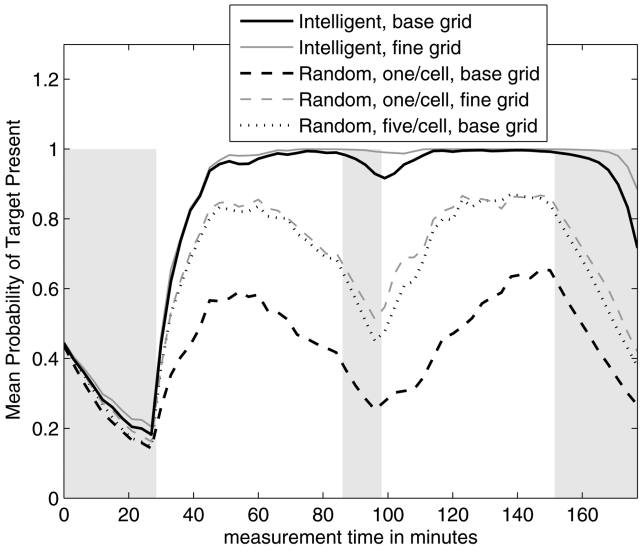


Fig. 7. Target probability for FM data.

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,

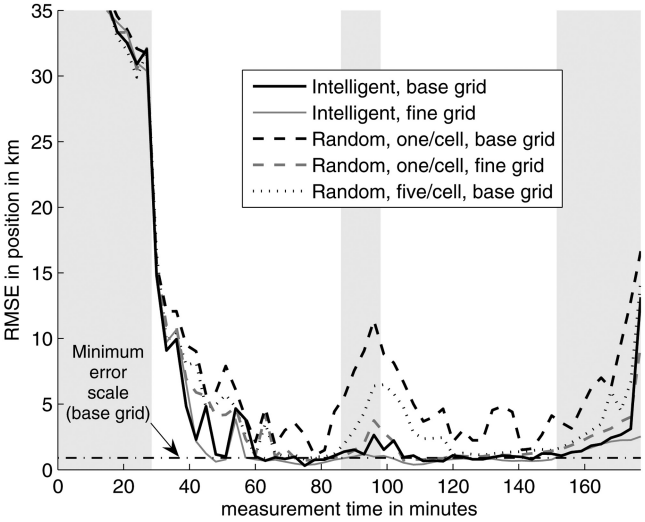


Fig. 8. Localization error for CW data. Gray regions indicate where the target is inherently undetectable.

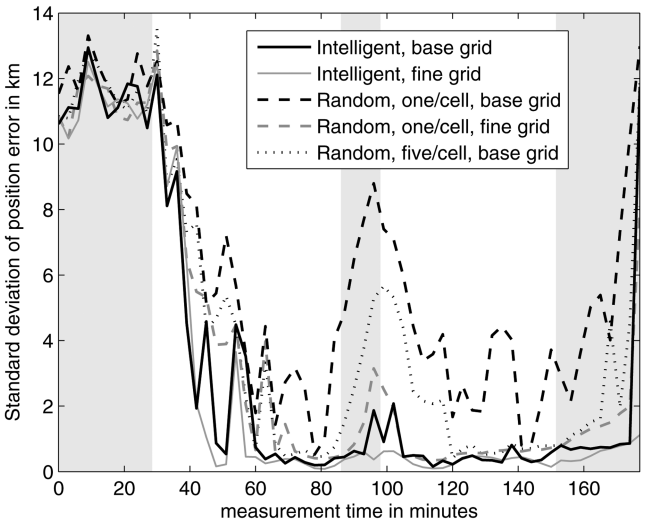


Fig. 9. Standard deviation of position error for CW data.

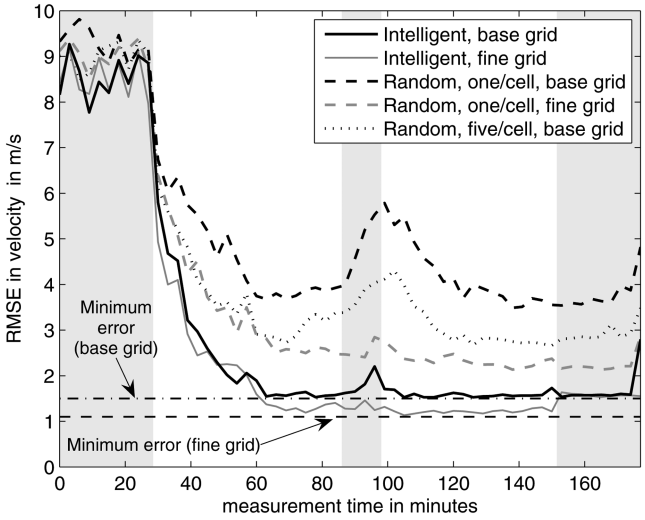


Fig. 10. Velocity error for CW data.

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

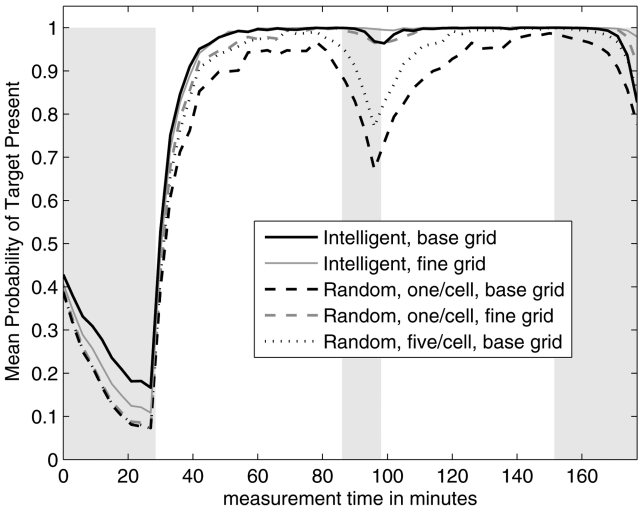


Fig. 11. Target probability for CW data.

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.

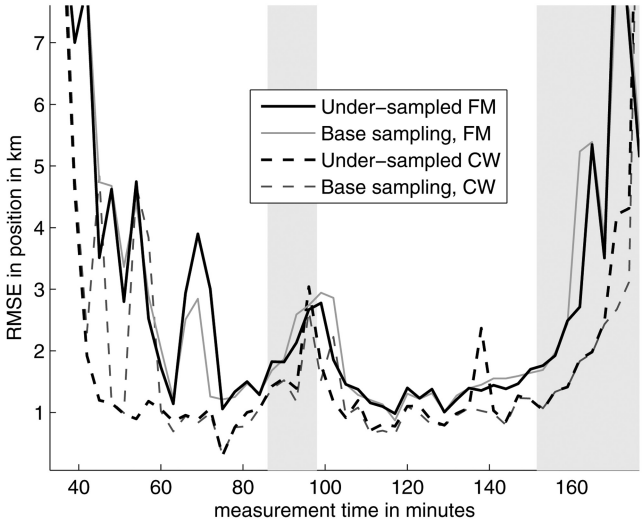


Fig. 12. Sensitivity analysis of RMSE for FM and CW.

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

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

while, for $I = K$, we have

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

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.

Algorithms for Asynchronous Track-to-Track Fusion

XIN TIAN
YAAKOV BAR-SHALOM

Most track-to-track fusion (T2TF) algorithms for distributed tracking systems in the literature assume that the local trackers are synchronized. However, in the real world, synchronization cannot be usually achieved among distributed local trackers where local measurements are obtained and local tracks are updated at different times with different rates. In addition, communication links between local trackers and the fusion center (FC) are subject to possible delays, which results in delayed local tracks for the fusion at the FC. This paper presents and compares algorithms for asynchronous Track-to-Track Fusion (AT2TF) for the fusion of asynchronous and delayed tracks. First, the optimal algorithm for AT2TF with no memory and with partial information feedback (AT2TFwoMpf) is presented. The algorithm, denoted as AT2TFwoMpfOpt, serves as the baseline algorithm for performance comparison. Three approximate AT2TF algorithms from the literature are compared with AT2TFwoMpfOpt and are shown to have consistency problems and loss in fusion accuracy. Then the Information Matrix Fusion (IMF) algorithm from the literature is generalized for the fusion of asynchronous tracks. Based on the generalized IMF (GIMF), AT2TF algorithms are derived for the information configurations with both partial and full information feedback. These algorithms are shown to have good consistency and nearly optimal fusion accuracy. Due to the simplicity of their implementation, these algorithms are appealing candidates for practical applications.

Manuscript received January 1, 2010; released for publication July 7, 2010.

Refereeing of this contribution was handled by Stefano Coraluppi.

Authors' address: Dept. of Electrical and Computer Engineering, University of Connecticut, Storrs CT 06269-2157, E-mail: {xin.tian, ybs}@engr.uconn.edu.

1557-6418/10/$17.00

1. INTRODUCTION

Algorithms for synchronous track-to-track fusion (T2TF) have been widely studied. For the optimal T2TF, it is critical to take into account the crosscovariances between tracks of the same target due to (i) the common process noise, and (ii) information feedback [2]. The optimal memoryless (without memory—"woM") T2TF with no information feedback (T2TFwoMnf) was studied in [3], [11]. In [15], the complete set of information configurations and the optimal algorithms for the synchronous T2TF were presented. These are T2TF without memory (T2TFwoM) with no, partial and full information feedback (designated as T2TFwoMnf, T2TFwoMpf and T2TFwoMff, respectively), as well as T2TF with memory (T2TFwM) with no, partial and full information feedback (designated as T2TFwMnf, T2TFwMpf and T2TFwff). The information matrix fusion (IMF) [14, 8] is a special case of T2TFwM. The advantage of IMF over the optimal T2TF is that it does not require the crosscovariances between the local tracks, which greatly simplifies its implementation. However, IMF is optimal only when the fuser operates at full rate [8, 7]. For reduced rate, IMF is heuristic. As reported in [6], IMF has consistency problems for extremely large process noise levels; however for most tracking scenarios it is consistent and has good tracking accuracy.

In the real world, synchronization cannot be usually achieved among distributed local trackers where local measurements are obtained and local tracks are updated at different time instants. In addition, the communication between local trackers and the fusion center (FC) is subject to possible delays, and thus the fusion of delayed tracks should also be addressed. In [5] the problem of the fusion of delayed tracks is converted and solved as the fusion of out-of-sequence measurements (OOSM). However, the algorithm deals only with the fusion of delayed synchronous tracks at full rate. A pseudo measurement approach of fusing asynchronous tracks can be found in [12]. In [13], three approximate algorithms for AT2TF were proposed. Later in the present research, they are evaluated by simulations and shown to have consistency problems. This is because the crosscorrelation between the central and local tracks due to information feedback is not accounted.

In this paper, first, the optimal (under linear Gaussian—LG—assumption) synchronous T2TF algorithm is generalized for the asynchronous situation, where the information configuration of memoryless fusion with partial information feedback (feedback only to the central track) [15] is used. The resulting algorithm accounts exactly for the crosscovariances between the central and local tracks. It handles both the asynchronous sampling times of the local trackers and the fusion of delayed tracks, and guarantees the consistency of the fused estimates. The optimal algorithms for the more complicated AT2TF with memory is not considered here, due to the limited gain in tracking accuracy, especially when significant geometric diversity exists among the local

tracks.[1] The drawback of the optimal AT2TF algorithm is its high communication cost and high complexity, which is very difficult to use for scenarios with more than two trackers and AT2TF with full information feedback.

Then the Information Matrix Fusion (IMF) is generalized for the fusion of asynchronous tracks. The algorithm for AT2TF with partial information feedback (AT2TFpf) based on the generalized IMF (GIMF) approach, denoted as AT2TFpfIMF, is presented and compared with AT2TFwoMpfOpt. It is shown that AT2TFpfIMF, although heuristic, is consistent and has a similar level of fusion accuracy as AT2TFwoMpfOpt. Due to the simplicity of AT2TFpfIMF compared to AT2TFwoMpfOpt, it is an appealing candidate for practical applications. Finally, the use of the GIMF approach for AT2TF with full information feedback (AT2TFff) is presented. The resulting fusion algorithm AT2TFffIMF is shown to yield consistent and accurate fusion results. Its variations with further savings in communication and little loss in fusion consistency and accuracy are also investigated.

The paper is organized as follows. Section 2 formulates the AT2TF problem. Section 3 presents the optimal fusion algorithm, AT2TFwoMpfOpt. Section 4 compares AT2TFwoMpfOpt with three approximate algorithms from [13]. Section 5 presents and evaluates the GIMF based algorithms for AT2TF with partial and full information feedback. The paper is summarized in Section 6.

2. PROBLEM FORMULATION

For the sake of simplicity, the basic scenario of the fusion of two tracks of a target from two local trackers is considered.[2] The trackers operate asynchronously with sampling intervals T_1 and T_2. Tracker 1 is collocated with the FC, whose track is available for fusion with no time delay. Tracker 2 is a remote tracker which sends its track $(\hat{x}_2(t_c \mid t_c), P_2(t_c \mid t_c))$ to the FC once in a while, where the communication time t_c is the time stamp of the local track. The track arrives at the FC with a random communication delay T_D. When track 2 is received, the FC fuses track 1 with the delayed track 2 at fusion time t_f (with $t_f \geq t_c + T_D$), which can be written as

$$[\hat{x}_c(t_f \mid t_f), P_c(t_f \mid t_f)] = \mathbf{f}[\hat{x}_1(t_f \mid t_f), P_1(t_f \mid t_f), \hat{x}_2(t_c \mid t_c), P_2(t_c \mid t_c), \ldots] \tag{1}$$

where $(\hat{x}_c(t_f \mid t_f), P_c(t_f \mid t_f))$ is the fused track. Depending on the fusion algorithm, additional information will be required, which is indicated by the "…" in (1). For the configuration with partial information feedback, after the fusion, track 1 continues with the fused track $(\hat{x}_c(k \mid k), P_c(k \mid k))$ which has improved accuracy, while local tracker 2 operates by itself unaffected by the fusion. For AT2TF with full information feedback, the fused track is sent back to tracker 2. When the feedback arrives, tracker 2 will fuse it with its local information and continue with the fused result. (See Section 5.3 for the details.)

3. THE OPTIMAL ALGORITHM FOR AT2TFPF WITHOUT MEMORY AND WITH PARTIAL INFORMATION FEEDBACK—AT2TFWOMPFOPT

The optimal algorithm for asynchronous T2TF (AT2TF) is presented only for the configuration of fusion with no memory and with partial information feedback, in view of the following facts:

- T2TF with no memory has some performance loss compared to the T2TF with memory; however, this is not significant especially when there is significant geometric diversity among the local trackers.
- For AT2TF in the presence of communication delays, the exact algorithm for the configuration with full information feedback is impractical.

The objective of the AT2TF is to fused track 1, given by $\hat{x}_1(t_f \mid t_f), P_1(t_f \mid t_f)$, with the predicted track 2, given by $\hat{x}_2(t_f \mid t_c), P_2(t_f \mid t_c)$. Compared to the synchronous T2TF [15], there are two additional issues to address. The first one is that the sampling (measurement) times of the two trackers are different. This causes difficulties for the calculation of the crosscovariances between tracks. The solution to this problem is to use the union of the sampling times, where zero filter gains are used for the tracks at sampling times when there is no actual measurement available for update. Then the crosscovariance between tracks can be calculated as in the synchronized case using (2) below.

Fig. 1 illustrates the idea of the union of the sampling times. Fig. 1(a) shows the time axis of tracker 1, on which the black circles indicate when tracker 1 received measurements and did actual track updates.[3] Fig. 1(b) shows the same for track 2. Fig. 1(c) shows the union of the sampling times of the two trackers on the same time axis. Then tracks 1 and 2 are discretized according to the union of the sampling times in Fig. 1(d)–(e), where the black circles represent actual track updates and the white circles represent virtual track updates, i.e., with zero filter gains.

To differentiate the original tracks and the discretized tracks according to the union of the sampling times, the latter are denoted as $\hat{x}_{i^*}$ with "*" superscript for the track index. The exact crosscovariance between the two tracks at the any time $t_a > t_l$ is calculated as

[1] In such cases, track estimates from the local trackers provide complementary perspectives of the target state.

[2] The problem of track-to-track association is not considered.

[3] It is assumed that the local trackers have no delay between when a measurement is taken and the track update. Delay is assumed in the communication between tracker 2 and the FC.

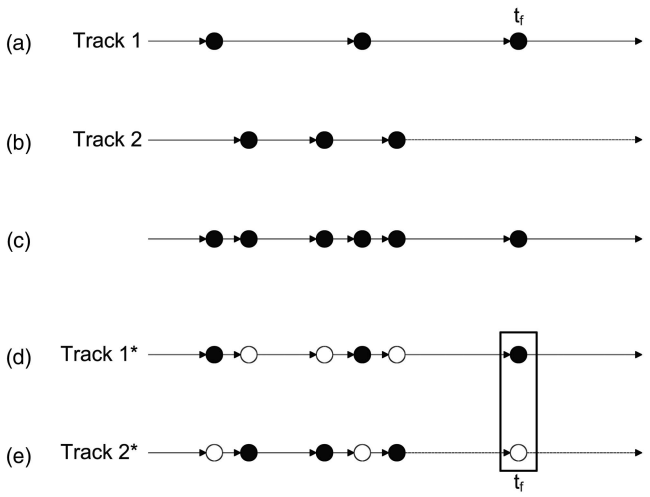


Fig. 1. The union of the sampling times.

follows

$$P_{1^*2^*}(t_a \mid t_a) = W^e_{1^*}(t_a,t_l)P_{1^*2^*}(t_l \mid t_l)W^e_{2^*}(t_a,t_l)' + \sum_{i=l+1}^{a} W^v_{1^*}(t_a,t_{i-1})Q(t_i,t_{i-1})W^v_{2^*}(t_a,t_{i-1})' \tag{2}$$

where t_l, designated as the "prior time," is the most recent time at which the crosscovariance between the two tracks is available; the summation in (2) is over the set $\{t_l,\ldots,t_a\}$, which is the union of the sampling times in the time interval $[t_l,t_a]$; and

$$W^e_{s^*}(t_a,t_l) = \prod_{i=0}^{a-l-1} [I - K_{s^*}(t_{a-i})H_{s^*}(t_{a-i})]F(t_{a-i},t_{a-i-1}) \tag{3}$$

$$W^v_{s^*}(t_a,t_{i-1}) = -\left\{ \prod_{j=0}^{a-i-1} [I - K_{s^*}(t_{a-j})H_{s^*}(t_{a-j})]F(t_{a-j},t_{a-j-1}) \right\} \cdot [I - K_{s^*}(t_i)H_{s^*}(t_i)], \qquad s = 1,2. \tag{4}$$

where $K_{s^*}(t_i)$, $i = l+1,\ldots,a$ are the local Kalman filter gains, which are zero for the virtual updates; $H_{s^*}(t_i)$ are the observation matrices at local tracker s and $F(t_i,t_{i-1})$ are the state transition matrices from t_{i-1} to t_i. Note that the calculation of the exact crosscovariance between two tracks requires the local filter gains and observation matrices at every sampling time, which puts a high requirement on communication capacity. An approximate approach to save communication cost can be found in [15].

Note that, for the synchronous T2TF, the system can use either the Discretized Continuous-time Kinematic Model or the Direct Discrete-Time Kinematic Model (see [1] Secs. 6.2 and 6.3). However, for AT2TF, the use of the union the sampling times requires to break the local process noises down to finer pieces (shorter time intervals). To preserve the process noise whiteness after the finer discretization, only the Discretized Continuous-Time Kinematic Model should be used.

The second issue is that the fusion of local estimates with time delays makes it more difficult to calculate the crosscovariance between the local tracks. The flowchart in Fig. 2 for the crosscovariance calculation and track-to-track fusion should be carefully followed. Starting from the first fusion, as shown in Fig. 2(a), $t_{l(1)}$ denotes the prior time of the 1st fusion when the most up-to-date covariances and crosscovariances between the two tracks, i.e., $P_1(t_{l(1)} \mid t_{l(1)})$, $P_2(t_{l(1)} \mid t_{l(1)})$ and $P_{12}(t_{l(1)} \mid t_{l(1)})$, are available at the FC. The communication time $t_{c(1)}$ is the time when track 2, namely, $(\hat{x}_2(t_{c(1)} \mid t_{c(1)}), P_2(t_{c(1)} \mid t_{c(1)}))$, is sent to the FC for the first fusion. Due to the time delay in data transmission, at fusion time $t_{f(1)}$, track $(\hat{x}_1(t_{f(1)} \mid t_{f(1)}),P_1(t_{f(1)} \mid t_{f(1)}))$ will be fused with the predicted track $(\hat{x}_2(t_{f(1)} \mid t_{c(1)}),P_2(t_{f(1)} \mid t_{c(1)}))$.

The first fusion is done as follows

- Discretize both track 1 and the delayed track 2 from $t_{l(1)}$ to $t_{f(1)}$ according to the union of the sampling times.
- Propagate the prior information from $t_{l(1)}$ to $t_{c(1)}$

$$P_{1*}(t_{c(1)} \mid t_{c(1)}) = \begin{cases} P_1(t_{c(1)} \mid t_{c(1)}), & \text{if an actual track 1 update occurred at } t_{c(1)} \\ F(t_{c(1)},t_{c1})P_1(t_{c1} \mid t_{c1})F(t_{c(1)},t_{c1})' + Q(t_{c(1)},t_{c1}), & \text{otherwise} \end{cases} \tag{5}$$

where t_{c1} is the latest time before $t_{c(1)}$ when track 1 was updated and $Q(t_{c(1)},t_{c1})$ is the cumulative effect of the process noise in the interval $(t_{c1},t_{c(1)}]$. For tracker 2

$$P_{2^*}(t_{c(1)} \mid t_{c(1)}) = P_2(t_{c(1)} \mid t_{c(1)}) \tag{6}$$

and the crosscovariance $P_{1^*2^*}(t_{c(1)} \mid t_{c(1)})$ is calculated using (2) from $t_{l(1)}$ to $t_{c(1)}$.

Note that, with $\hat{x}_2(t_{c(1)} \mid t_{c(1)})$ and $P_2(t_{c(1)} \mid t_{c(1)})$ sent to the FC, at this point the covariances and crosscovariances between the two tracks, namely, $P_{1^*}(t_{c(1)} \mid t_{c(1)})$, $P_{2^*}(t_{c(1)} \mid t_{c(1)})$ and $P_{1^*2^*}(t_{c(1)} \mid t_{c(1)})$, are available at $t_{c(1)}$, which makes $t_{c(1)}$ the new prior time $t_{l(2)}$ for the second fusion.
- Then predict the received track 2 from $t_{c(1)}$ to the fusion time $t_{f(1)}$

$$\hat{x}_{2^*}(t_{f(1)} \mid t_{f(1)}) = \hat{x}_2(t_{f(1)} \mid t_{c(1)}) = F(t_{f(1)},t_{c(1)})\hat{x}_2(t_{c(1)} \mid t_{c(1)}) \tag{7}$$

$$P_{2^*}(t_{f(1)} \mid t_{f(1)}) = P_2(t_{f(1)} \mid t_{c(1)}) = F(t_{f(1)},t_{c(1)})P_2(t_{c(1)} \mid t_{c(1)})F(t_{f(1)},t_{c(1)})' + Q(t_{f(1)},t_{c(1)}) \tag{8}$$

where $F(t_{f(1)},t_{c(1)})$ is the state transition matrix from time $t_{c(1)}$ to $t_{f(1)}$ and $Q(t_{f(1)},t_{c(1)})$ is the cumulative effect of the process noises in $[t_{c(1)},t_{f(1)}]$.

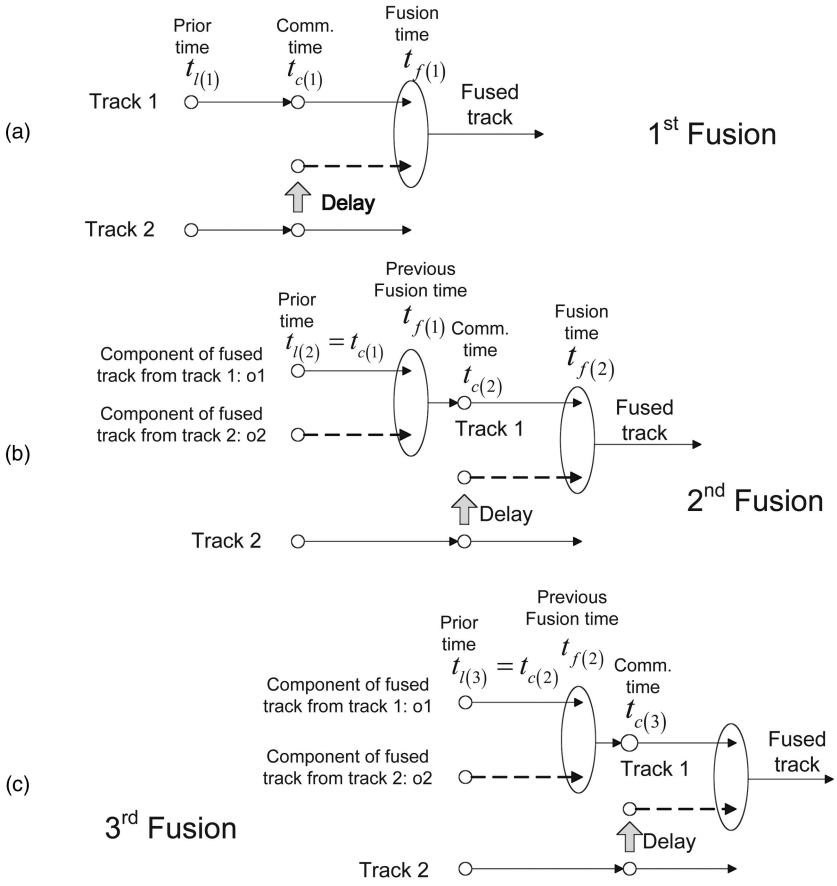


Fig. 2. Flowchart of T2TFwoMpf with feedback to tracker 1 and delayed track 2 ("- - - -" shows prediction).

- With (5)–(6), the crosscovariance $P_{1^*2^*}(t_{f(1)}f \mid t_{f(1)})$ is calculated using the union of the sampling times using (2) from $t_{c(1)}$ to $t_{f(1)}$.
- $\hat{x}_{1^*}(t_{f(1)} \mid t_{f(1)})$ and $P_{1^*}(t_{f(1)} \mid t_{f(1)})$ are available at the FC.
- With the information above, the optimal AT2TF is done using the LMMSE fuser [1]

$$\begin{aligned}\hat{x}_c(t_{f(1)} \mid t_{f(1)}) &= \hat{x}_{1^*}(t_{f(1)} \mid t_{f(1)}) + [P_{1^*}(t_{f(1)} \mid t_{f(1)}) - P_{1^*2^*}(t_{f(1)} \mid t_{f(1)})] \\ &\quad \cdot [P_{1^*}(t_{f(1)} \mid t_{f(1)}) + P_{2^*}(t_{f(1)} \mid t_{f(1)}) - P_{1^*2^*}(t_{f(1)} \mid t_{f(1)})' - P_{1^*1^*}(t_{f(1)} \mid t_{f(1)})]^{-1}[\hat{x}_{2^*}(t_{f(1)} \mid t_{f(1)}) - \hat{x}_{1^*}(t_{f(1)} \mid t_{f(1)})] \\ &= \hat{x}_{1^*}(t_{f(1)} \mid t_{f(1)}) + K_{1^*2^*}(t_{f(1)})[\hat{x}_{2^*}(t_{f(1)} \mid t_{f(1)}) - \hat{x}_{1^*}(t_{f(1)} \mid t_{f(1)})] \\ &= (I - K_{1^*2^*}(t_{f(1)}))\hat{x}_{1^*}(t_{f(1)} \mid t_{f(1)}) + K_{1^*2^*}(t_{f(1)})\hat{x}_{2^*}(t_{f(1)} \mid t_{f(1)}) \end{aligned} \tag{9}$$

$$\begin{aligned}P_c(t_{f(1)} \mid t_{f(1)}) &= P_{1^*}(t_{f(1)} \mid t_{f(1)}) - [P_{1^*}(t_{f(1)} \mid t_{f(1)}) - P_{1^*2^*}(t_{f(1)} \mid t_{f(1)})] \\ &\quad \cdot [P_{1^*}(t_{f(1)} \mid t_{f(1)}) + P_{2^*}(t_{f(1)} \mid t_{f(1)}) - P_{1^*2^*}(t_{f(1)} \mid t_{f(1)}) - P_{1^*2^*}(t_{f(1)} \mid t_{f(1)})']^{-1}[P_{1^*}(t_{f(1)} \mid t_{f(1)}) - P_{1^*2^*}(t_{f(1)} \mid t_{f(1)})'].\end{aligned} \tag{10}$$

The second fusion, as illustrated in Fig. 2(b),[4] is slightly different from the first fusion in propagating the crosscovariance between track 1 and the delayed track from the prior time $t_{l(2)}$ to the new communication time $t_{c(2)}$, which needs to take into account the impact of the previous fusion. This time (2) can not be used directly, since, after the first fusion, track 1 continued with the fused track (due to the partial information feedback), which contains two parts: one from the old track 1 (indicated by index "o1"), the other from the predicted track 2 (indicated by index "o2"). The crosscovariances $P_{1^*2^*}(t_{c(2)} \mid t_{c(2)})$ in the second fusion is calculated as follows:

- Calculate the crosscovariances $P_{\mathrm{o}1^*2^*}(t_{f(1)} \mid t_{f(1)})$ and $P_{\mathrm{o}2^*2^*}(t_{f(1)} \mid t_{f(1)})$ using (2) from $t_{c(1)}$ to $t_{f(1)}$.

[4]Note that, in Fig. 2, it is assumed that the second communication happens after the previous fusion. For scenarios where this assumption does not hold, the scheme can be easily modified to accommodate the change.

- Then, by evaluating the crosscovariance between the fused track estimate (9) and local track 2 at $t_{f(1)}$, one has

$$P_{1^*2^*}(t_{f(1)} \mid t_{f(1)}) = (I - K_{1^*2^*}(t_{f(1)}))P_{o1^*2^*}(t_{f(1)} \mid t_{f(1)}) + K_{1^*2^*}(t_{f(1)})P_{o2^*2^*}(t_{f(1)} \mid t_{f(1)}). \tag{11}$$

- Propagate the crosscovariance $P_{1^*2^*}(t_{f(1)} \mid t_{f(1)})$ from $t_{f(1)}$ to $t_{c(2)}$ using (2). Now, with the new $P_{1^*2^*}(t_{c(2)} \mid t_{c(2)})$ calculated, $t_{c(2)}$ becomes the new prior time for the next fusion, namely $t_{l(3)} = t_{c(2)}$, and the old prior information can be discarded.

The rest of the second fusion can be done exactly the same as in the first fusion. The third fusion and the ones afterwards are done as the second fusion.

4. AT2TFWOMPFOPT VS. THREE APPROXIMATE ALGORITHMS FROM THE LITERATURE

Three approximate algorithms for AT2TF from [13], denoted as AT2TFpfApprC, AT2TFpfApprB and AT2TFpfApprA, are compared with AT2TFwoMpfOpt proposed in Section 3. AT2TFpfApprC is the simplest one which assumes the errors of the local tracks are independent. AT2TFpfApprB and AT2TFpfApprA consider the crosscovariance between local tracks due to the common process noises. However, neither takes into account the crosscovariance due to the partial information feedback from FC to tracker 1.

A 2-D tracking scenario with two local trackers 1 and 2 tracking one target is used. The target motion follows a CWNA model[5] in [1] with process noise power spectral density (PSD) $\tilde{q}$. The target state is defined as $x = [\xi\ \dot{\xi}\ \zeta\ \dot{\zeta}]'$, i.e., position and velocity in 2-D Cartesian coordinates, with initial value set, without loss of generality, as [2000 m, −2 m/s, 5000 m, −5 m/s]. Tracker 1 is collocated with the FC at the origin (0,0), while tracker 2 is located at (X_2, Y_2). Tracker i ($i = 1,2$) takes position measurements of the target in its polar coordinates every T_i with zero mean white noise errors. The range standard deviation for both trackers is $\sigma_i^r = 10$ m and the azimuth standard deviation is $\sigma_i^a = 1°$. The local tracks are generated using the Converted Measurement Kalman Filter [1]. Tracker 2 sends its track at prespecified time instants to the FC with a communication delay of T_D. The simulation results are obtained from 100 Monte Carlo runs.

Scenario 1: Fusion of tracks with high process noise intensity and significant geometric diversity. Tracker 2 location: (10000,0) m; Sampling intervals: $T_1 = 2$ s, $T_2 = 2.5$ s; Process noise PSD: $\tilde{q} = 1$ m^2/s^3 (maneuvering index 0.03–0.3); Comm.; delay: $T_D = 2$ s; Fusion times: [9 : 6 : 147] s.

[5]As explained in Section 3, only the discretized continuous-time kinematic models can be used for AT2TFpfwoMopt.

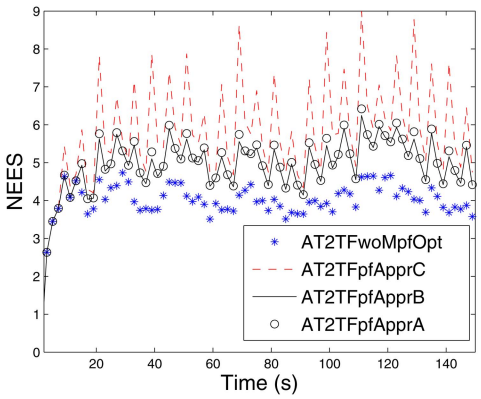


Fig. 3. AT2TFwoMpfOpt vs. three approximate AT2TF algorithms: Consistency test (Scenario 1: high process noise).

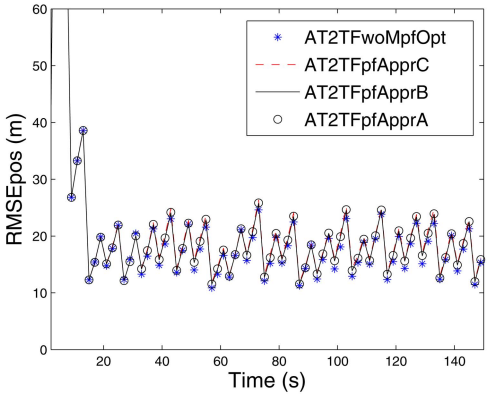


Fig. 4. AT2TFwoMpfOpt vs. three approximate AT2TF algorithms: RMS position errors (Scenario 1: high process noise).

As shown in Figs. 3–4, the optimal fusion algorithm AT2TFwoMpfOpt is consistent by checking the Normalized Estimation Error Squared (NEES), and it has small tracking errors especially at the fusion times. Algorithms AT2TFpfApprA–C, however, have consistency problems, because the crosscorrelation due to the partial information feedback is not accounted for. When the process noise level is high, the impact of this on the RMSE is less significant since the track estimates have short memories.

Scenario 2: Fusion of tracks with low process noise intensity and significant geometric diversity. Tracker 2 location: (10000,0) m; Sampling intervals: $T_1 = 2$ s, $T_2 = 2.5$ s; Low process noise PSD: $\tilde{q} = 0.01$ m^2/s^3 (maneuvering index 0.003–0.03); Comm. delay: $T_D = 2$ s; Fusion times: [9 : 6 : 147] s.

As shown in Figs. 5–6, when the process noise level is low, the RMSE performance of the three approximate algorithms becomes worse and their consistency is much worse.

COMMENTS ON AT2TFWOMPFOPT

- The optimal algorithm for AT2TF needs to take into account the crosscorrelation between the local tracks due to the common process noise and, especially, the information feedback.

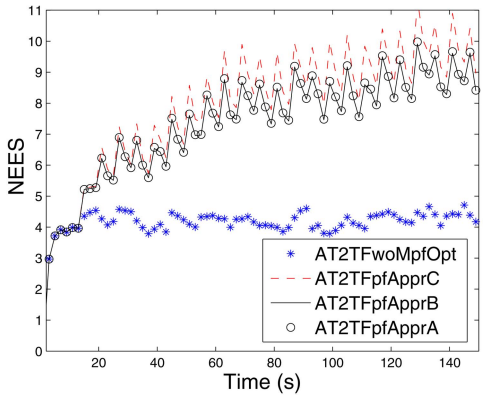


Fig. 5. AT2TFwoMpfOpt vs. three approximate AT2TF algorithms: Consistency test (Scenario 2: low process noise).

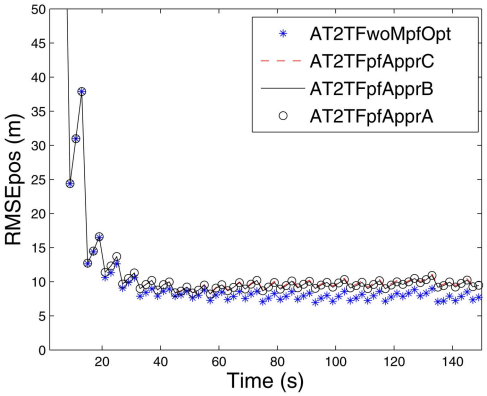


Fig. 6. AT2TFwoMpfOpt vs. three approximate AT2TF algorithms: RMS position errors (Scenario 2: low process noise).

- The drawbacks of the optimal fusion algorithm are its high communication cost and complexity. Approximate algorithms that lower the cost of AT2TFwoMpfOpt can be found in [16].

To avoid the calculation of the crosscovariance, a generalized Information Matrix Fusion (GIMF) [9, 6, 14] can be used for AT2TF. This is discussed in Section 5.

5. THE GENERALIZED INFORMATION MATRIX FUSION FOR AT2TF

The Information Matrix Fusion (IMF) [14, 8] is optimal (equivalent to the CMF) only at full rate. At a reduced rate, the algorithm is heuristic, but it works remarkably well over the practical range of process noise levels [7]. In this section, the generalized form of the IMF is presented for asynchronous T2TF. Then, based on the generalized IMF (GIMF), algorithms for AT2TF with partial and full information feedback are presented and evaluated.

5.1. The Generalized Form of the Information Matrix Fusion

Consider the fusion of track 1 at the FC and a delayed local track from tracker 2. Suppose one has

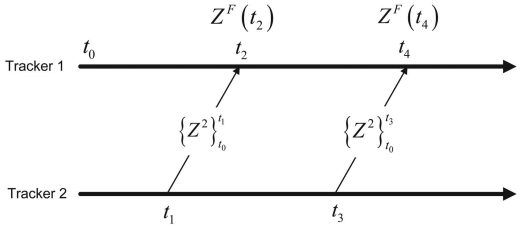


Fig. 7. Information flow of AT2TFpfIMF.

- track $(\hat{x}^1(t_f \mid t_f), P^1(t_f \mid t_f))$, from tracker 1 (same as FC)
- tracks $(\hat{x}^2(t_1 \mid t_1), P^2(t_1 \mid t_1))$ and $(\hat{x}^2(t_2 \mid t_2), P^2(t_2 \mid t_2))$ from tracker 2, $t_1 < t_2 \le t_f$.

All the above are from the same target. The fused track at t_f according to the Generalized Information Matrix fusion (GIMF) is given by

$$P(t_f)^{-1} = P^1(t_f \mid t_f)^{-1} + [P^2(t_f \mid t_2)^{-1} - P^2(t_f \mid t_1)^{-1}] \tag{12}$$

$$P(t_f)^{-1}\hat{x}(t_f) = P^1(t_f \mid t_f)^{-1}\hat{x}^1(t_f \mid t_f) + [P^2(t_f \mid t_2)^{-1}\hat{x}^2(t_f \mid t_2) - P^2(t_f \mid t_1)^{-1}\hat{x}^2(t_f \mid t_1)] \tag{13}$$

which contains the information from $(\hat{x}^1(t_f \mid t_f), P^1(t_f \mid t_f))$ and the information gain $\{Z^2\}_{t_1}^{t_2}$ from track 2 which is due to the local measurements during $t_1 < t \le t_2$ and quantified by the expression in the brackets in (12). While the GIMF defined by (12)–(13) is not optimal,[6] these equations will be used to obtain several near-optimal practical fusers in the sequel.

5.2. AT2TF with Partial Information Feedback Using GIMF—AT2TFpfIMF

This subsection presents the algorithm for AT2TF with partial information feedback using IMF (AT2TFpfIMF) and compares it with the exact algorithm AT2TFwoMpfOpt in Section 3.

Fig. 7 shows the information flow of AT2TFpfIMF. Suppose at time t_1, for the first time, tracker 2 sent its track $(\hat{x}^2(t_1 \mid t_1), P^2(t_1 \mid t_1))$ to the Fusion Center (FC), which represents the information set $\{Z^2\}_{t_0}^{t_1}$. The track arrived at the FC at time t_2 and was fused with track 1 using the GIMF from Section 5.1 as

$$P^F(t_2)^{-1} = P^1(t_2 \mid t_2)^{-1} + [P^2(t_2 \mid t_1)^{-1} - \mathbf{0}] \tag{14}$$

$$P^F(t_2)^{-1}\hat{x}^F(t_2) = P^1(t_2 \mid t_2)^{-1}\hat{x}^1(t_2 \mid t_2) + [P^2(t_2 \mid t_1)^{-1}\hat{x}^2(t_2 \mid t_1) - \mathbf{0}]. \tag{15}$$

Note that at t_0 we assume zero initial information about the target state $(P^2(t_0 \mid t_0)^{-1} = \mathbf{0})$ which accounts for the zero terms above. The fused track contains the

[6] As indicated before the IMF optimality requires the fusion to be performed at every time any of the local tracks are updated [2].

information set

$$Z^F(t_2) = \{Z^1\}_{t_0}^{t_2} + \{Z^2\}_{t_0}^{t_1} \tag{16}$$

where $\{Z^1\}_{t_0}^{t_2}$ is from track 1 and $\{Z^2\}_{t_0}^{t_1}$ is from the delayed track 2. After the fusion, according to the partial information feedback, tracker 1 continues with the fused track, namely,

$$\hat{x}^1(t_2^+ \mid t_2^+) = \hat{x}^F(t_2) \tag{17}$$

$$P^1(t_2^+ \mid t_2^+) = P^F(t_2) \tag{18}$$

where t_2^+ means at t_2 after the fusion.

For the next fusion, local tracker 2 sends its track $(\hat{x}^2(t_3 \mid t_3), P^2(t_3 \mid t_3))$ to the FC at t_3; the fusion at t_4 is done using the GIMF approach as follows

$$\begin{aligned} &P^F(t_4)^{-1} \\ &\quad = P^1(t_4 \mid t_4)^{-1} + [P^2(t_4 \mid t_3)^{-1} - P^2(t_4 \mid t_1)^{-1}] \end{aligned} \tag{19}$$

$$\begin{aligned} &P^F(t_4)^{-1}\hat{x}^F(t_4) \\ &\quad = P^1(t_4 \mid t_4)^{-1}\hat{x}^1(t_4 \mid t_4) \\ &\qquad + [P^2(t_4 \mid t_3)^{-1}\hat{x}^2(t_4 \mid t_3) - P^2(t_4 \mid t_1)^{-1}\hat{x}^2(t_4 \mid t_1)]. \end{aligned} \tag{20}$$

The fused track contains the information set

$$Z^F(t_4) = \{Z^F(t_2) + \{Z^1\}_{t_2}^{t_4}\} + \{Z^2\}_{t_1}^{t_3} \tag{21}$$

where $\{Z^F(t_2) + \{Z^1\}_{t_2}^{t_4}\}$ is from track 1 at t_4 and $\{Z^2\}_{t_1}^{t_3}$ is from the information gain at tracker 2 from t_1 to t_3.

The subsequent fusions are done in the same fashion. The performance of AT2TFpfIMF is compared next with AT2TFwoMpfOpt in a tracking scenario similar to those used in Section 3.

Scenario 3: Tracker 2 location: (5000,0) m; Sampling intervals: $T_1 = 2$ s, $T_2 = 2.5$ s; Process noise PSD $\tilde{q} = 10^{-1}$ m^2/s^3 (maneuvering index 0.009–0.09); Comm. delay: $T_D = 7$ s; Fusion times: $[11 : 8 : 150]$ s.

As shown in Figs. 8–9, AT2TFpfIMF, although heuristic, is consistent and has small errors as AT2TFwoMpfOpt. Compared to tracker 1 operating by itself, the improvement in tracking accuracy from the information feedback is very significant, primarily because of the geometric diversity between the two trackers. At the fusion times, the performance gap between AT2TF and the CMF is caused by the communication delay.

Following reasons contribute to the applicability of GIMF in AT2TF:

- The information gain from track 2, $\{Z^2\}_{t_1}^{t_2}$, quantified by $[P^2(t_f \mid t_2)^{-1} - P^2(t_f \mid t_1)^{-1}]$ in (12), is due to the local measurements from $(t_1\ t_2]$ and can be viewed as approximately independent from the other tracks. This coincides with the idea of the tracklet fusion [10].
- The subtraction structure of the information gain $[P^2(t_f \mid t_2)^{-1} - P^2(t_f \mid t_1)^{-1}]$ provides a desirable feature that cancels (approximately) its crosscorrelation with other local tracks caused by the common process noises with the use of prediction.

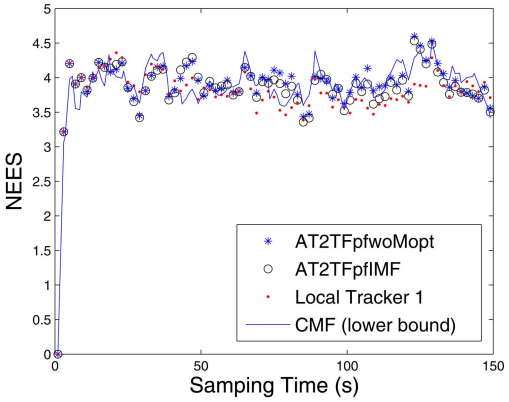


Fig. 8. AT2TFwoMpf vs. AT2TFpfIMF: Consistency test (Scenario 3).

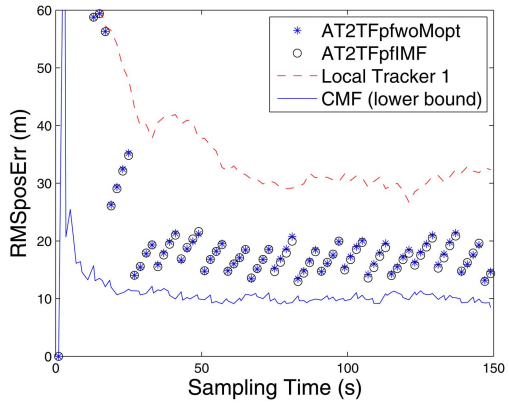


Fig. 9. AT2TFwoMpf vs. AT2TFpfIMF: RMS position errors (Scenario 3).

Thus GIMF for AT2TF has close to optimal fusion performance and is much simpler than the exact fusion of the local tracks. It is also applicable to the configuration of full information feedback, which will be discussed in Section 5.3.

5.3. AT2TF with Full Information Feedback Using GIMF—AT2TFfIMF

Due to the random communication delay in the asynchronous T2TF problem, it is too complicated to derive the optimal AT2TF algorithm with full information feedback. However, without the need of calculating the crosscovariance between the tracks, the GIMF approach allows full information feedback in AT2TF and the fusion algorithm can be used for an arbitrary number of local trackers.

Fig. 10 shows the information flow of AT2TF with full information feedback (AT2TFff) using the GIMF approach. The fusion at t_2 is the same as in Section 5.2 for AT2TFpf. The fused track $(\hat{x}^F(t_2), P^F(t_2))$ from (14)–(15) contains the information set

$$Z^F(t_2) = \{Z^1\}_{t_0}^{t_2} + \{Z^2\}_{t_0}^{t_1} \tag{22}$$

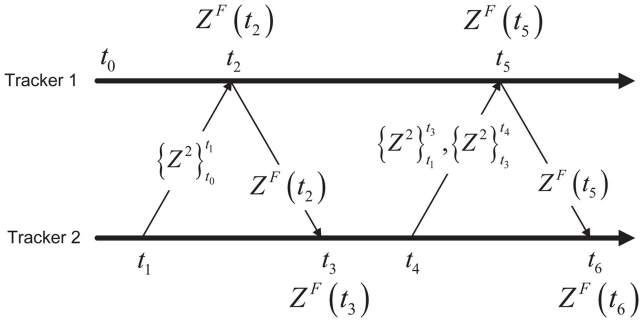


Fig. 10. Information flow of AT2TFffIMF.

where $\{Z^1\}_{t_0}^{t_2}$ is from track 1 and $\{Z^2\}_{t_0}^{t_1}$ is from the delayed track 2.

Then track 1 continues with the fused track which is also sent as feedback to local tracker 2. At time t_3 the feedback arrives at tracker 2, and is fused with the local information gain from time t_1 to t_3 using the GIMF approach as follows

$$P^F(t_3)^{-1} = [P^2(t_3 \mid t_3)^{-1} - P^2(t_3 \mid t_1)^{-1}] + P^F(t_3 \mid t_2)^{-1} \tag{23}$$

$$P^F(t_3)^{-1}\hat{x}^F(t_3) = [P^2(t_3 \mid t_3)^{-1}\hat{x}^2(t_3 \mid t_3) - P^2(t_3 \mid t_1)^{-1}\hat{x}^2(t_3 \mid t_1)] + P^F(t_3 \mid t_2)^{-1}\hat{x}^F(t_3 \mid t_2) \tag{24}$$

where track $(\hat{x}^F(t_3 \mid t_2), P^F(t_3 \mid t_2))$ is the prediction of track $(\hat{x}^F(t_2), P^F(t_2))$ from t_2 to t_3.

The information set of the fused track $(\hat{x}^F(t_3), P^F(t_3))$ is

$$Z^F(t_3) = \{Z^2\}_{t_1}^{t_3} + Z^F(t_2) \tag{25}.$$

Tracker 2 then continues with the fused track

$$\hat{x}^2(t_3^+ \mid t_3^+) = \hat{x}^F(t_3) \tag{26}$$

$$P^2(t_3^+ \mid t_3^+) = P^F(t_3) \tag{27}$$

where t_3^+ means at t_3 after the fusion.

At time t_4, tracker 2 sends the local information gain from $(t_1\ t_4]$ to the FC,[7] which contains the information sets $\{Z^2\}_{t_1}^{t_3}$ and $\{Z^2\}_{t_3}^{t_4}$. Note that the two information sets are separated at t_3 by the event of the fusion of the previous information feedback from the FC.

The information from $\{Z^2\}_{t_1}^{t_3}$ can be retrieved from the local the track pair $(\hat{x}^2(t_1 \mid t_1), P^2(t_1 \mid t_1))$ and $(\hat{x}^2(t_3 \mid t_3), P^2(t_3 \mid t_3))$, which were sent to the FC. Similarly, the information from $\{Z^2\}_{t_3}^{t_4}$ is retrieved using $(\hat{x}^2(t_3^+ \mid t_3^+), P^2(t_3^+ \mid t_3^+))$ and $(\hat{x}^2(t_4 \mid t_4), P^2(t_4 \mid t_4))$ which need to be sent to the FC.[8] When the local tracks arrive at the FC at t_5, they are fused with track 1 using the GIMF as follows

$$\begin{aligned} P^F(t_5)^{-1} &= P^1(t_5 \mid t_5)^{-1} + [P^2(t_5 \mid t_3)^{-1} - P^2(t_5 \mid t_1)^{-1}] \\ &\quad + [P^2(t_5 \mid t_4)^{-1} - P^2(t_5 \mid t_3^+)^{-1}] \end{aligned} \tag{28}$$

$$\begin{aligned} P^F(t_5)^{-1}\hat{x}^F(t_5) &= P^1(t_5 \mid t_5)^{-1}\hat{x}^1(t_5 \mid t_5) \\ &\quad + [P^2(t_5 \mid t_3)^{-1}\hat{x}^2(t_5 \mid t_3) - P^2(t_5 \mid t_1)^{-1}\hat{x}^2(t_5 \mid t_1)] \\ &\quad + [P^2(t_5 \mid t_4)^{-1}\hat{x}^2(t_5 \mid t_4) - P^2(t_5 \mid t_3^+)^{-1}\hat{x}^2(t_5 \mid t_3^+)]. \end{aligned} \tag{29}$$

The fused track at t_5 contains the information set

$$Z^F(t_5) = \{Z^F(t_2) + \{Z^1\}_{t_2}^{t_5}\} + \{Z^2\}_{t_1}^{t_3} + \{Z^2\}_{t_3}^{t_4} \tag{30}$$

where $\{Z^F(t_2) + \{Z^1\}_{t_2}^{t_5}\}$ is from track 1, namely $(\hat{x}^1(t_5 \mid t_5), P^1(t_5 \mid t_5))$ before the fusion.

Then tracker 1 continues with the fused track $(\hat{x}^F(t_5), P^F(t_5))$ which is also sent back to local tracker 2. At t_6, when the feedback arrives, the local information fusion is done similarly to the fusion at t_3. Thus

$$P^F(t_6)^{-1} = [P^2(t_6 \mid t_6)^{-1} - P^2(t_6 \mid t_4)^{-1}] + P^F(t_6 \mid t_5)^{-1} \tag{31}$$

$$P^F(t_6)^{-1}\hat{x}^F(t_6) = [P^2(t_6 \mid t_6)^{-1}\hat{x}^2(t_6 \mid t_6) - P^2(t_6 \mid t_4)^{-1}\hat{x}^2(t_6 \mid t_4)] + P^F(t_6 \mid t_5)^{-1}\hat{x}^F(t_6 \mid t_5) \tag{32}$$

where track $(\hat{x}^F(t_6 \mid t_5), P^F(t_6 \mid t_5))$ is the prediction of $(\hat{x}^F(t_5), P^F(t_5))$ from the feedback. The information set of the fused track $(\hat{x}^F(t_6), P^F(t_6))$ is

$$Z^F(t_6) = \{Z^2\}_{t_4}^{t_6} + Z^F(t_5). \tag{33}$$

The subsequent fusions repeat the procedure described above. The performance of AT2TFffIMF is demonstrated in the tracking scenario introduced in Section 3 with the parameters specified next.

Scenario 4: Tracker 2 location: $(5000, 0)$ m; Sampling intervals: $T_1 = 2$ s, $T_2 = 3.5$ s; Process noise PSD: $\tilde{q} = 10^{-1}$ m^2/s^3 (maneuvering index 0.009–0.09); Comm. delay: $T_D = 6$ s (both directions); Fusion times: $[5 : 17 : 150]$ s.

Figs. 11–12 show that the local tracks with information feedback are consistent and both achieve significantly improved tracking accuracy. At the fusion times, the performance gap in the RMS position errors between the fused track and the CMF is due to the communication delay.

5.4. AT2TFffIMF with Reduced Communication

As discussed in Section 5.3, at time t_4, two pairs of tracks $(\hat{x}^2(t_1 \mid t_1), P^2(t_1 \mid t_1))$, $(\hat{x}^2(t_3 \mid t_3), P^2(t_3 \mid t_3))$ and $(\hat{x}^2(t_3^+ \mid t_3^+), P^2(t_3^+ \mid t_3^+))$, $(\hat{x}^2(t_4 \mid t_4), P^2(t_4 \mid t_4))$ were

[7] Here it is assumed that $t_4 > t_3$, which means tracker 2 sent its track to the FC after it got the information feedback from the previous fusion. However, this assumption is not essential. The information flow can be easily modified to accommodate the other case.

[8] Note that this causes additional communication cost. Algorithms that reduces this communication cost will be discussed in Section 5.4.

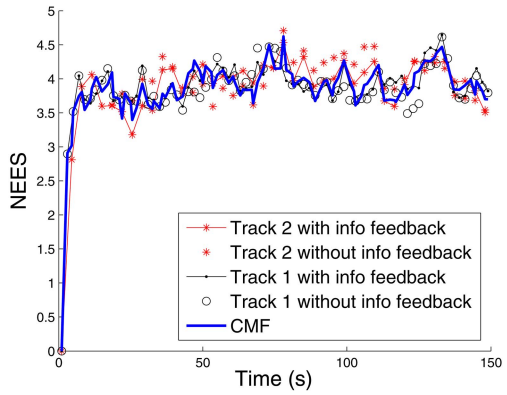


Fig. 11. AT2TFffIMF: Consistency test (Scenario 4: low process noise).

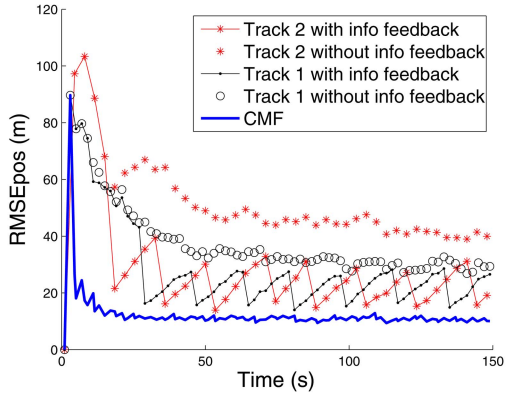


Fig. 12. AT2TFffIMF: RMS position errors (Scenario 4: low process noise).

needed to retrieve the local information gain from t_2 to t_4. To reduce the communication cost, one option is to fuse these two information gains into a single one before the transmission. Using the GIMF, one has

$$\begin{aligned} &P^2(t_4^+ \mid t_4^+)^{-1} \\ &\quad = [P^2(t_4 \mid t_3)^{-1} - P^2(t_4 \mid t_1)^{-1}] \\ &\qquad + [P^2(t_4 \mid t_4)^{-1} - P^2(t_4 \mid t_3^+)^{-1}] \end{aligned} \tag{34}$$

$$\begin{aligned} &P^2(t_4^+ \mid t_4^+)^{-1}\hat{x}^2(t_4^+ \mid t_4^+) \\ &\quad = [P^2(t_4 \mid t_3)^{-1}\hat{x}^2(t_4 \mid t_3) - P^2(t_4 \mid t_1)^{-1}\hat{x}^2(t_4^+ \mid t_1)] \\ &\qquad + [P^2(t_4 \mid t_4)^{-1}\hat{x}^2(t_4 \mid t_4) - P^2(t_4 \mid t_3^+)^{-1}\hat{x}^2(t_4 \mid t_3^+)]. \end{aligned} \tag{35}$$

Then the fused track $(P^2(t_4^+ \mid t_4^+)^{-1}\hat{x}^2(t_4^+ \mid t_4^+), P^2(t_4^+ \mid t_4^+)^{-1})$, which summarizes the information gain from $\{Z^2\}_{t_1}^{t_3}$ and $\{Z^2\}_{t_3}^{t_4}$, is sent to the FC.[9]

For AT2TF at t_5 at the FC, the straightforward way is to predict the track from t_4 to t_5, which yields $(\hat{x}^2(t_5 \mid t_4^+), P^2(t_5 \mid t_4^+))$ and fuse it with track $(\hat{x}^1(t_5 \mid t_5), P^1(t_5 \mid t_5))$ as if their errors were independent, i.e.,

$$P^F(t_5)^{-1} = P^1(t_5 \mid t_5)^{-1} + P^2(t_5 \mid t_4^+)^{-1} \tag{36}$$

$$\begin{aligned} P^F(t_5)^{-1}\hat{x}^F(t_5) &= P^1(t_5 \mid t_5)^{-1}\hat{x}^1(t_5 \mid t_5) \\ &\quad + P^2(t_5 \mid t_4^+)^{-1}\hat{x}^2(t_5 \mid t_4^+). \end{aligned} \tag{37}$$

However, this **direct prediction** approach, denoted as AT2TFffIMF$_{\text{DP}}$, ignores completely the crosscovariance between the predicted track and track 1 due to the common process noise.

A more sophisticated approach, denoted as AT2TFffIMF$_{\text{FBP}}$, uses **fusion before prediction**, where track $(P^2(t_4^+ \mid t_4^+)^{-1}\hat{x}^2(t_4^+ \mid t_4^+), P^2(t_4^+ \mid t_4^+)^{-1})$ is fused first with track 1 at t_4, which gives

$$\begin{aligned} &P^F(t_4 \mid t_4)^{-1} \\ &\quad = P^1(t_4 \mid t_4)^{-1} + P^2(t_4^+ \mid t_4^+)^{-1} \end{aligned} \tag{38}$$

$$\begin{aligned} &P^F(t_4 \mid t_4)^{-1}\hat{x}^F(t_4 \mid t_4) \\ &\quad = P^1(t_4 \mid t_4)^{-1}\hat{x}^1(t_4 \mid t_4) \\ &\qquad + P^2(t_4^+ \mid t_4^+)^{-1}\hat{x}^2(t_4^+ \mid t_4^+). \end{aligned} \tag{39}$$

Then the fusion at t_5 can be done using the GIMF as

$$\begin{aligned} &P^F(t_5)^{-1} \\ &\quad = P^1(t_5 \mid t_5)^{-1} + [P^F(t_5 \mid t_4)^{-1} - P^1(t_5 \mid t_4)^{-1}] \end{aligned} \tag{40}$$

$$\begin{aligned} &P^F(t_5)^{-1}\hat{x}^F(t_5) \\ &\quad = P^1(t_5 \mid t_5)^{-1}\hat{x}^1(t_5 \mid t_5) \\ &\qquad + [P^F(t_5 \mid t_4)^{-1}\hat{x}^F(t_5 \mid t_4) - P^1(t_5 \mid t_4)^{-1}\hat{x}^1(t_5 \mid t_4)] \end{aligned} \tag{41}$$

where $[P^F(t_5 \mid t_4)^{-1} - P^1(t_5 \mid t_4)^{-1}]$ is the information gain between predicted tracks $(\hat{x}^F(t_5 \mid t_4), P^F(t_5 \mid t_4))$ and $(\hat{x}^1(t_5 \mid t_4), P^1(t_5 \mid t_4))$ due to the fusion of $(P^2(t_4^+ \mid t_4^+)^{-1} \cdot \hat{x}^2(t_4^+ \mid t_4^+), P^2(t_4^+ \mid t_4^+)^{-1})$.

The performances of AT2TFffIMF$_{\text{DP}}$ and AT2TFffIMF$_{\text{FBP}}$ are compared by simulations in a tracking scenario similar to those used in Section 3 with the parameters specified next.

Scenario 5: Tracker 2 location: (5000, 0) m; Sampling intervals: $T_1 = 2$ s, $T_2 = 3.5$ s; Process noise PSD: $\tilde{q} = 1$ m^2/s^3 (maneuvering index 0.03–0.3); Comm. delay: $T_D = 6$ s (both directions); Fusion times: [5 : 17 : 150] s. As before, the simulation results are obtained from 100 MC runs.

Fig. 13 shows that AT2TFffIMF$_{\text{FBP}}$ has better consistency at the fusion times than AT2TFffIMF$_{\text{DP}}$. This is because AT2TFffIMF$_{\text{FBP}}$ by using the fusion before prediction approach has better crosscorrelation cancelation effect. For the tracking scenario considered, the moderate inconsistency of AT2TFffIMF$_{\text{DP}}$ causes little loss in fusion accuracy. The accuracies of both of these fusion algorithms are practically as good as AT2TFffIMF.

[9]Here the information form of the track is used, which is equivalent to the regular form as long as the covariance of the track is invertible.

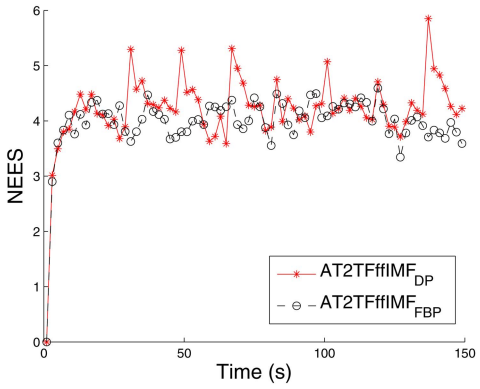


Fig. 13. AT2TFffIMF$_{DP}$ vs. AT2TFffIMF$_{FBP}$: Consistency test (Scenario 5: high process noise).

6. CONCLUSIONS

The optimal algorithm for Asynchronous Track-to-Track Fusion (AT2TF) was obtained for the information configuration of fusion with no memory and partial information feedback—AT2TFwoMpfOpt. It accounts exactly for the crosscorrelation between the two local tracks due to the common process noise and information feedback. The drawback of the exact AT2TF fusion algorithm is that it has high communication and computation cost, and is very difficult to use when there are more than two trackers or for the configuration with full information feedback.

An approximate algorithm (AT2TFpfIMF) for AT2TF with partial information feedback based on the GIMF was presented. It has low communication and computation cost and is shown to have good consistency and near optimal fusion accuracy. The use of the GIMF approach for AT2TF with full information feedback was also presented. The proposed algorithm (AT2TFffIMF) was shown to be consistent and have excellent fusion accuracy. Two variations of the algorithm, which have lower communication cost, were derived as well. Both have practically the same fusion accuracy as the original algorithm.

The proposed suboptimal AT2TF algorithms based on GIMF have low complexity and can be easily used for an arbitrary number of local trackers, which makes them appealing candidates for practical applications.

REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction.
New York: Wiley, 2001.

[2] Y. Bar-Shalom, P. K. Willet, and X. Tian
Handbook of Algorithms for Target Tracking and Data Fusion.
YBS publishing, 2011.

[3] Y. Bar-Shalom and L. Campo
The effect of the common process noise on the two-sensor fused-track covariance.
IEEE Transactions on Aerospace and Electronic Systems, **22**, 6 (Nov. 1986), 803–804.

[4] Y. Bar-Shalom
Update with out-of-sequence measurements in tracking.
IEEE Transacations on Aerospace and Electronic Systems, **38**, 3 (July 2002), 769–778.

[5] S. Challa, J. Legg, and X. Wang
Track-to-track fusion of out-of-sequence tracks.
In *Proceedings of the 5th International Conference on Information Fusion*, July 2002, 919–926.

[6] K. C. Chang, R. K. Saha, and Y. Bar-Shalom
On optimal track-to-track fusion.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 4 (Oct. 1997), 1271–1276.

[7] K. C. Chang, Z. Tian, and R. Saha
Performance evaluation of track fusion with information matrix filter.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 2 (Apr. 2002), 455–466.

[8] C. Y. Chong, S. Mori, and K. C. Chang
Distributed multitarget multisensor tracking.
Multitarget-Multisensor Tracking: Advanced Applications, edited by Y. Bar-Shalom, MA: Artech House, 1990, ch. 8. Reprinted by YBS publishing, 1998.

[9] C. Y. Chong
Hierarchical estimation.
In *Proceedings of MIT/ONR Workshop on C3*, 1979.

[10] O. E. Drummond, W. D. Blair, G. C. Brown, T. L. Ogle, Y. Bar-Shalom, R. L. Cooperman, and W. H. Barker
Performance assessment and comparison of various tracklet methods for maneuvering targets.
In *Proceedings of SPIE Conference on Signal Processing, Sensor Fusion, and Target Recognition XII*, vol. 5096, 2003.

[11] X. R. Li, Y. M. Zhu, J. Wang, and C. Z. Han
Unified optimal linear estimation fusion—partI: Unified model and fusion rules.
IEEE Transactions on Information Theory, **49**, 9 (Sept. 2003), 2192–2207.

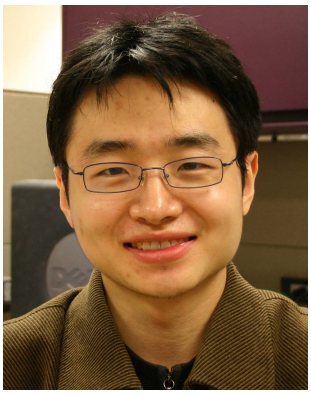
[12] M. Mallick, S. Schimdt, L. Y. Pao, and K. C. Chang
Out-of-sequence track filtering using the decorrelated pseudo measurement approach.
In *Proceedings of SPIE Conference on Signal and Data Processing for Small Targets*, Apr. 2004, 154–166.

[13] A. Novoselsky, S. E. Sklarz, and M. Dorfan
Track to track fusion using out-of-sequence track information.
In *Proceedings of the 10th International Conference on Information Fusion*, July 2007.

[14] J. L. Speyer
Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem.
IEEE Transactions on Automatic Control, **2**, 2 (Apr. 1979), 54–57.

[15] X. Tian and Y. Bar-Shalom
Track-to-Track Fusion Configurations and Association in a Sliding Window.
Journal of Advances in Information Fusion, **4**, 2 (Dec. 2009), 146–164.

[16] X. Tian and Y. Bar-Shalom
The optimal algorithm for asynchronous track-to-track fusion.
In *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, Apr. 2010, #7698-46.



Xin Tian received the B.S. degree in 2002 and M.S. degree in 2005, both from the Department of Information and Communication Engineering, Xi'an Jiaotong University, China. In 2010 he received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Connecticut, USA.

Dr. Tian's research areas include statistical signal processing, tracking and information fusion algorithms, detection theory, decision theory under uncertainty, and sensor management. He is currently a research scientist at I-fusion Inc., Germantown, MD.

Yaakov Bar-Shalom (S'63—M'66—SM'80—F'84) was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering.

From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, CA. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab.

His current research interests are in estimation theory and target tracking and has published over 370 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate text *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000).

He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas.

During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of Automatica. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–08.

In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. He is the recipient of the 2008 IEEE Dennis J. Picard Medal for Radar Technologies and Applications.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

2010 BOARD OF DIRECTORS*

2008–2010	2009–2011	2010–2012
Steve Andler	Éloi Bossé	Simon Maskell
Joachim Biermann	Uwe D. Hanebeck	Peter Willett
Mahendra Mallick	Roy Streit	Wolfgang Koch

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Elisa Shahbazian, 2009	W. Dale Blair, 2005	Pramod Varshney, 2001
Darko Musicki, 2008	Chee Chong, 2004	Yaakov Bar-Shalom, 2000
Erik Blasch, 2007	Xiao-Rong Li, 2003	Jim Llinas, 1999
Pierre Valin, 2006	Yaakov Bar-Shalom, 2002	Jim Llinas, 1998

SOCIETY VISION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied information fusion technologies.

SOCIETY MISSION

Advocate

To advance the profession of fusion technologies, propose approaches for solving real-world problems, recognize emerging technologies, and foster the transfer of information.

Serve

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

Communicate

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

Educate

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

Integrate

Integrate ideas from various approaches for information fusion, and look for common threads and themes–look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

Disseminate

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors of papers in one of the technical areas listed on the inside cover of JAIF are encouraged to submit their papers for peer review at http://jaif.msubmit.net.

Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at http://jaif.msubmit.net. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.

Journal of Advances in Information Fusion Volume 5 Number 2 December 2010 ISSN 1557-6418