General Direction-of-Arrival Tracking with Acoustic Nodes

Volkan Cevher, *Student Member, IEEE,* James H. McClellan, *Fellow, IEEE,*

Abstract

Traditionally in target tracking, much emphasis is put on the motion model that realistically represents the target's movements. In this paper, we first present the classical constant velocity model and then introduce a new model that incorporates an acceleration component along the heading direction of the target. We also show that the target motion parameters can be considered part of a more general feature set for target tracking. This is exemplified by showing that target frequencies, which may be unrelated to the target motion, can also be used to improve the tracking performance. In order to include the frequency variable, a new array steering vector is presented for the direction-of-arrival (DOA) estimation problems. The independent partition particle filter (IPPF) is used to compare the performances of the two motion models by tracking multiple maneuvering targets using the acoustic sensor outputs directly. The treatment is quite general since IPPF allows general type of noise models as opposed to Gaussianity imposed by Kalman type of formulations. It is shown that by incorporating the acceleration into the state vector, the tracking performance can be improved in certain cases as expected. Then, we demonstrate a case in which the frequency variable improves the tracking and classification performance for targets with close DOA tracks.

Index Terms

Motion dynamics, particle filter, Monté-Carlo simulation methods, reference priors, importance sampling, time-frequency analysis

I. Introduction

The direction-of-arrival (DOA) estimation problem has been extensively studied in the signal processing literature [1]. Narrow-band solutions based on beamforming such as multiple signal classification

V. Cevher and J.H. McClellan are with the Center for Signal and Image Processing, School of ECE, Georgia Institute of Technology, Atlanta GA 30332-0250.

Prepared through collaborative participation in the Advanced Sensors Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-02-0008.

(MUSIC) [2], minimum variance beamforming, and Pisarenko's method suffer degraded performance when the targets are moving relatively fast during the estimation batch (i.e., the snapshot period). The performance loss in these high resolution algorithms can be attributed to the nonstationarity of the array data caused by the rapid target motion. In order to remedy this problem, one can incorporate target motion dynamics to jointly estimate the DOA's while tracking targets [3]. These refined DOA estimates provide better performance in exchange for increased computational complexity.

In the case of wideband acoustic signals, the pioneering work by Wang and Kaveh [4] on coherent subspace processing coherently integrates the array autocorrelation matrices corresponding to the multiple frequencies of interest, so that signal-to-noise (SNR) and resolution gains can be achieved. The work by Gershman and Amin [5] approximates the signals within the DOA snapshot period as chirps and performs time-frequency MUSIC on the acoustic array outputs. The varying frequency approximation enables these wideband methods to tackle more realistic estimation problems. However, these algorithms, like the narrow-band techniques, produce snapshot DOA estimates; and hence, require heuristics for target association.

Advances in large scale integration of computer systems have made Monté-Carlo techniques a feasible alternative to solve the target tracking problem. Conventionally, given a target dynamics model, the underlying motion equations are simplified by linearization and Gaussian noise assumptions so that an analytical solution can be obtained. The extended Kalman filter is such a method; it is also the best minimum mean-squares linear estimator for the problem at hand. Monté-Carlo techniques, on the other hand, do not linearize or assume Gaussian noise; however, they approximate the posterior density of interest (e.g., a density that describes the likelihood of a target's position in space) by particles that represent a discrete version of the posterior. The idea is that if a sufficient number of *effective* particles can be used, the estimation performance will be close to the theoretically optimal solution.

A solution to the multiple target tracking problem has been given for the narrow-band case by Orton and Fitzgerald in [6] using an independent partition particle filter (IPPF). The independent partition assumption of the IPPF solves the DOA association problem common to the multiple target tracking algorithms. The implementation of the IPPF is shown in [6], given the target motion dynamics developed in [3] to track constant velocity targets with Brownian disturbance acting on the target heading directions. In this paper, the IPPF is again used, but along with a new target motion model that allows accelerations along the heading direction of the targets. It is shown with simulations that since the new motion model enables the IPPF to relax the constant velocity assumption, target tracking is improved when the targets have high accelerations as intuitively expected.

In target tracking, it is usually a good idea to retain certain target features even if they are not related to the motion parameters. This may, in turn, also help the data fusion problem with different type of sensors since the extra algorithmic computational load is usually much less taxing on the energy budget of the sensor than transmitting the raw data to a central processor. Hence, in addition to the motion parameters, we also show how to incorporate new features such as time-varying frequency signatures of the targets into the particle filter algorithm. It is assumed that a separate time-frequency tracker is tracking the dominant instantaneous frequencies of the targets; however, the issues related to the frequency estimation are not considered in this paper. It is demonstrated by simulations that in the case of multiple targets, the introduction of the frequency variable improves the tracking performance of the IPPF when two targets have similar motion parameters but different time-frequency signatures.

Observability is one of the main concerns in acoustic tracking problems because it must be possible to determine the target states from the array data uniquely. Observability of a time-varying system depends on both the observations and the state equation (i.e., the acoustic sensor outputs and the motion model) [7]. However, when the observations are sufficient to determine the target DOA's, the observability test reduces to a simple rank test on the Jacobian of the local representation of the motion model [3]. In this paper, it is assumed that the observations are sufficient to determine the target DOA's and the system observability is proved using the observability rank condition. We argue that with the same assumption on the observations, it is possible to automatically generate the IPPF code to do target tracking for some other target motion dynamics as long as the rank condition is satisfied. This can enable dynamic switching of motion models appropriate for different types of targets, resulting in a more flexible tracker.

The array model employed in the simulations has a special structure. We use a single node consisting of a circular omnidirectional microphone array that will supply DOA and frequency information about the targets. However, the derivations do not depend on the particular structure of the nodes. The reader should be cognizant of the fact that tracking, in this paper, implies the temporal estimation of the target DOA's as opposed to target positions. Spatial diversity of multiple nodes can be exploited in triangulating the targets of interest, which also presents a data fusion problem. However, the data fusion from multiple nodes can be facilitated by some of the variables in the target feature set such as the target orientation and frequency. It should be noted that these two variables are invariant from the node positions when the nodes have the same reference frame.[1] Hence, in a simple application such as triangularization, if the DOA information coming from each node also includes these features, the data association for multiple

[1]If the nodes have different reference directions, the orientation angles of targets can be found in another reference frame by simple additions and subtractions of the reference angles of the nodes.

targets can be done optimally with minimal information exchange.

The organization of the paper is as follows. Section II introduces the models that account for the target motion dynamics as well as the acoustic observations. Section III describes how to construct the necessary probability density functions as the backbone of the particle filter solution. The IPPF algorithm details are discussed in section IV before the simulation results, which are presented in section V.

II. Data Models

Consider K far-field targets coplanar with a sensor node consisting of P acoustic sensors. The sensor node (or sensor array) is not assumed to possess any special structure. Two motion models will be first presented, differing in the assumptions of constant velocity versus constant acceleration.

A. State Model-I

The targets are assumed to have constant speeds with a Brownian disturbance acting on their heading directions [3], [6]. With the introduction of the frequency variable, the model has the following state vector:

$$\mathbf{x}_k(t) \triangleq \begin{bmatrix} \theta_k(t) \\ Q_k(t) \\ \phi_k(t) \\ f_k(t) \end{bmatrix} \tag{1}$$

where $\theta_k(t)$, $\phi_k(t)$, and $f_k(t)$ are the DOA, the heading direction, and the instantaneous frequency of the k^{th} target. $Q_k(t) \triangleq \log q_k(t)$ is a compound variable where $q_k(t) \triangleq v_k/r_k(t)$. Target DOA's are measured clockwise with respect to the y-axis whereas the target heading directions are measured counter clockwise with respect to the x-axis. Figure 1 illustrates the geometry of the problem. The state update equation can be derived by relating the DOA's of the target at times t and $t+T$ using the geometrical relation of position 1 at $(r_k(t)\sin\theta_k(t), r_k(t)\cos\theta_k(t))$ to position 2 at $(r_k(t)\sin\theta_k(t) + v_kT\cos\phi_k(t), r_k(t)\cos\theta_k(t) + v_kT\sin\phi_k(t))$. Then, it is straightforward to obtain the following update relations:

$$\tan\theta_k(t+T) = \frac{r_k(t)\sin\theta_k(t) + v_kT\cos\phi_k(t)}{r_k(t)\cos\theta_k(t) + v_kT\sin\phi_k(t)} \tag{2}$$

and

$$r_k(t+T) = \sqrt{r_k^2(t) + 2r_k(t)v_kT\sin\left(\theta_k(t) + \phi_k(t)\right) + v_k^2T^2} \tag{3}$$

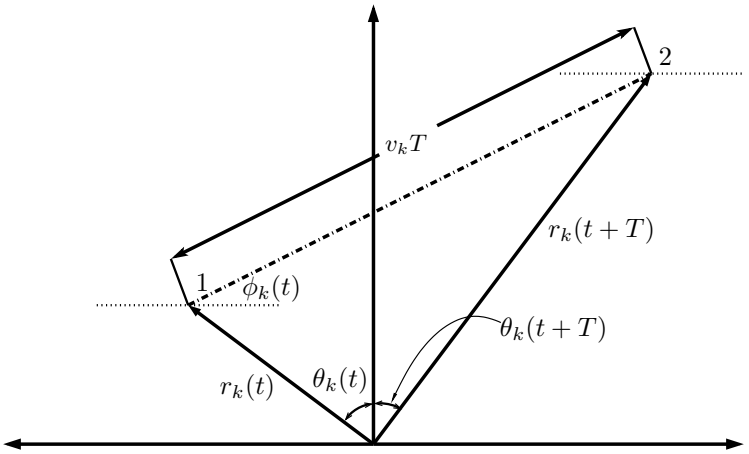


Fig. 1. The k^{th} target is at position 1 at time t and moves to position 2 in T seconds with a constant speed. The target is assumed to be in the far-field of the sensor array whose center coincides with the origin.

Equations (2) and (3) form a scalable system for the target motion dynamics at hand. To elaborate on this, consider scaling the range and the speed of the k^{th} target. It can be shown that this scaled target has the same set of update equations as above since the scale factor can be cancelled out. This, in fact, leads to the introduction of the compound variable $q_k(t)$ defined earlier. In the state update, however, the logarithm of $q_k(t)$ is used since an additive noise component can be employed (as opposed to the multiplicative noise when $q_k(t)$ is used). Note that the particle filter can cope with the general type noise and that additivity of the noise is not required; however, the logarithm is used to decouple the noise from the state update, which is given below. Also, note that the target maneuvers (i.e., the perpendicular accelerations) are modelled in the state update through the state noise on $\phi_k(t)$:

$$\begin{aligned}\mathbf{x}_k(t+T) &= \hat{\mathbf{f}}_{\mathbf{I}}(\mathbf{x}_k(t), \mathbf{u}(t+T)) = \mathbf{f}_{\mathbf{I}}(\mathbf{x}_k(t)) + \mathbf{u}(t+T) \\ &= \begin{bmatrix} \arctan\left\{\frac{\sin\theta_k(t)+e^{Q_k(t)}T\cos\phi_k(t)}{\cos\theta_k(t)+e^{Q_k(t)}T\sin\phi_k(t)}\right\} \\ Q_k(t) - 1/2\log[1+2e^{Q_k(t)}T\sin\left(\theta_k(t)+\phi_k(t)\right)+(e^{Q_k(t)}T)^2] \\ \phi_k(t) \\ f_k(t)+2b_k(t)T \end{bmatrix} + \begin{bmatrix} u_{\theta,k}(t+T) \\ u_{Q,k}(t+T) \\ u_{\phi,k}(t+T) \\ u_{f,k}(t+T) \end{bmatrix}\end{aligned} \tag{4}$$

where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}\{\sigma_\theta^2, \sigma_Q^2, \sigma_\phi^2, \sigma_f^2\})$. When the target speed becomes negative due to acceleration, the Q_k term becomes complex due to the logarithm. This problem can be solved by updating only the real part of Q_k, but adding the imaginary part (which is π when q_k is negative) to the heading parameter ϕ_k, which, in effect, will reverse the direction of the target.

Implicit in (4) is a second order polynomial approximation done for the phase of the signals of interest. Hence, for $t \in [t_i, t_i + T)$, the following relation is assumed on the k^{th} signal:

$$s_k(t) \approx \alpha_k(t_i) e^{j2\pi[f_{0,k}(t_i)t + b_k(t_i)t^2]} \tag{5}$$

where $\alpha_k(t_i)$ is the complex amplitude, $f_k(t) = f_{0,k}(t_i) + 2b_k(t_i)t$ is the instantaneous frequency; and $2b_k(t_i)$ is the rate of change of the instantaneous frequency of the k^{th} signal at time t during the i^{th} batch period. The rest of the paper assumes that dominant narrow-band frequencies of each target are tracked by a separate time-frequency filter. Some frequency tracking examples using Markov chains can be found in [8]–[10]. Note that the state estimate of the IPPF also results in a distribution on the instantaneous target frequencies, which can be exploited by the frequency tracker in determining $b_k(t_{i+1})$'s for the next recursion.

The variances of the components of the state noise vector $\mathbf{u}$ are usually very small, which may lead to sample impoverishment [11] (explained in Sec. V.A.3). In fact, if the process noise is zero, the state variables can be treated as static variables in an estimation problem where using a particle filter may not be appropriate. Techniques to prevent this sample impoverishment or degeneracy are discussed in [6]. Moreover, the state noise vector is chosen to be Gaussian due to its analytical tractability. Justifications of this model can also be found in [6].

B. State Model-II

In this second formulation, the targets are now assumed to have slowly varying accelerations along their heading directions. The new state vector is the following:

$$\mathbf{x}_k(t) \triangleq \begin{bmatrix} \theta_k(t) \\ Q_k(t) \\ \psi_k(t) \\ \phi_k(t) \\ f_k(t) \end{bmatrix} \tag{6}$$

where $\theta_k(t)$, $Q_k(t)$, $\phi_k(t)$, and $f_k(t)$ are as defined above. $\psi_k(t)$ is defined using the k^{th} target's acceleration $a_{\phi,k}$ along its heading direction: $\psi_k(t) \triangleq \frac{a_{\phi,k}}{2r_k(t)}$. Figure 2 illustrates the geometry of the problem used to derive the state update relations.

The state update equation can be derived similar to the State Model-I:

$$\mathbf{x}_k(t+T) = \hat{\mathbf{f}}_{\mathbf{II}}(\mathbf{x}_k(t), \mathbf{u}_k(t+T)) = \mathbf{f}_{\mathbf{II}}(\mathbf{x}_k(t)) + \mathbf{u}_k(t+T) \tag{7}$$

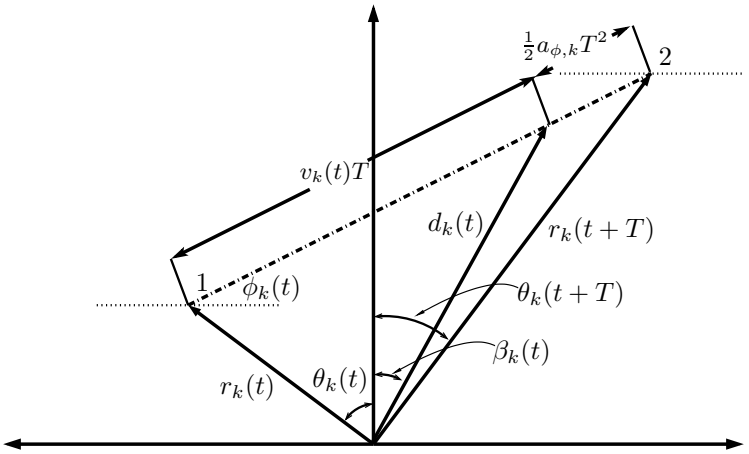


Fig. 2. The k^{th} target is at position 1 at time t and moves to position 2 in T seconds with constant acceleration. Notice that $\beta_k(t)$ corresponds to $\theta_k(t+T)$ in Fig. 1.

with

$$\theta_k(t+T) = \beta_k(t) + \tan^{-1}\left[\frac{T^2\rho_k(t)\cos(\beta_k(t)+\phi_k(t))}{1+T^2\rho_k(t)\sin(\beta_k(t)+\phi_k(t))}\right] + u_{\theta,k}(t+T) \tag{8}$$

where

$$\beta_k(t) = \tan^{-1}\left[\frac{\sin\theta_k(t) + Te^{Q_k(t)}\cos\phi_k(t)}{\cos\theta_k(t) + Te^{Q_k(t)}\sin\phi_k(t)}\right] \tag{9}$$

and

$$\rho_k(t) = \frac{a_{\phi,k}}{2d_k(t)} \tag{10}$$

$$\begin{aligned} Q_k(t+T) = {} & \log\{2T\rho_k(t)[1+2Te^{Q_k(t)}\sin(\theta_k(t)+\phi_k(t)) + T^2e^{2Q_k(t)}]^{1/2} + e^{Q_k(t)}\} \\ & - \frac{1}{2}\log[1+2Te^{Q_k(t)}\sin(\theta_k(t)+\phi_k(t)) + T^2e^{2Q_k(t)}] \\ & - \frac{1}{2}\log[1+2T^2\rho_k(t)\sin(\beta_k(t)+\phi_k(t)) + T^4\rho_k^2(t)] + u_{Q,k}(t+T) \end{aligned} \tag{11}$$

$$\begin{aligned} \psi_k(t+T) &= \frac{\rho_k(t)}{[1+2T^2\rho_k(t)\sin(\beta_k(t)+\phi_k(t)) + T^4\rho_k^2(t)]^{1/2}} + u_{\psi,k}(t+T) \\ \rho_k(t) &= \frac{\psi_k(t)T}{[1+2Te^{Q_k(t)}\sin(\theta_k(t)+\phi_k(t)) + T^2e^{2Q_k(t)}]^{1/2}} \end{aligned} \tag{12}$$

$$\phi_k(t+T) = \phi_k(t) + u_{\phi,k}(t+T) \tag{13}$$

$$f_k(t+T) = f_k(t) + 2b_k(t)T + u_{f,k}(t+T) \tag{14}$$

The state noise is defined similarly as $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}\{\sigma_\theta^2, \sigma_Q^2, \sigma_\psi^2, \sigma_\phi^2, \sigma_f^2\})$. Intuitively, the noise variables

u_ψ and u_Q need to be correlated due to the dependence of the velocity on acceleration along the heading direction. An expression for the correlation between these two noise variables can be derived to better track the targets when the targets are actually not changing their directions. However, as the targets undergo slow maneuvers between each time step, it is the authors' observation that the correlation between the acceleration along the heading direction at the beginning of the batch and the actual target speed becomes much weaker. Hence, the noise variables on the velocity and the acceleration related variables are modelled independently to better accommodate the maneuvering targets as well as for simplicity.

C. Observation Model

The sensor array in the simulations consists of P omnidirectional acoustic sensors situated uniformly on a circle of radius R. A steering vector associated with the array defines the complex array response for a source at DOA θ. The received signal at the p^{th} sensor corresponding to the i^{th} target is first derived using the same second order polynomial approximation on the phases of the signals shown in (5). For an isotropic and non-dispersive medium, the signal received at the p^{th} sensor can be written as

$$\begin{aligned} s_i(t-({\alpha_i}^T\mathbf{z}_p)) &\approx s_i(t)e^{j2\pi[-f_{0,i}(t)({\alpha_i}^T\mathbf{z}_p)-2b_i(t)t({\alpha_i}^T\mathbf{z}_p)+b_i(t)({\alpha_i}^T\mathbf{z}_p)^2]} \\ &\approx s_i(t)e^{j2\pi[b_i(t)({\alpha_i}^T\mathbf{z}_p)^2-f_i(t)({\alpha_i}^T\mathbf{z}_p)]} \end{aligned} \tag{15}$$

where $i = 1, 2, \cdots, K$, $\mathbf{z}_p$ is the p^{th} sensor position, and $\alpha_i \triangleq (1/c)[\ \cos(\theta_i),\ \ \sin(\theta_i)\ ]^T$ is the i^{th} slowness vector in cartesian coordinates. Equation (15) leads to the following array steering vector for the i^{th} source signal:

$$\mathbf{a}(\theta_i) = \begin{bmatrix} e^{j2\pi\{b_i(t)({\alpha_i}^T\mathbf{z}_1)^2-f_i(t)({\alpha_i}^T\mathbf{z}_1)\}} \\ e^{j2\pi\{b_i(t)({\alpha_i}^T\mathbf{z}_2)^2-f_i(t)({\alpha_i}^T\mathbf{z}_2)\}} \\ \vdots \\ e^{j2\pi\{b_i(t)({\alpha_i}^T\mathbf{z}_P)^2-f_i(t)({\alpha_i}^T\mathbf{z}_P)\}} \end{bmatrix} \tag{16}$$

A similar derivation of the array steering vector for signals with constant narrow-band frequencies can be found in [1]. Note that, in the second line of (15), the term $b_i(t)({\alpha_i}^T\mathbf{z}_p)^2$ can be ignored for small aperture sizes and slowly varying frequencies to obtain the same steering vector used in [5].

As mentioned above, the phase characteristics of the signals of interest are approximated with chirps within a batch period T. We also require that each steering vector $\mathbf{a}(\theta_i)$ uniquely correspond to a signal whose direction is the objective of the DOA estimation problem. Signals coming from multiple targets are added to form the observations. For the IPPF, these acoustic observations are updated every $\tau = T/M$

seconds where M denotes the number of batch samples. Then, the array outputs are written as follows:

$$\mathbf{y}(t) = \mathbf{A}(\mathbf{\Theta}(t))\mathbf{s}(t) + \mathbf{n}(t) \qquad t/\tau = 1, 2, \cdots, M \tag{17}$$

In (17), $\mathbf{y}(t)$ is the noisy array output vector, $\mathbf{n}(t)$ is additive noise (e.g., $\mathbf{n}(t) \sim \mathcal{N}(0, \sigma_n^2)$), and $\mathbf{A}(\mathbf{\Theta}(t))$ has the steering vectors in its columns. It should be noted that the sensor positions must be perfectly known in order to define $\mathbf{A}(\mathbf{\Theta}(t))$ for this model [12].

For notational convenience and tractability, the data collected at each time increment τ during the batch period T is stacked to form the data vector $\mathbf{Y}_t : MP \times 1$. The signal vector $\mathbf{S}_t$ and the noise vector $\mathbf{W}_t$ are formed in the same manner. Thus, the array data (or observation) model for the batch period can be compactly written as:

$$\begin{aligned} \mathbf{Y}_t &= \hat{\mathbf{h}}(\mathbf{\Theta}(t), \mathbf{W}_t) = \mathbf{h}(\mathbf{\Theta}(t)) + \mathbf{W}_t \\ \mathbf{Y}_t &= \mathbf{A}_t \mathbf{S}_t + \mathbf{W}_t \end{aligned} \tag{18}$$

where the steering matrix $\mathbf{A}_t = \mathrm{diag}\{\mathbf{A}(\theta(t)), \mathbf{A}(\theta(t+\tau)), \cdots, \mathbf{A}(\theta(t+(M-1)\tau))\}$ implicity incorporates the DOA information of the targets.

D. Observability of the Motion Model

Even though the formulation of the problem as a state and observation model seems intuitive at first, it would be unwise to use this type of formulation if the new state model (7) is not observable given the measurements (18). For the array model, it is assumed that the array can resolve the target DOA parameters uniquely. This leaves only the examination of the motion model to determine the observability of the system [3], [7]. It is verified in [3] that the state model described by (4) satisfies the strongly locally observability rank condition. The strongly locally observability rank condition checks if the determinant of the Jacobian of the state vector with respect to the target DOA is non-zero for observability [13].

It follows then that the new state vector $\mathbf{x}_k(t) \in \mathbb{R}^6$ is observable if $\{\theta_k(t+m\tau); m = 0, 1, \cdots, \Gamma_k - 1\}$ and $\{b_k(t); m = 0, 1, \cdots, \Gamma_k - 1\}$ are sufficient to determine $\mathbf{x}_k(t)$ for a finite integer Γ_k . The Jacobian of the state vector (7) with respect to the target DOA's is very tedious to calculate by hand. However, a symbolic algebra manipulator can be programmed to prove that the Jacobian matrix is full rank for all the regions of the system for the new state formulation, making the new state vector observable.

III. PDF Constructions for the IPPF

The particle filter is a convenient way of recursively updating a target posterior of interest. In this section, we will show the derivations necessary for the operation of the particle filter, which are summarized in Figure 3.

A. Data Likelihood

While formulating the state update equations in our problem, one encounters two nuisance parameters: the signal vector $\mathbf{S}_t$ and the noise variance for the additive Gaussian noise vector $\mathbf{W}_t$. For simplicity, we will assume that the noise variance is approximately constant during the batch period $[t, t+T)$. Following the notation introduced in [6] , we will denote this noise variance as $\sigma_w^2(t)$ corresponding to the batch period starting at time t. The noise has the complex Gaussian probability density function (pdf) described by Goodman [14]. The data likelihood given the signal and noise vectors can be written as follows:

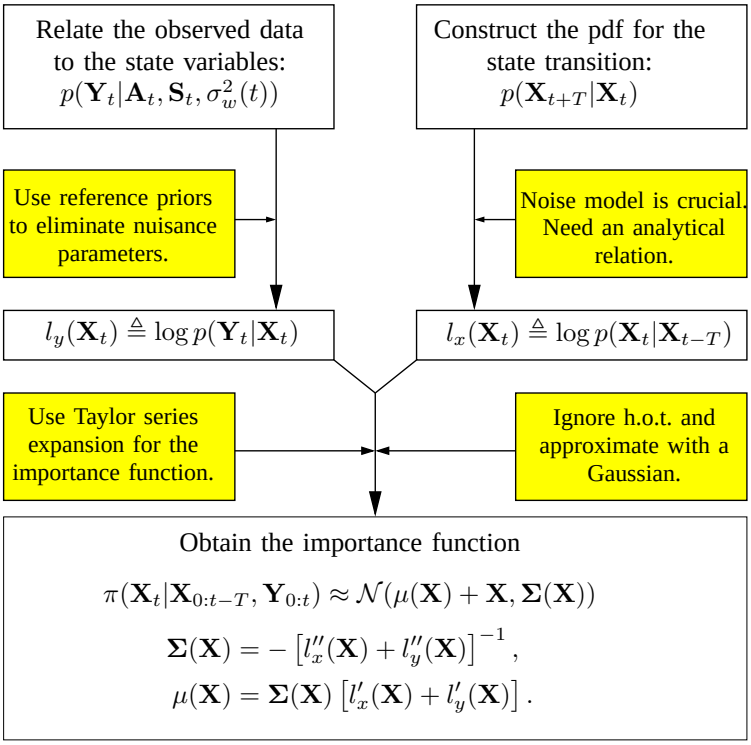


Fig. 3. The mechanics of the necessary derivations needed by the particle filter. $\mathbf{X}$ is chosen to be the predicted state vector by the state update relation without any noise.

$$\begin{aligned} L &\triangleq \log p(\mathbf{Y}_t|\mathbf{A}_t, \mathbf{S}_t, \sigma_w^2(t)) \\ &= -MP\log\pi - MP\log\sigma_w^2(t) - \frac{1}{\sigma_w^2(t)}(\mathbf{Y}_t - \mathbf{A}_t\mathbf{S}_t)^H(\mathbf{Y}_t - \mathbf{A}_t\mathbf{S}_t) \end{aligned} \tag{19}$$

If the priors are known for the signals and noise variance given the state vector at time t, they can be integrated out (a procedure also known as marginalization). If one desires to assume the least about these parameters and let the observed data speak for itself, then the use of reference priors comes into play[2]. Hence, even for moderate sample sizes, the information in the data dominates the *prior* information because of the vague nature of the prior knowledge [16]. The intuitive choice of the prior is usually the uniform prior on the *natural* space of the parameter. A good discussion of these issues can be found in [16], [17], and [18].

The square root of the determinant of the Fisher information matrix is used as our reference prior (a.k.a. Jeffrey's prior). The resulting reference prior is not integrable (and hence, is improper) on the entire unbounded space for the parameter vectors. This, in turn, stipulates compactness arguments on the parameter space such as the ones used in [16], [19]. Assuming that the columns of $\mathbf{A}_t$ are linearly independent, the reference prior is given by [20]

$$p(\mathbf{S}_t|\mathbf{A}_t) \propto |\mathbf{A}_t^H \mathbf{A}_t|^{1/2} \tag{20}$$

where $|.|$ denotes the determinant of a matrix. At this point, we can use (20) to integrate out the signal vector from our problem.

$$\begin{aligned} p(\mathbf{Y}_t|\mathbf{A}_t, \sigma_w^2(t)) &= \int p(\mathbf{Y}_t|\mathbf{A}_t, \mathbf{S}_t, \sigma_w^2(t)) p(\mathbf{S}_t|\mathbf{A}_t) d\mathbf{S}_t \\ \Rightarrow p(\mathbf{Y}_t|\mathbf{A}_t, \sigma_w^2(t)) &\propto \exp\left[-\frac{\mathbf{Y}_t^H (\mathbf{I} - \mathbf{A}_t(\mathbf{A}_t^H \mathbf{A}_t)^{-1} \mathbf{A}_t^H) \mathbf{Y}_t}{\sigma_w^2(t)} \right] \end{aligned} \tag{21}$$

Finally, notice that (20) is not integrable if $\mathbf{S}_t$ has infinite multidimensional support. However, the condition $\{\mathbf{S}_t : |[\mathbf{S}_t]_i| < \gamma_i\}$ can be easily imposed on the i^{th} signal component for some large γ_i. This makes the prior (20) integrable on the signal vector space and, in turn, the marginalization integrals become proper. This condition is always satisfied in practice (e.g., when the signals of interest have finite magnitudes at all times.)

B. State Likelihood

In the previous section, we omitted the motivation for constructing the pdf for the data and put the emphasis on the use of reference priors. Now, it is necessary to elaborate on the reasons for constructing

[2]Bernardo derives the reference prior using an estimation model based on communication channel with a source and data [15]. The reference prior maximizes the mutual information between the source and the data.

the pdf's for the data and the state. The state and observation models (4), (7) and (18) form a hidden Markov model (HMM), which can be compactly described by the following pdf's:

$$\begin{aligned} &p(\mathbf{X}_t|\mathbf{X}_{t-T}) \\ &p(\mathbf{Y}_t|\mathbf{X}_t) \end{aligned} \tag{22}$$

where $\mathbf{X}_t = [\mathbf{x}_1^T(t), \mathbf{x}_2^T(t), \cdots, \mathbf{x}_K^T(t)]^T$ and $p(\mathbf{Y}_t|\mathbf{X}_t) = p(\mathbf{Y}_t|\mathbf{A}_t)$ or $p(\mathbf{Y}_t|\mathbf{A}_t, \sigma_w^2(t))$ depending upon whether or not we treat the noise variance as a known parameter. Here, we introduce a common notation in the particle filtering literature, $\mathbf{z}_{0:t} \triangleq \{\mathbf{z}_0, \mathbf{z}_T, \ldots, \mathbf{z}_t\}$. The recursive update for the HMM model described by (22) can be written as follows [11]:

$$p(\mathbf{X}_{0:t}|\mathbf{Y}_{0:t}) = p(\mathbf{X}_{0:t-T}|\mathbf{Y}_{0:t-T}) \frac{p(\mathbf{Y}_t|\mathbf{X}_t)p(\mathbf{X}_t|\mathbf{X}_{t-T})}{p(\mathbf{Y}_t|\mathbf{Y}_{0:t-T})} \tag{23}$$

Hence, the recursive evaluation of $p(\mathbf{X}_{0:t}|\mathbf{Y}_{0:t})$ requires the pdf's shown in (22). The previous section considered the construction of the second pdf in the model. This section will concentrate on the first pdf in (22).

The objective is to find $p(\mathbf{X}_t|\mathbf{X}_{t-T})$ given the state model. By inspection of (4) or (7), one can see that $\mathbf{X}_t$ is also normal with mean $\mathbf{f}_{\mathbf{I},\mathbf{II}}(\mathbf{X}_{t-T})$ and covariance equal to that of the additive noise. Therefore, we can write the pdf for the state update as follows:

$$p(\mathbf{X}_t|\mathbf{X}_{t-T}) \sim \mathcal{N}(\mathbf{f}_{\mathbf{I},\mathbf{II}}(\mathbf{X}_{t-T}), \mathrm{diag}\{\sigma_\theta^2, \sigma_Q^2, \sigma_\Psi^2, \sigma_\phi^2, \sigma_f^2\}) \tag{24}$$

We have two important remarks on the construction of the pdf's for our problem. The first one is that we generally need the analytical expressions for the pdf's to make use of the particle filter, which does not assume a Gaussian model in general. The second remark is about the model order of the HMM. The motion equations describe a first order HMM model and hence the update equations (23) depend only on the previous state. If more complicated motion equations are formulated in the state model so that the HMM model order increases, then a new recursive update formulation becomes necessary.

C. The Importance Function

An appropriate choice of the importance function $\pi(.)$ may reduce the variance of the simulation errors.[3] However, it was shown analytically in [22] that the importance weights have increasing variance

[3]i.e., if we choose the exact posterior as the importance function then due to the nature of the data generating process the variance of the estimator is inversely proportional to the number of particles N [21].

with time, which leads to increasing estimation errors (or simulation errors, will be used interchangeably). Here, we restate an important result: the unconditional variance of the importance weights, i.e. with the observations $\mathbf{Y}_{0:t}$ being interpreted as random variables, increases over time. This fact is also known as the degeneracy phenomenon: after a few iterations, all but one of the normalized importance weights will be very close to zero [22].

It is crucial to note that the optimal importance function $\pi(\mathbf{X}_t|\mathbf{X}^{(i)}_{0:t-T}, \mathbf{Y}_{0:t})$ is proportional to $p(\mathbf{Y}_t|\mathbf{X}_t)\times p(\mathbf{X}_t|\mathbf{X}^{(i)}_{t-T})$ with the proportionality independent of $\mathbf{X}_t$. We have previously derived the analytical relations for $p(\mathbf{Y}_t|\mathbf{X}_t)$ and $p(\mathbf{X}_t|\mathbf{X}_{t-T})$. Moreover, define

$$\begin{aligned} l_y(\mathbf{X}_t) &\triangleq \log p(\mathbf{Y}_t|\mathbf{X}_t) \\ l_x(\mathbf{X}_t) &\triangleq \log p(\mathbf{X}_t|\mathbf{X}_{t-T}) \end{aligned} \tag{25}$$

$$\begin{aligned} \mathbf{\Sigma}(\mathbf{X}) &= -[l''_x(\mathbf{X}) + l''_y(\mathbf{X})]^{-1} \\ \mu(\mathbf{X}) &= \mathbf{\Sigma}(\mathbf{X})[l'_x(\mathbf{X}) + l'_y(\mathbf{X})] \end{aligned} \tag{26}$$

then, a suboptimal importance function which minimizes the variance of the importance weights is given by the following [6], [22]:

$$\pi(\mathbf{X}_t|\mathbf{X}_{0:t-T}, \mathbf{Y}_{0:t}) \approx \mathcal{N}(\mu(\mathbf{X}) + \mathbf{X}, \mathbf{\Sigma}(\mathbf{X})) \tag{27}$$

$\mathbf{X}$ is judiciously chosen to be the mode of $p(\mathbf{X}_t|\mathbf{X}_{t-T}, \mathbf{Y}_t)$ so that $\mu(\mathbf{X}) \approx \mathbf{0}$ [22].

IV. Algorithm Details

In this section, we will give the details of our modifications to the independent partition particle filtering algorithm by Orton and Fitzgerald. The outline of the IPPF is given in [6] and is repeated for completeness.

A. Partitioning and Data Association

Each particle in the IPPF consists of multiple state vectors, e.g., for three targets, a particle consists of three different state vectors corresponding to each target's motion parameters and frequency. The target association problem is solved by the independence assumption on these partitions. To elaborate, the independence assumption results in partition importance functions $\pi_k(.)$'s. In order to generate the partition importance functions, (26) is calculated for the whole particle; however, only the block diagonal portions of $\mathbf{\Sigma}(\mathbf{X})$ are used for each partition, which results in $\pi_k(\mathbf{x}) \approx \mathcal{N}(\mu(\mathbf{x})+\mathbf{x}, \mathbf{\Sigma}_k(\mathbf{X}))$ with $\mathbf{\Sigma}_k(\mathbf{X})$

Pseudo Code for the IPPF

i. At time t, for each particle i ($i = 1, \ldots, N$) and its partition k ($k = 1, \ldots, K$), $\mathbf{x}_t^{(i)}(k)$:
   - sample from the partition importance function: $\mathbf{x} \sim \pi_k(\mathbf{x}_t(k)|\mathbf{x}_{t-T}^{(i)}(k), \mathbf{Y}_t)$
   - set $\mathbf{x}_t^{(i)}(k) = \mathbf{x}$ and calculate the partition weight $q_k^{(i)}(\mathbf{x})$
   - normalize the partition weights across all particles for each partition:

$$q_k^{(i)} = \frac{q_k^{(i)}}{\sum_i q_k^{(i)}}$$

   - for each partition, resample with replacement across the particles using the distribution generated by $q_k^{(i)}$ and generate a new set of particles $\mathbf{X}_t^{(i)}$ with their respective $q_k^{(i)}$ for each partition. Also, reindex $\mathbf{X}_{t-T}^{(i)}$ accordingly

ii. For each particle i ($i = 1, \ldots, N$), $\mathbf{X}_t^{(i)}$:
   - calculate the importance weights using

$$w_t^{(i)} = w_{t-T}^{(i)} \frac{p(\mathbf{Y}_t|\mathbf{X}_t^{(i)}) p(\mathbf{X}_t^{(i)}|\mathbf{X}_{t-T}^{(i)})}{\pi(\mathbf{X}_t|\mathbf{X}_{t-T}^{(i)}, \mathbf{Y}_t) \prod_k q_k^{(i)}}$$

   - normalize the importance weights $w_t^{(i)}$ across the particles

iii. Resample the particles $\mathbf{X}_t^{(i)}$ using a Metropolis-Hastings scheme keeping the reversibility of the chain

■

corresponding to the k^{th} $(dim\{\mathbf{x}\} \times dim\{\mathbf{x}\})$ block diagonal matrix entry of $\mathbf{\Sigma}(\mathbf{X})$. In particular, the off-diagonal matrices in the particle Hessian corresponding to the cross partitions are ignored. Note that after the particle is formed, the discrepancies generated by this method are augmented by the importance weights, which are calculated using the the full Hessians generated from the new particle.

When the target DOA's cross, previous target states help distinguish the next state through the state update probability. The important thing to remember is that unless one target is moving in tandem with the other target, the partitions for two targets will be different from each other by the other elements of the state vector (e.g, frequency, heading direction, and so on), which will be emphasized by the partition probability $q_k(\mathbf{x})$. Then, the partition cross sampling is used to help the data association by generating particles having high probability across all partitions. This method of generating particles helps the IPPF to propagate particles with good predictive states and automatically handles the data association.

One modification is the use of the state transition probability (24) for the weighted resampling functions $q_k(\mathbf{x})$. This choice alone seems to constrain the particles by the state update equation and hence is expected to have poor performance for maneuvering targets. However, this choice of the weighted resampling function makes sure that the created particles form a cloud around the expected mode of the target state. The maneuvering target cases, on the other hand, are handled by the absolutely critical Monté-

Carlo Markov chain (MCMC) resampling step. A classical Metropolis-Hastings scheme, which keeps the reversibility of the Markov chain can be used to resample the state space using the data likelihood. In the test cases run, the algorithm seems to handle the maneuvers better as the number of iterations increase in the MCMC step. Details of an MCMC resampling scheme are given in [23].

When the targets maneuver, the expected mode of the next state predicted by the state update equation changes. At the resampling state, the particles that are closer to this changed mean survive while the particles around the predicted mean diminish. Hence, the resampling step, in effect, not only makes the particles span most of the state space, but also compensates for the effects of the maneuver. It should be noted that maneuvering has more impact on the heading direction than the other state variables. Hence, a slight modification exploiting this fact in the resampling step may also improve the performance of the algorithm for a given number of particles.

B. Effects of the Frequency Variable

The new state vectors include new motion variables, but the most important extension comes from the frequency variable in the form of deriving new gradients and Hessians (26) for the linearization of the optimal importance function. We will concentrate on l'_y and l''_y since it is necessary to approximate l''_y by a positive definite matrix [4] and l'_y is used in setting up the equations. Derivations of l'_x and l''_x due to the new motion parameters are straightforward. The notation in this section closely follows [3]. Define

$$\begin{aligned} J(t) &\triangleq \mathbf{Y}_t^H(\mathbf{I} - \mathbf{A}_t(\mathbf{A}_t^H\mathbf{A}_t)^{-1}\mathbf{A}_t^H)\mathbf{Y}_t \\ &= \sum_{m=0}^{M-1} |\mathbf{y}(t+m\tau) - \mathbf{P}_A(t+m\tau)\mathbf{y}(t+m\tau)|^2 \\ &= \sum_{m=0}^{M-1} \mathrm{trace}\{\mathbf{P}_A^{\perp}(t+m\tau)\hat{\mathbf{R}}_y(t+m\tau)\} = \sum_{m=0}^{M-1} J_m(t) \end{aligned} \tag{28}$$

where $\mathbf{P}_A$ and $\mathbf{P}_A^{\perp} \triangleq \mathbf{I} - \mathbf{P}_A$ are the projection matrices onto the column and the null spaces of $\mathbf{A}$ and $\mathbf{A}^H$, respectively. $\hat{\mathbf{R}}_y(t+m\tau) = \mathbf{y}(t+m\tau)\mathbf{y}^H(t+m\tau)$ is the one-sample estimate of the array covariance matrix at batch time indexed by m. Note that $l_y = -MJ/\sigma_w^2$, hence the gradients and the

[4] l''_y represents the local covariance of the particles around their modes and is required to be positive definite; however, this positive definiteness is not always available and modifications are required to guarantee that l''_y remains positive definite at each iteration of the filter. See [3] for more discussion.

Hessians of $J(t)$ are linearly related to l'_y and l''_y. Define the gradient of $J(t)$ as G:

$$G \triangleq \frac{\partial J(t)}{\partial \mathbf{X}_t} = \mathrm{vec}\left\{ \frac{1}{M} \sum_{m=0}^{M-1} \Big[V_m(t)\mathrm{diag}(\nabla_\theta J_m) + \Xi_m(t)\mathrm{diag}(\nabla_f J_m) \Big] \right\} \tag{29}$$

where

$$\begin{aligned} V_m(t) &\triangleq V(t+m\tau) = \left[\begin{array}{cccc} \mathbf{v}_1(t+m\tau), & \mathbf{v}_2(t+m\tau), & \cdots, & \mathbf{v}_K(t+m\tau) \end{array} \right] \\ \Xi_m(t) &\triangleq \Xi(t+m\tau) = \left[\begin{array}{cccc} \xi_1(t+m\tau), & \xi_2(t+m\tau), & \cdots, & \xi_K(t+m\tau) \end{array} \right] \end{aligned} \tag{30}$$

with $\mathbf{v}_k(t+m\tau) \triangleq \frac{\partial \theta_k(t+m\tau)}{\partial \mathbf{x}_k(t)}$ and $\xi_k(t+m\tau) \triangleq \frac{\partial f_k(t+m\tau)}{\partial \mathbf{x}_k(t)}$. Moreover,

$$\begin{aligned} \nabla_\theta J_m &= \left[\begin{array}{cccc} \partial J_m/\partial\theta_1(t+m\tau), & \partial J_m/\partial\theta_2(t+m\tau), & \cdots, & \partial J_m/\partial\theta_K(t+m\tau) \end{array} \right] \\ \nabla_f J_m &= \left[\begin{array}{cccc} \partial J_m/\partial f_1(t+m\tau), & \partial J_m/\partial f_2(t+m\tau), & \cdots, & \partial J_m/\partial f_K(t+m\tau) \end{array} \right] \end{aligned} \tag{31}$$

Equation (29) follows from the chain rule, where target frequency and its DOA are assumed independent from each other. Define $\Lambda_{i,m}(t)$ and $\Upsilon_{i,m}(t)$ as the Hessian of $\theta_i(t+m\tau)$ and $f_i(t+m\tau)$ with respect to $\mathbf{X}_t$, respectively; and form $\Lambda_m(t) = \mathrm{diag}[\Lambda_{1,m}(t), \ldots, \Lambda_{K,m}(t)]$ and $\Upsilon_m(t) = \mathrm{diag}[\Upsilon_{1,m}(t), \ldots, \Upsilon_{K,m}(t)]$. Then, the Hessian $H \triangleq \frac{\partial^2 J(t)}{\partial \mathbf{X}_t \partial \mathbf{X}_t}$ is given by

$$H = H_{\theta\theta} + H_{ff} + H_{\theta f} \tag{32}$$

where

$$H_{\theta\theta} = \frac{1}{M} \sum_{m=0}^{M-1} \left\{ [\nabla^2_{\theta\theta} J_m \otimes 1] \odot [\mathrm{vec} V_m(t) \mathrm{vec}^H V_m(t)] + [\mathrm{diag}(\nabla_\theta J_m \otimes I)] \odot \Lambda_m(t) \right\} \tag{33}$$

$$H_{ff} = \frac{1}{M} \sum_{m=0}^{M-1} \left\{ [\nabla^2_{ff} J_m \otimes 1] \odot [\mathrm{vec} \Xi_m(t) \mathrm{vec}^H \Xi_m(t)] + [\mathrm{diag}(\nabla_f J_m \otimes I)] \odot \Upsilon_m(t) \right\} \tag{34}$$

$$H_{\theta f} = \frac{1}{M} \sum_{m=0}^{M-1} \left\{ [\nabla^2_{\theta f} J_m \otimes 1] \odot [\mathrm{vec} V_m(t) \mathrm{vec}^H \Xi_m(t) + \mathrm{vec} \Xi_m(t) \mathrm{vec}^H V_m(t)] \right\} \tag{35}$$

vec stands for the concatenation of the columns of a matrix; $\otimes$ and $\odot$ denote the Kronecker and Schur products, respectively. Moreover, $\mathbf{1}$ and I denote a matrix of all ones and the identity matrix of $\dim\{\mathbf{x}\} \times \dim\{\mathbf{x}\}$.

In order to guarantee the positive definiteness of l''_y, the terms containing $\Lambda_m(t)$ and $\Upsilon_m(t)$ on (33) and (34) are ignored while calculating the Hessian in (32) as discussed in [3]. Define $A_i(t+m\tau) = \partial A(t+m\tau)/\partial\theta_i(t)$, $C_m(t+m\tau) = \partial A(t+m\tau)/\partial f_m(t)$, and $\gamma_i(t+m\tau) = \frac{\partial \mathbf{P}^\perp_A(t+m\tau)\mathbf{X}_{t+m\tau}}{\partial \theta_i(t)}$. The

following derivatives need to be approximated while calculating (33) and (34):

$$\begin{aligned}\frac{\partial^2 J_m}{\partial\theta_i(t)\partial\theta_j(t)} &\simeq 2\mathrm{Re}\{\gamma_i^H(t+m\tau)\gamma_j^H(t+m\tau)\}\\ \frac{\partial^2 J_m}{\partial f_i(t)\partial f_j(t)} &\simeq 2\mathrm{Re}\{\mathrm{tr}[A^{\dagger H}(t+m\tau)C_i^H(t+m\tau)P_A^{\perp}(t+m\tau)C_j^H(t+m\tau)A^{\dagger H}(t+m\tau)]\}\end{aligned} \tag{36}$$

$H_{\theta f}$ is not guaranteed to be negative definite and can be ignored; however, the authors found that with the approximation below, it almost never affects the definiteness of the Hessian and can be used in the calculation of the Hessian (32):

$$\frac{\partial^2 J_m}{\partial\theta_i(t)\partial f_i(t)} \simeq 2\mathrm{Re}\{\mathrm{tr}[A^{\dagger H}(t+m\tau)A_i^H(t+m\tau)P_A^{\perp}(t+m\tau)C_j^H(t+m\tau)A^{\dagger H}(t+m\tau)]\}, \tag{37}$$

V. Simulation Results

In the simulations, our objectives are the following: (i) compare the effectiveness of the two state formulations for tracking targets; (ii) show the effect of the frequency variable on tracking. Comparison of State Model-I with the extended Kalman filter also can be found in [6] and hence is not repeated here. Issues related to initialization of the filters can be found in [23].

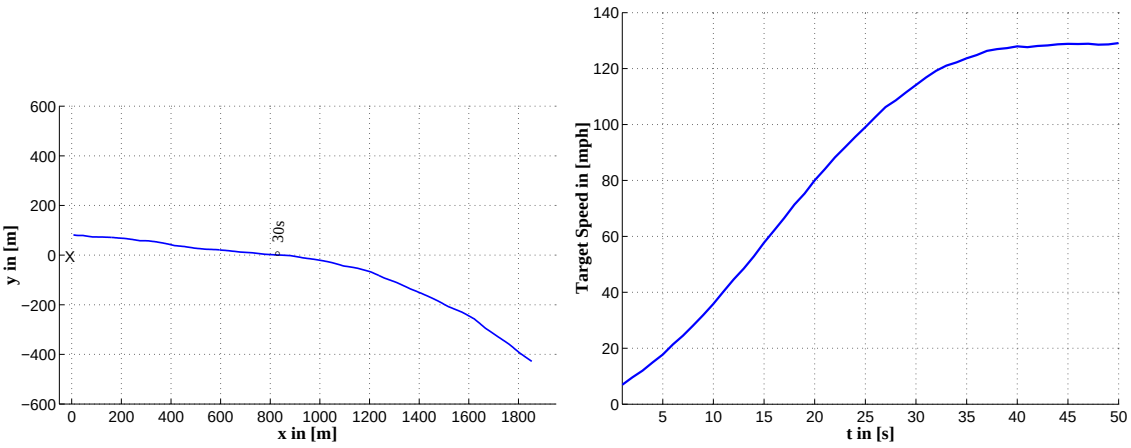
A. Single Target Tracking

A circular sensor array of 15 omnidirectional sensors is used to track a single target. The radius of the array is such that the inter-element spacing is equal to 0.45 times the wavelength (λ) of the target of interest. Figure 4 shows the track and the temporal speed evolution of the target. The simulation parameters are given in Table I.

TABLE I
Simulation Parameters (A)

Number of Particles, N	100
θ noise σ_θ	0.1°
Q noise σ_Q	0.1
ψ noise σ_ψ	0.0001
ϕ noise σ_ϕ	4°
Signal to Noise Ratio, SNR	$7dB$
Target Narrow-band Frequency, f_0	$200Hz$
Number of Batch Samples, M	8

In Table I, some explanation of M, the number of batch samples, is necessary. If the DOA tracking was done by a snapshot algorithm, a much higher number of batch samples would be necessary to

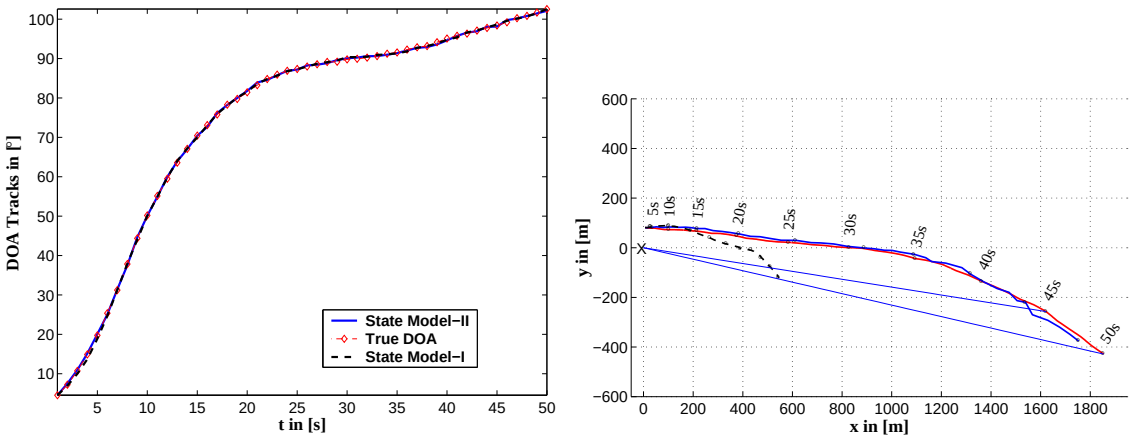


(a) (b)

Fig. 4. (a) One target initially heading in the positive x-direction with speed approximately equal to 7 mph, starts to maneuver at $t = 30s$. The sensor node is situated at the origin. (b) The target has almost constant acceleration between $t = 0$ and $t = 35$. The speed of the target is almost constant during the maneuver.

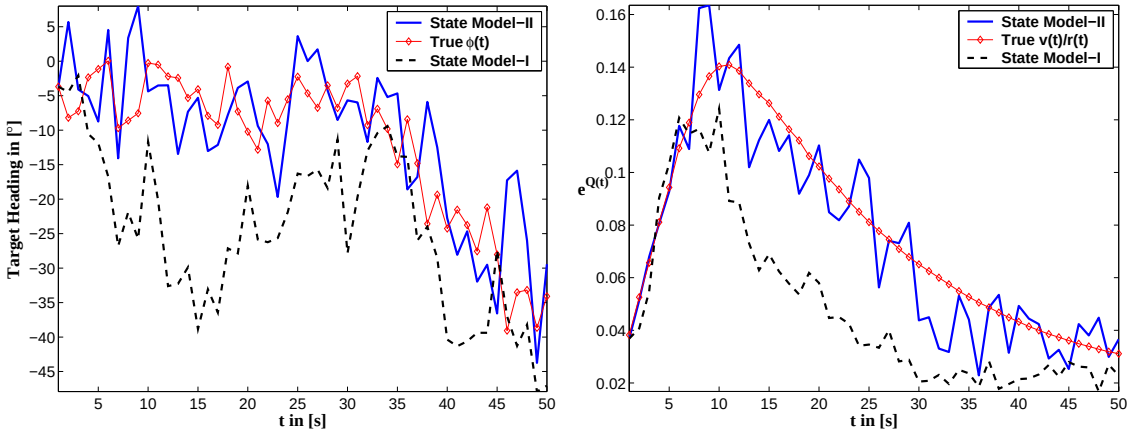
get high resolution DOA's. However, inclusion of the motion dynamics reduces the number of batch samples required to estimate the target DOA's. Ideally, a higher number of batch samples also helps the IPPF estimate the DOA's better: the gradient and Hessian terms that form the mean and covariance matrix of the approximate importance function incorporate more data and hence are expected to improve. Interestingly, the authors determined on synthetic data that even $M = 2$ makes a good approximation to these parameters when the target accelerations are small and the algorithm is initialized close to the true values. Additional improvement in DOA estimation performance by increasing M is empirically found to quickly reach the point of diminishing returns. When State Models I and II are run for the same target track in Fig. 4 with $M = 2$, both models perform the same. However, as M is increased, State Model-II starts to perform better.

Figure 5 illustrates the estimation performance of the IPPF. The DOA estimation of both filters is almost identical as shown in Fig. 5 (a). However, if the true target tracks were estimated using the IPPF state estimates in conjunction with the correct target initial range and speeds (which are *not* available to the IPPF), the resulting tracks would be quite different as illustrated in Fig. 5 (b). Even if the DOA's are the same, State Model-I explains the increase in target speed with a change in the heading direction, but leads to an incorrect heading direction estimate, which is a crucial parameter in data fusion. Figure 6 shows the estimated ϕ and Q parameters of the target. As one can observe, different values of these parameters can lead to the same DOA track even if they do not correspond to the true physical target track.



(a) (b)

Fig. 5. (a) The diamonds are the true DOA track. Dashed line is the estimate from State Model-I while the solid line is the estimate from State Model-II. The two estimates are nearly identical in tracking the target DOA's. (b) Small errors in the DOA's are accentuated by the range. State Model-II obtains a good estimate of the true target track because it uses acceleration.



(a) (b)

Fig. 6. (a) State Model-I tries to explain target accelerations through the heading parameter; hence, it fails to capture the true target parameters. The true target heading direction has additive Gaussian noise with $\sigma_\phi = 4^\circ$, which is marked with diamond. Solid line is the State Model-II estimate. (b) Note that $e^{Q(t)}$ corresponds to $v(t)/r(t)$ for the target. State Model-II has a better $Q(t)$ estimate since the target headings are close to the true headings.

B. Multiple Target Tracking with Varying Narrow-band Frequencies

It is challenging to track two narrow-band targets whose DOA tracks are closer than the Rayleigh resolution. As the reader has seen so far, the IPPF produces high resolution DOA estimates; however, it can fail when the target DOA's as well as the movement parameters are very close. The objective in this section is to show that by incorporating a frequency variable, it is sometimes possible to still track targets even in this difficult case. In this example, three target tracks are simulated with different

frequency scenarios (Fig. 7). Table II summarizes the simulation parameters. Figure 8 shows the tracking

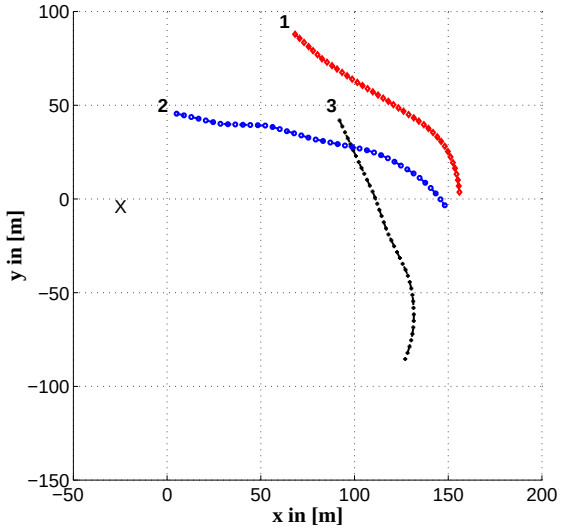


Fig. 7. Three targets move with constant speed with some Brownian disturbance acting on their heading directions. State Model-I is sufficient for the tracking in this case.

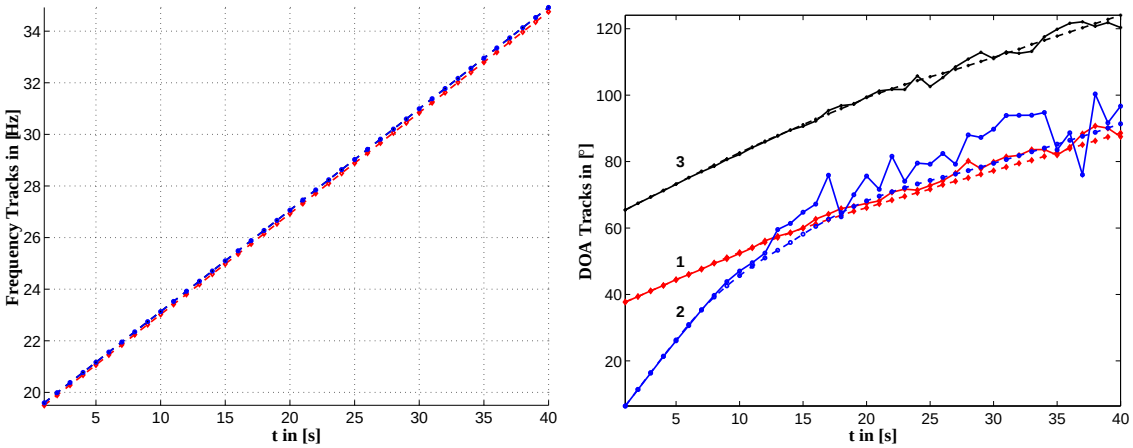
TABLE II
SIMULATION PARAMETERS (B)

Number of Particles, N	100
θ noise σ_θ	0.1°
Q noise σ_Q	0.01
ϕ noise σ_ϕ	4°
Frequency noise σ_f	0.001
Signal to Noise Ratio, SNR	$7dB$
Number of Batch Samples, M	8

performance when the targets have the same time-frequency signatures. Targets marked with the diamonds (1) and the circles (2) have very similar DOA tracks after time $t = 15s$. Initially, the IPPF does a good job in tracking; however, as targets get closer, it fails.

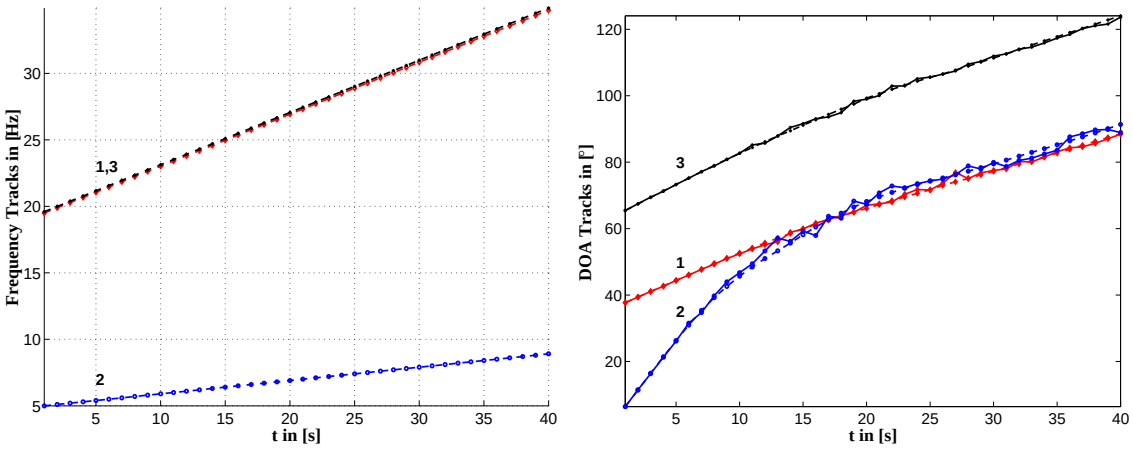
Figure 9 simulates the same problem, but, in this case, targets 1 and 2 have different time-frequency signatures. Note that the IPPF had problems resolving the DOA's of these targets since they had close motion parameters and had the same time-frequency signature. Due to the way the particles are generated (by the partition importance functions $\pi_k(.)$ as explained in section IV), the DOA tracking of target 3 is also affected as shown in Fig. 8. Hence, the change in the time-frequency signature of target 2 helps

the independence assumption on the partitions and improves the DOA tracking performance of the filter as illustrated in Fig. 9 (b). Moreover, the partition weights $q_k(\mathbf{x})$ for each partition not only depends on the motion parameters, but also the frequency. Hence, when the time-frequency signatures of the targets differ, the IPPF can create better predictive states, resulting in overall better tracking.



(a) (b)

Fig. 8. (a) Initial target frequencies are 19.50Hz, 19.60Hz, and 19.60Hz. (b) As the targets 1 and 2 get close to each other, their motion parameters are not sufficient to distinguish their DOA's; hence, the IPPF's DOA tracking perfomance deteriorates.



(a) (b)

Fig. 9. (a) Only the frequency track corresponding to the target marked with circle is changed from the previous example. (b) The tracking is improved due now to the difference in the frequency of the targets that move close to each other.

VI. Conclusions

The state and observation equation set is the heart of any tracking model. In this paper, two new state models and a new observation model are demonstrated using the IPPF. Given an observable state update, it is relatively easy to generate the IPPF equations automatically for an acoustic observation model that can resolve target DOA's. When all the targets have either zero acceleration or small accelerations, both state models (I and II) have nearly identical tracking performance. Since the computational complexity of state model II is about twice that of state model I, this trade-off favors model I. In one of the examples shown, two targets with the same DOA track have quite different tracks in x-y space depending on the acceleration model. The ghost track estimated by model I, due to the incorrect heading direction estimate, is undesirable in the data association process by the multiple nodes; hence, model II is preferred since it included acceleration in its state.

A derivation for the array steering matrix in the case where the target signal phases are locally chirps was presented. In addition, it is demonstrated that if the state is also augmented to include the frequency of individual targets, then it is possible to track targets with similar motion parameters when the targets have different time-frequency signatures. This is an important result, since it enables the acoustic trackers using these new state models to track multiple targets moving close to each other such as in a convoy.

VII. Acknowledgements

The authors would like to thank Matthew Orton for his invaluable comments on the intuition of the IPPF.

References

[1] D.H. Johnson and D.E. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Prentice Hall Signal Processing Series, 1993.

[2] P. Stoica and A. Nehorai, "Music, maximum likelihood, and cramér-rao bound," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, pp. 720–741, May 1989.

[3] Y. Zhou, P.C. Yip, and H. Leung, "Tracking the direction-of-arrival of multiple moving targets by passive arrays: Algorithm," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2655–2666, October 1999.

[4] H. Wang and M. Kaveh, "On the performance of signal-subspace processing-part i: Narrow-band systems," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, pp. 1201–1209, October 1986.

[5] A.B. Gershman and M.G. Amin, "Wideband direction-of-arrival estimation of multiple chirp signals using spatial time-frequency distributions," *IEEE Signal Processing Letters*, vol. 7, no. 6, pp. 152–155, June 2000.

[6] M. Orton and W. Fitzgerald, "A bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 216–223, February 2002.

[7] W.L. Brogan, *Modern Control Theory*, Prentice Hall, 1991.

[8] J.F. Giovannelli, J. Idier, R. Boubertakh, and A. Herment, "Unsupervised frequency tracking beyond the nyquist frequency using markov chains," *IEEE Transactions on Signal Processing*, vol. 50, no. 12, pp. 2905–2914, December 2002.

[9] R.F. Barrett and D.A. Holdsworth, "A frequency tracking using hidden markov models with amplitude and phase information," *IEEE Transactions on Signal Processing*, vol. 41, no. 10, pp. 2965–2976, October 1993.

[10] R.F. Barrett and D.A. Holdsworth, "Frequency line tracking using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 4, pp. 586–598, April 1990.

[11] M.S. 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*, vol. 50, no. 2, pp. 174–188, 2002.

[12] V. Cevher and J.H. McClellan, "2-d sensor position perturbation analysis: Equivalence to awgn on array outputs," Washington DC., 2-4 August 2002.

[13] E.D. Sontag, "On the observability of autonomous discrete time nonlinear system," *Int. J. Contr.*, vol. 36, pp. 867–874, 1982.

[14] N.R. Goodman, "Statistical analysis based on a certain multivariate complex gaussian distribution (an introduction)," *Annals of Mathematical Statistics*, pp. 152–177, March 1963.

[15] J.M. Bernardo, "Reference posterior distributions for bayesian inference," *J.R. Statist. Soc. B*, vol. 41, pp. 113–147, 1979.

[16] J.M. Bernardo and J.M. Ramon, "An introduction to bayesian reference analysis: Inference on the ratio of multinomial parameters," *Bayesian Statistics*, vol. 47, pp. 101–135, 1998.

[17] J.O. Berger and J.M. Bernardo, "On the development of reference priors," *Bayesian Statistics*, vol. 4, pp. 35–60, 1992.

[18] A.D. Lanterman, "Schwarz, wallace, and rissanen: Intertwining themes in theories of model selection," *International Statistical Review*, vol. 69, pp. 185–212, 2001.

[19] J.J. Rissanen, "Fisher information and stochastic complexity," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 40–47, January 1996.

[20] G.E. Box and G.C. Tiao, "On sequential simulation-based methods for bayesian filtering," Tech. Rep. CUED/F-INFENG/TR.310, Department of Engineering, University of Cambridge, 2001.

[21] A.D. Ackerberg, "A new use of importance sampling to reduce computational burden in simulation estimation," under revision at *Review of Economic Studies*, Available http://www.econ.ucla.edu/ackerber/smoonber2.pdf.

[22] A. Doucet, N. Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2001.

[23] M. Orton and W. Fitzgerald, "A bayesian approach to tracking multiple targets using sensor arrays and particle filters," Tech. Rep. CUED/F-INFENG/TR.403, Department of Engineering, University of Cambridge, 2001.