Approximate Conditional Mean Particle Filtering for Linear/Nonlinear Dynamic State Space Models

Derek Yee, James P. Reilly*, T. Kirubarajan, and K. Punithakumar

Department of Electrical and Computer Engineering,

McMaster University, 1280 Main St. W.,

Hamilton, Ontario, Canada L8S 4K1

Abstract

We consider linear systems whose state parameters are separable into linear and nonlinear sets, and evolve according to some known transition distribution, and whose measurement noise is distributed according to a mixture of Gaussians. In doing so, we propose a novel particle filter that addresses the optimal state estimation problem for the aforementioned class of systems. The proposed filter, referred to as the ACM-PF, is a combination of the approximate conditional mean filter and the sequential importance sampling particle filter. The algorithm development depends on approximating a mixture of Gaussians distribution with a moment-matched Gaussian in the weight update recursion. A condition indicating when this approximation is valid is given.

In order to evaluate the performance of the proposed algorithm, we address the blind signal detection problem for an impulsive flat fading channel and the tracking of a maneuvering target in the presence of glint noise. Extensive computer simulations were carried out. For computationally intensive implementations (large number of particles), the proposed algorithm offers performance that is comparable to other state–of–the–art particle filtering algorithms. In the scenario where computational horsepower is heavily constrained, it is shown that the proposed algorithm offers the best performance amongst the considered algorithms for these specific examples.

J. Reilly, corresponding author: ph: 905 525 9140 x22895, fax: 905 521 2922, email: reillyj@mcmaster.ca

I. Introduction

In many real–world applications, unknown quantities are to be estimated given a set of noisy observations. Common examples include target tracking [4], [15], channel tracking in wireless communications [18], [21], and the extraction of speech signals from contaminating audio environments [6], [13], [28]. In many cases, the unknowns can be characterized by a process equation and the observations by a measurement equation, which together, form the so-called dynamic state space model (DSSM). With this DSSM, we can adopt a Bayesian filtering approach, and thereby, recast the problem to one of tracking a hidden process given a set of noisy observations.

Often, observations arrive sequentially in time. Therefore, it is more appropriate to consider filters that recursively estimate the unknowns of interest. More specifically, we aim to recursively compute the exact posterior probability density function (pdf) of interest. Indeed, within the Bayesian framework, the posterior pdf captures all the information about the unknowns. Thus, if an algorithm can recursively deduce the exact posterior pdf of interest, we can refer to it as the optimal solution of the aforementioned Bayesian filtering problem.

For a linear Gaussian DSSM, the celebrated Kalman filter (KF) provides the optimal solution [17]. In general, however, the optimal filter is analytically intractable for the unyielding nonlinear, non–Gaussian DSSM. Thus, a number of researchers have introduced ingenious approximations that have resulted in mathematically tractable suboptimal filters.

Historically, the Extended Kalman filter (EKF) is the first of such filters [2]. The main idea is to invoke linearization and thereby form an approximation of the original nonlinear model that is amenable to an application of the KF. The resulting recursive formulas constitute the EKF. In a number of applications, the EKF has performed adequately. However, there also exist many scenarios where the EKF has performed poorly, in particular for non–Gaussian distributed noise disturbances. Thus, it was suggested to consider filters that involve a collection of EKF's.

These filters are known as Gaussian Sum filters (GSFs) [2], [26], and the underlying assumption is to approximate the true posterior pdf by a Gaussian mixture approximation. Each component in the Gaussian mixture approximation, also called a mixand, is computed by a EKF or a KF. Thus, depending on the nature of the non-Gaussianity, these filters utilize a bank of EKF's or KF's to construct an approximation of the true posterior pdf. As such, these methods are

more powerful, and more complex to implement. Indeed, if we do not introduce any alleviating procedure, the number of mixands exponentially increases over time. Moreover, if the GSF employs a bank of EKF's, it will still suffer from its inaccuracies. Therefore, it is of interest to consider alternative methods.

One such method is known as the approximate conditional mean (ACM) filter [22], [31]. As shown in [22], for a linear DSSM with non-Gaussian observation noise distributed in accordance with a Gaussian mixture distribution, the ACM filter yields near optimal performance. Thus, for this scenario, the ACM filter provides an efficient alternative over the computationally expensive GSF.

Up to this point, it is apparent that we are without a universally effective approach for online signal processing of difficult nonlinear non–Gaussian DSSM's. Therefore, it is necessary to consider alternative solutions. Recently, *particle filtering* has emerged as a promising solution to the general nonlinear non-Gaussian filtering problem [9], [11]. The underlying idea is to use a randomly weighted set of samples or *particles* to recursively build in time, a point-mass approximation of the true posterior pdf. Unlike traditional methods described earlier, particle filters do not make any type of approximating assumptions; rather, they build an approximation of the entire posterior pdf itself. In fact, particle filters are applicable to almost any DSSM where signal variations are present. This is even true for nonlinear dynamics and noise distributed according to non-Gaussian distributions. Consequently, particle filters are expected to outperform popular, traditional EKF type algorithms. Indeed, in the last few years, there have been an abundant number of papers on particle filtering and their applications.

A particularly significant contribution is the development of an efficient particle filter (PF) called the mixture Kalman filter (MKF) [20]. The MKF exploits the conditional linear Gaussian sub–structure of the considered DSSM. Since this approach reduces the dimension of the space in which the particles are sampled from, it is more efficient than a standard implementation of the PF. Although the assumption of a linear Gaussian sub–structure limits the amenable class of DSSM's, a few important classes of DSSM's satisfy this assumption. These include jump Markov linear systems (JMLS) [12], partially observed Gaussian (POG) DSSM [3] and the class of DSSM's corresponding to the transmission of data over a wireless channel in mobile communications [5], [16].

Another important class of models that has received relatively little attention are DSSM's in

the form of

$$\boldsymbol{x}_k^1 = \boldsymbol{A}^1(\boldsymbol{x}_{k-n:k}^2)\boldsymbol{x}_{k-1}^1 + \boldsymbol{F}^1(\boldsymbol{x}_{k-n:k}^2) + \boldsymbol{w}_k^1 \quad (1)$$

$$\boldsymbol{x}_k^2 \sim p(\boldsymbol{x}_k^2|\boldsymbol{x}_{k-n:k-1}^2) \quad (2)$$

$$\boldsymbol{y}_k = \boldsymbol{M}(\boldsymbol{x}_k^2)\boldsymbol{x}_k^1 + \boldsymbol{H}(\boldsymbol{x}_k^2) + \boldsymbol{e}_k \quad (3)$$

where $\boldsymbol{A}^1(\cdot)$, $\boldsymbol{F}^1(\cdot)$, $\boldsymbol{M}(\cdot)$, $\boldsymbol{H}(\cdot)$ are known functions with appropriate dimensions, n is some arbitrary integer, $\boldsymbol{w}_k^1 \sim \mathcal{N}(\boldsymbol{w}_k^1; \boldsymbol{0}, \boldsymbol{Q}_k^1)$, $p(\boldsymbol{x}_k^2|\boldsymbol{x}_{k-n:k-1}^2)$ is a known possibly non–Gaussian transition distribution and $\boldsymbol{e}_k$ is observation noise distributed according to a Gaussian Mixture Model (GMM) of the form

$$p(\boldsymbol{e}_k) = \sum_{j=1}^{N} p_j \mathcal{N}(\boldsymbol{e}_k; \overline{\boldsymbol{e}}_k^{(j)}, \boldsymbol{R}_k^{(j)}) \quad (4)$$

where $\sum_{j=1}^{N} p_j = 1$. [1]

Like most filtering problems, the estimates of interest are the maximum a posteriori (MAP) estimate of $\boldsymbol{x}_k = [\boldsymbol{x}_k^{1T}, \boldsymbol{x}_k^{2T}]^T$, the minimum mean square error (MMSE) estimate of $\boldsymbol{x}_k$ and its associated conditional covariance $cov_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k]$, i.e.,

$$\widehat{\boldsymbol{x}}_k^{MAP} = \arg\max_{\boldsymbol{x}_k} \left\{ p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \right\} \quad (5)$$

$$\mathbb{E}_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k] = \int \boldsymbol{x}_k p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) d\boldsymbol{x}_k \quad (6)$$

$$cov_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k] = \int \tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^T p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) d\boldsymbol{x}_k \quad (7)$$

where $\tilde{\boldsymbol{x}}_k = \boldsymbol{x}_k - \mathbb{E}_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k]$. However, these estimates are generally intractable, thus, it is necessary to resort to some sort of approximate solution. One of the more popular suboptimal solutions is the MKF [20], which in the literature is also known as the Rao–Blackwellized PF [3] or the Marginalized PF [25]. The MKF enjoys asymptotic convergence in the posterior distribution and the typical estimates associated with this distribution. In fact, as the number of particles approaches infinity, the estimates converge toward the true Bayesian estimators of interest. In terms of variance in the estimates, the MKF also bests the standard PF. Thus, it

[1] A mixture of Gaussians effectively models a wide variety of pdf's. This is a useful property as it increases the utility of the DSSM. There are a number of techniques to fit a GMM to an arbitrary pdf. The most popular approaches include the expectation–maximization (EM) algorithm [8] and its variants, e.g., [23]. Other techniques also exist; for a sample see [24], [27].

has received considerable attention in the literature and has found success in many challenging signal processing problems, e.g., see [3], [5], [12]. However, for finite number of particles, it may be possible to outperform the former. To this end, we introduce a novel filter known as the approximate condition mean PF (ACM–PF), and show through simulations, that the proposed algorithm renders an appealing alternative to the sequential importance sampling PF (SIS–PF) and the well known MKF.

The remainder of this paper is organized as follows. In Section 2, we review the SIS–PF, the MKF and introduce the proposed ACM–PF. Section 3 presents some simulation results, and Section 4 concludes this paper.

Notation: We use $(\cdot)_{1:m}$ to indicate all the elements from time 1 to time m. The notation $\mathrm{diag}(\boldsymbol{x})$ denotes the diagonal matrix with vector $\boldsymbol{x}$ on its diagonal. Lastly, a N–dimensional complex Gaussian random vector $\boldsymbol{x}$ has a pdf of the form

$$\mathcal{CN}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\pi^N|\boldsymbol{\Sigma}|}e^{-(\boldsymbol{x}-\boldsymbol{\mu})^H\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}.$$

II. Particle filter Based Algorithms

A. Sequential Importance Sampling Particle Filter

A standard PF such as the sequential importance sampling PF (SIS–PF) employs a weighted set of samples to approximate the *joint* posterior pdf $p(\boldsymbol{x}_k^1, \boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})$. According to this method, if a set of particles $\{\boldsymbol{x}_k^{1,(i)}, \boldsymbol{x}_k^{2,(i)}\}_{i=1}^{N_p}$ is drawn from an *importance distribution* $q(\boldsymbol{x}_k^1, \boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^1, \boldsymbol{x}_{1:k-1}^2, \boldsymbol{y}_k)$, i.e., $(\boldsymbol{x}_k^{1,(i)}, \boldsymbol{x}_k^{2,(i)}) \sim q(\boldsymbol{x}_k^1, \boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^{1,(i)}, \boldsymbol{x}_{1:k-1}^{2,(i)}, \boldsymbol{y}_k)$ for $i = 1, \ldots, N_p$, and each particle $(\boldsymbol{x}_k^{1,(i)}, \boldsymbol{x}_k^{2,(i)})$ is assigned a so–called importance weight

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k^{1,(i)}, \boldsymbol{x}_k^{2,(i)})p(\boldsymbol{x}_k^{1,(i)}, \boldsymbol{x}_k^{2,(i)}|\boldsymbol{x}_{1:k-1}^{1,(i)}, \boldsymbol{x}_{1:k-1}^{2,(i)})}{q(\boldsymbol{x}_k^{1,(i)}, \boldsymbol{x}_k^{2,(i)}|\boldsymbol{x}_{1:k-1}^{1,(i)}, \boldsymbol{x}_{1:k-1}^{2,(i)}, \boldsymbol{y}_{1:k})}, \tag{8}$$

(6) and (7) can be approximated by [10], [11]:

$$\widehat{\mathbb{E}}_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k] = \sum_{i=1}^{N_p} \tilde{w}_k^{(i)} \boldsymbol{x}_k^{(i)} \tag{9}$$

$$\widehat{cov}_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k] = \sum_{i=1}^{N_p} \tilde{w}_k^{(i)} \tilde{\boldsymbol{x}}_k^{(i)} \tilde{\boldsymbol{x}}_k^{(i)\,T} \tag{10}$$

where $\tilde{\boldsymbol{x}}_k^{(i)} = \boldsymbol{x}_k^{(i)} - \widehat{\mathbb{E}}_{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k]$ and $\tilde{w}_k^{(i)} = [\sum_{j=1}^{N_p} w_k^{(j)}]^{-1} w_k^{(i)}$ is the normalized importance weight.

In the above, the likelihood $p(\boldsymbol{y}_k|\boldsymbol{x}_k^1, \boldsymbol{x}_k^2)$ is in the form of

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k^1, \boldsymbol{x}_k^2) = \sum_{j=1}^{N} p_j \mathcal{N}(\boldsymbol{y}_k; \boldsymbol{M}(\boldsymbol{x}_k^2)\boldsymbol{x}_k^1 + \boldsymbol{H}(\boldsymbol{x}_k^2) + \overline{\boldsymbol{e}}_k^{(j)}, \boldsymbol{R}_k^{(j)}) \tag{11}$$

and the prior $p(\boldsymbol{x}_k^1, \boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^1, \boldsymbol{x}_{1:k-1}^2)$, because of the Markov properties of (1) and (2), is given by

$$p(\boldsymbol{x}_k^1, \boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^1, \boldsymbol{x}_{1:k-1}^2) = p(\boldsymbol{x}_k^1|\boldsymbol{x}_{k-1}^1, \boldsymbol{x}_{k-n:k}^2) p(\boldsymbol{x}_k^2|\boldsymbol{x}_{k-n:k-1}^2) \tag{12}$$

where

$$p(\boldsymbol{x}_k^1|\boldsymbol{x}_{k-1}^1, \boldsymbol{x}_{k-n:k}^2) \;\; = \;\; \mathcal{N}(\boldsymbol{x}_k^1; \boldsymbol{A}^1(\boldsymbol{x}_{k-n:k}^2)\boldsymbol{x}_{k-1}^1 + \boldsymbol{F}^1(\boldsymbol{x}_{k-n:k}^2), \boldsymbol{Q}_k^1). \tag{13}$$

Similar to other particle filtering algorithms, the efficiency of the SIS–PF depends on the choice of the importance distribution. There are many choices, with the key constraint being that the support of the importance distribution includes that of the posterior pdf. Often, for the sake of convenience, one adopts the prior $p(\boldsymbol{x}_k^1, \boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^1, \boldsymbol{x}_{1:k-1}^2)$ as the importance distribution, in which case, the importance weights simplify to

$$w_k^{(i)} \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k^{1,(i)}, \boldsymbol{x}_k^{2,(i)}) w_{k-1}^{(i)}. \tag{14}$$

However, if the observations are informative, it maybe profitable to design an importance distribution that exploits the observation in the proposal of new particles. For more details, see [11].

Evidently, the SIS–PF provides a general method to approximate various Bayesian estimators of interest, even for nonlinear possibly non–Gaussian DSSM. Unfortunately, the computational complexity of the algorithm quickly increases with the dimension of the state vector, rendering it ineffective in many practical applications. Therefore, we must consider more efficient implementations of the PF. One such algorithm is the so–called MKF, which is the topic of the following section.

B. Mixture Kalman Filter

The idea of the MKF is to exploit the linear Gaussian sub-structure of the given DSSM. Indeed, if we follow the lead of [20], and introduce an indicator random variable $I_k \in \mathcal{I}^N =$

$\{n|n = 1, \ldots, N\}$ that satisfies

$$I_k = \begin{cases} 1 & \text{if } \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{e}_k; \overline{\boldsymbol{e}}_k^{(1)}, \boldsymbol{R}_k^{(1)}) \\ \vdots & \\ N & \text{if } \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{e}_k; \overline{\boldsymbol{e}}_k^{(N)}, \boldsymbol{R}_k^{(N)}) \end{cases}$$

where $p(I_k = 1) = p_1$, ..., $p(I_k = N) = p_N$, we can note that (1) and (3) conditional on $\boldsymbol{x}_{1:k}^2$ and $I_{1:k}$ forms a linear Gaussian (LG) system in $\boldsymbol{x}_k^1$ for which the KF is the optimal estimator. Intuitively, the random variable I_k indicates the *effective* distribution of $\boldsymbol{e}_k$ at time step k. Thus if we write $p(\boldsymbol{x}_k^1, \boldsymbol{x}_{1:k}^2, I_{1:k}|\boldsymbol{y}_{1:k})$ as

$$p(\boldsymbol{x}_k^1, \boldsymbol{x}_{1:k}^2, I_{1:k}|\boldsymbol{y}_{1:k}) = p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, I_{1:k}, \boldsymbol{y}_{1:k}) p(\boldsymbol{x}_{1:k}^2, I_{1:k}|\boldsymbol{y}_{1:k}), \tag{15}$$

it is apparent that we can use the optimal KF to obtain the Gaussian pdf $p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, I_{1:k}, \boldsymbol{y}_{1:k})$, and the PF to estimate $p(\boldsymbol{x}_{1:k}^2,\, I_{1:k}|\boldsymbol{y}_{1:k})$. This approach is also known as the Rao–Blackwellized Particle filter [3] or the Marginalized Particle filter [25].

At time k, if we draw N_p particles of $(\boldsymbol{x}_k^2, I_k)$ from an importance distribution $q(\boldsymbol{x}_k^2, I_k|\boldsymbol{x}_{1:k-1}^2,\, I_{1:k-1}, \boldsymbol{y}_k)$, that is, $(\boldsymbol{x}_k^{2,(i)}, I_k^{(i)}) \sim q(\boldsymbol{x}_k^2, I_k|\boldsymbol{x}_{1:k-1}^{2,(i)}, I_{1:k-1}^{(i)}, \boldsymbol{y}_k)$ for $i = 1, \ldots, N_p$ and recursively update the importance weights $\{w_k^{(i)}\}_{i=1}^{N_p}$ as

$$w_k^{(i)} \quad \propto \quad \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k}^{2,(i)}, I_{1:k}^{(i)}, \boldsymbol{y}_{1:k-1}) p(\boldsymbol{x}_k^{2,(i)}|\boldsymbol{x}_{1:k-1}^{2,(i)}) p(I_k^{(i)})}{q(\boldsymbol{x}_k^{2,(i)}, I_k^{(i)}|\boldsymbol{x}_{1:k-1}^{2,(i)}, I_{1:k-1}^{(i)}, \boldsymbol{y}_{1:k})} w_{k-1}^{(i)}. \tag{16}$$

Expectations of interest, such as $\mathbb{E}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$, $cov_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$, $\mathbb{E}_{p(\boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^2]$ and $cov_{p(\boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^2]$ can be approximated by

$$\widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^1\right] = \sum_{i=1}^{N_p} \tilde{w}_k^{(i)} \boldsymbol{x}_{k|k}^{1,(i)} \tag{17}$$

$$\widehat{cov}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^1\right] = \sum_{i=1}^{N_p} \tilde{w}_k^{(i)} \left(\boldsymbol{P}_{k|k}^{1,(i)} + (\boldsymbol{x}_{k|k}^{1,(i)} - \widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^1\right])(\boldsymbol{x}_{k|k}^{1,(i)} - \widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^1\right])^T \right) \tag{18}$$

$$\widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^2\right] = \sum_{i=1}^{N_p} \tilde{w}_k^{(i)} \boldsymbol{x}_k^{2,(i)} \tag{19}$$

$$\widehat{cov}_{p(\boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^2\right] = \sum_{i=1}^{N_p} \tilde{w}_k^{(i)} \tilde{\boldsymbol{x}}_k^{2,(i)} \tilde{\boldsymbol{x}}_k^{2,(i)\,T} \tag{20}$$

where $\boldsymbol{x}_{k|k}^{1,(i)} = \mathbb{E}_{p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^{2,(i)}, I_{1:k}^{(i)}, \boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$, $\boldsymbol{P}_{k|k}^{1,(i)} = cov_{p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^{2,(i)}, I_{1:k}^{(i)}, \boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$ and $\tilde{\boldsymbol{x}}_k^{2,(i)} = \boldsymbol{x}_k^{2,(i)} - \widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^2]$.

As mentioned before, given $\{\boldsymbol{x}_{1:k}^2, I_{1:k}\}$, (1) and (3) form a linear Gaussian system in $\boldsymbol{x}_k^1$ for which the Kalman filter is the optimal filter. Therefore, by virtue of Kalman filtering, $p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, I_{1:k}, \boldsymbol{y}_{1:k})$, $p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k-1}^2, I_{1:k-1}, \boldsymbol{y}_{1:k-1})$ and $p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k}^2, I_{1:k}, \boldsymbol{y}_{1:k-1})$ satisfy

$$p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, I_{1:k}, \boldsymbol{y}_{1:k}) = \mathcal{N}(\boldsymbol{x}_k^1; \boldsymbol{x}_{k|k}^1, \boldsymbol{P}_{k|k}^1) \tag{21}$$

$$p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k-1}^2, I_{1:k-1}, \boldsymbol{y}_{1:k-1}) = \mathcal{N}(\boldsymbol{x}_k^1; \boldsymbol{x}_{k|k-1}^1, \boldsymbol{P}_{k|k-1}^1) \tag{22}$$

$$p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k}^2, I_{1:k}, \boldsymbol{y}_{1:k-1}) = \mathcal{N}(\boldsymbol{y}_k; \boldsymbol{y}_{k|k-1}, \boldsymbol{S}_{k|k-1}) \tag{23}$$

where

$$\boldsymbol{x}_{k|k}^1 = \boldsymbol{x}_{k|k-1}^1 + \boldsymbol{P}_{k|k-1}^1 \boldsymbol{M}(\boldsymbol{x}_k^2)^T \boldsymbol{S}_{k|k-1}^{-1} \left(\boldsymbol{y}_k - \boldsymbol{M}(\boldsymbol{x}_k^2)\boldsymbol{x}_{k|k-1}^1 - \boldsymbol{H}(\boldsymbol{x}_k^2) - \overline{\boldsymbol{e}}_k^{(I_k)}\right) \tag{24}$$

$$\boldsymbol{P}_{k|k}^1 = \boldsymbol{P}_{k|k-1}^1 - \boldsymbol{P}_{k|k-1}^1 \boldsymbol{M}(\boldsymbol{x}_k^2)^T \boldsymbol{S}_{k|k-1}^{-1} \boldsymbol{M}(\boldsymbol{x}_k^2) \boldsymbol{P}_{k|k-1}^1 \tag{25}$$

$$\boldsymbol{x}_{k|k-1}^1 = \boldsymbol{A}^1(\boldsymbol{x}_{k-n:k}^2)\boldsymbol{x}_{k-1|k-1}^1 + \boldsymbol{F}^1(\boldsymbol{x}_{k-n:k}^2) \tag{26}$$

$$\boldsymbol{P}_{k|k-1}^1 = \boldsymbol{A}^1(\boldsymbol{x}_{k-n:k}^2)\boldsymbol{P}_{k-1|k-1}^1 \boldsymbol{A}^1(\boldsymbol{x}_{k-n:k}^2)^T + \boldsymbol{Q}_k^1 \tag{27}$$

$$\boldsymbol{y}_{k|k-1} = \boldsymbol{M}(\boldsymbol{x}_k^2)\boldsymbol{x}_{k|k-1}^1 + \boldsymbol{H}(\boldsymbol{x}_k^2) + \overline{\boldsymbol{e}}_k^{(I_k)} \tag{28}$$

$$\boldsymbol{S}_{k|k-1} = \boldsymbol{M}(\boldsymbol{x}_k^2)\boldsymbol{P}_{k|k-1}^1 \boldsymbol{M}(\boldsymbol{x}_k^2)^T + \boldsymbol{R}_k^{(I_k)}. \tag{29}$$

In practice, the efficiency of the MKF also depends on the choice of the importance distribution. The so–called optimal importance distribution (OID) $p(\boldsymbol{x}_k^2, I_k|\boldsymbol{x}_{1:k-1}^{2,(i)}, I_{1:k}, \boldsymbol{y}_{1:k})$ minimizes the variance of the importance weights. Generally, this choice of the importance distribution increases the efficiency of the algorithm. Unfortunately, the OID suffers from two drawbacks. First, it requires the ability to sample from $p(\boldsymbol{x}_k^2, I_k|\boldsymbol{x}_{1:k-1}^{2,(i)}, I_{1:k}, \boldsymbol{y}_{1:k})$. Second, it requires the ability to evaluate

$$w_k^{(i)} \propto \sum_{m \in |\mathcal{I}^N|} \left[\int p(\boldsymbol{y}_k|\boldsymbol{x}_k^2, I_k = m) p(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^2) d\boldsymbol{x}_k^2\right] p(I_k = m) w_{k-1}^{(i)}. \tag{30}$$

Depending on the ability to evaluate the integral over $\boldsymbol{x}_k^2$, it may or may not be possible to evaluate the importance weight $w_k^{(i)}$.

A somewhat more accessible approach is to seek an importance function which closely approximates the OID. Ideally, the adopted distribution will be less computationally demanding than the OID. At the same time, it will also be able to exploit the most recent observation in the proposal of new particles. In the literature, there are a number of strategies to generate this importance distribution. For a sample, see [11] and [10].

Finally, we can select the computationally attractive prior $p(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^2)p(I_k)$ as the importance distribution, for which the importance weights reduce to

$$w_k^{(i)} \propto p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k}^{2,(i)}, I_{1:k}^{(i)}, \boldsymbol{y}_{1:k-1})w_{k-1}^{(i)}. \tag{31}$$

Compared to the prior, the OID and its approximations are more computationally demanding. Thus, the performance gained from the latter must be significant enough to justify their use. In scenarios where $|\mathcal{I}^N| >> 1$, it maybe necessary to choose the computationally attractive prior as the importance distribution. However, the prior does not exploit the most recent observation in the proposal of new particles, i.e., the generated particles may lie in uninteresting regions of the state space, and therefore, may lead to a less efficient implementation of the MKF.

With the aim of increasing the efficiency of the algorithm, we propose to focus particle filtering on $p(\boldsymbol{x}_{1:k}^2|\boldsymbol{y}_{1:k})$ rather than on the higher dimensional pdf $p(\boldsymbol{x}_{1:k}^2, I_{1:k}|\boldsymbol{y}_{1:k})$. Indeed, for particle filtering, it is advantageous to reduce the dimensionality of the space in which we draw samples from [10]. Thus, for the considered DSSM, we endeavor to design a novel PF that exploits the structure of the considered DSSM while dispensing of the need to introduce an indicator random variable I_k. The advantages are clear. By eliminating the need to introduce an indicator random variable I_k, the task of using a PF to approximate $p(\boldsymbol{x}_{1:k}^2, I_{1:k}|\boldsymbol{y}_{1:k})$ is reduced to one of approximating a lower dimensional pdf $p(\boldsymbol{x}_{1:k}^2|\boldsymbol{y}_{1:k})$. Since we are eliminating a possible source of extra Monte Carlo variation, we require a reduced number of particles to achieve a certain level of performance. In the sequel, we will develop these ideas and proceed with the derivation of the ACM–PF.

C. Approximate Conditional Mean Particle Filter

We begin by writing the joint posterior pdf $p(\boldsymbol{x}_{1:k}^1, \boldsymbol{x}_{1:k}^2|\boldsymbol{y}_{1:k})$ as

$$p(\boldsymbol{x}_k^1, \boldsymbol{x}_{1:k}^2|\boldsymbol{y}_{1:k}) = p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, \boldsymbol{y}_{1:k})p(\boldsymbol{x}_{1:k}^2|\boldsymbol{y}_{1:k}). \tag{32}$$

As mentioned before, our aim is to reduce the dimensionality of the space in which the PF draws its samples from. In the following, we only focus the particle filtering on $p(\boldsymbol{x}_{1:k}^2|\boldsymbol{y}_{1:k})$.

Following this idea, one will draw N_p particles of $\boldsymbol{x}_k^2$ according to $q(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^2, \boldsymbol{y}_k)$, i.e., $\boldsymbol{x}_k^{2,(i)} \sim q(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^{2,(i)}, \boldsymbol{y}_k)$ for $i = 1, \ldots, N_p$, then compute the associated importance weight $w_k^{(i)}$ via

$$w_k^{(i)} \propto \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k}^{2,(i)}, \boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_k^{2,(i)}|\boldsymbol{x}_{k-n:k-1}^{2,(i)})}{q(\boldsymbol{x}_k^{2,(i)}|\boldsymbol{x}_{1:k-1}^{2,(i)}, \boldsymbol{y}_{1:k})} w_{k-1}^{(i)}. \tag{33}$$

Having obtained $\{\boldsymbol{x}_k^{2,(i)}, w_k^{(i)}\}_{i=1}^{N_p}$, $\mathbb{E}_{p(\boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^2]$ can be estimated by (19) and $cov_{p(\boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^2]$ by (20), but with $w_k^{(i)}$ given by (33) not (16). As for $\mathbb{E}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$ and $cov_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$, these are approximated by

$$\widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^1\right] = \sum_{i=1}^{N_p} \tilde{w}_k^{(i)} \boldsymbol{x}_{k|k}^{1,(i)} \tag{34}$$

$$\widehat{cov}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^1\right] = \sum_{i=1}^{N_p} \tilde{w}_k^{(i)} \Bigg(\boldsymbol{P}_{k|k}^{1,(i)} + (\boldsymbol{x}_{k|k}^{1,(i)} - \widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^1\right])(\boldsymbol{x}_{k|k}^{1,(i)} - \widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}\left[\boldsymbol{x}_k^1\right])^T \Bigg) \tag{35}$$

where $\boldsymbol{x}_{k|k}^{1,(i)} = \mathbb{E}_{p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^{2,(i)}, \boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$ and $\boldsymbol{P}_{k|k}^{1,(i)} = cov_{p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^{2,(i)}, \boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$. However, unlike the MKF, we cannot use the KF to sequentially in time compute $\boldsymbol{x}_{k|k}^{1,(i)}$ and $\boldsymbol{P}_{k|k}^{1,(i)}$ for all time step k. Indeed, since

$$p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, \boldsymbol{y}_{1:k}) \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k^1, \boldsymbol{x}_k^2) \underbrace{\int p(\boldsymbol{x}_k^1|\boldsymbol{x}_{k-1}^1, \boldsymbol{x}_{1:k}^2) p(\boldsymbol{x}_{k-1}^1|\boldsymbol{x}_{1:k-1}^2, \boldsymbol{y}_{1:k-1}) d\boldsymbol{x}_{k-1}^1}_{p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, \boldsymbol{y}_{1:k-1})}, \tag{36}$$

it is a GMM such that the number of mixands exponentially increases with k. Therefore, we need a growing symphony of KF's to calculate $(\boldsymbol{x}_{k|k}^{1,(i)}, \boldsymbol{P}_{k|k}^{1,(i)})$, each corresponding to one mixand of $p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, \boldsymbol{y}_{1:k})$. In order to limit computational complexity, we approximate $p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, \boldsymbol{y}_{1:k-1})$ with a Gaussian pdf $\mathcal{N}(\boldsymbol{x}_k^1; \boldsymbol{x}_{k|k-1}^1, \boldsymbol{P}_{k|k-1}^1)$ for all time k. In (36), we write[2]

$$p(\boldsymbol{x}_k^1|\boldsymbol{x}_{1:k}^2, \boldsymbol{y}_{1:k-1}) = \mathcal{N}(\boldsymbol{x}_k^1; \boldsymbol{x}_{k|k-1}^1, \boldsymbol{P}_{k|k-1}^1). \tag{37}$$

This approximation is valid provided that the empirically–derived Condition 1 below is satisfied:

Condition 1: a moment matched Gaussian distribution appears to be a close match to the mixture, as long as its components are not too far apart; namely, the distance between the means for the components is up to about two standard deviations [4].

If this condition is violated, the ACM–PF may not be the algorithm of choice for the application under consideration.

[2]The strategy of approximating a multi–modal distribution with a single Gaussian, has in the past, led to many useful algorithms. Most well known being the popular IMM filter which has found considerable success in many practical applications [4]. The classical Gaussian sum filter also relies on this approximation. For other examples of such a strategy, see [22], [29], [31]. While these references do not directly justify the approximation employed in our algorithm, it does lend support to the "practical" utility of this approximation.

Under (37), it can be shown that the predicted measurement pdf, in (33), is

$$p(\boldsymbol{y}_k|\boldsymbol{x}^2_{1:k},\boldsymbol{y}_{1:k-1}) = \sum_{j=1}^{N} p_j \mathcal{N}(\boldsymbol{y}_k;\boldsymbol{y}^{(j)}_{k|k-1},\boldsymbol{S}^{(j)}_{k|k-1}) \tag{38}$$

where

$$\begin{aligned} \boldsymbol{y}^{(j)}_{k|k-1} &= \boldsymbol{M}(\boldsymbol{x}^2_k)\boldsymbol{x}^1_{k|k-1} + \boldsymbol{H}(\boldsymbol{x}^2_k) + \overline{\boldsymbol{e}}^{(j)}_k & (39)\\ \boldsymbol{S}^{(j)}_{k|k-1} &= \boldsymbol{M}(\boldsymbol{x}^2_k)\boldsymbol{P}^1_{k|k-1}\boldsymbol{M}(\boldsymbol{x}^2_k)^T + \boldsymbol{R}^{(j)}_k. & (40)\end{aligned}$$

More importantly is that we can now derive a computationally attractive recursion for the conditional mean and covariance of $\boldsymbol{x}^1_k$. In particular, if we denote the gradient operator as ∇, it can be shown that

$$\begin{aligned} \boldsymbol{x}^1_{k|k} &= \boldsymbol{x}^1_{k|k-1} + \boldsymbol{P}^1_{k|k-1}\boldsymbol{M}(\boldsymbol{x}^2_k)^T g_k(\boldsymbol{y}_k) & (41)\\ \boldsymbol{P}^1_{k|k} &= \boldsymbol{P}^1_{k|k-1} - \boldsymbol{P}^1_{k|k-1}\boldsymbol{M}(\boldsymbol{x}^2_k)^T G_k(\boldsymbol{y}_k)\boldsymbol{M}(\boldsymbol{x}^2_k)\boldsymbol{P}^1_{k|k-1} & (42)\end{aligned}$$

where

$$\begin{aligned} \boldsymbol{x}^1_{k|k-1} &= \boldsymbol{A}^1(\boldsymbol{x}^2_{k-n:k})\boldsymbol{x}^1_{k-1|k-1} + \boldsymbol{F}^1(\boldsymbol{x}^2_{k-n:k}) & (43)\\ \boldsymbol{P}^1_{k|k-1} &= \boldsymbol{A}^1(\boldsymbol{x}^2_{k-n:k})\boldsymbol{P}^1_{k-1|k-1}\boldsymbol{A}^1(\boldsymbol{x}^2_{k-n:k})^T + \boldsymbol{Q}^1_k & (44)\end{aligned}$$

and

$$\begin{aligned} g_k(\boldsymbol{y}_k) &= -\frac{1}{p(\boldsymbol{y}_k|\boldsymbol{x}^2_{1:k},\boldsymbol{y}_{1:k-1})}\nabla_{\boldsymbol{y}_k} p(\boldsymbol{y}_k|\boldsymbol{x}^2_{1:k},\boldsymbol{y}_{1:k-1}) & (45)\\ G_k(\boldsymbol{y}_k) &= \nabla_{\boldsymbol{y}_k} g_k(\boldsymbol{y}_k)^T. & (46)\end{aligned}$$

In the above, $g_k(\boldsymbol{y}_k)$ is the so–called score function of $p(\boldsymbol{y}_k|\boldsymbol{x}^2_{1:k},\boldsymbol{y}_{1:k-1})$ and $G_k(\boldsymbol{y}_k)$ is the gradient of $g_k(\boldsymbol{y}_k)^T$. For the DSSM under consideration, these are provided as follows:

Proposition 1: Given (38), $g_k(\boldsymbol{y}_k)$ and $G_k(\boldsymbol{y}_k)$ satisfy

$$g_k(\boldsymbol{y}_k) = p(\boldsymbol{y}_k|\boldsymbol{x}^2_{1:k},\boldsymbol{y}_{1:k-1})^{-1}\sum_{j=1}^{N} p_j \mathcal{N}(\boldsymbol{y}_k;\boldsymbol{y}^{(j)}_{k|k-1},\boldsymbol{S}^{(j)}_{k|k-1})[\boldsymbol{S}^{(j)}_{k|k-1}]^{-1}(\boldsymbol{y}_k - \boldsymbol{y}^{(j)}_{k|k-1}) \tag{47}$$

and

$$\begin{aligned} G_k(\boldsymbol{y}_k) = \sum_{j=1}^{N} p_j \frac{\mathcal{N}(\boldsymbol{y}_k;\boldsymbol{y}^{(j)}_{k|k-1},\boldsymbol{S}^{(j)}_{k|k-1})[\boldsymbol{S}^{(j)}_{k|k-1}]^{-1}}{p(\boldsymbol{y}_k|\boldsymbol{x}^2_{1:k},\boldsymbol{y}_{1:k-1})} \times \Bigg(\boldsymbol{I}_{n_y\times n_y} \\ + \boldsymbol{S}^{(j)}_{k|k-1} g(\boldsymbol{y}_k)(\boldsymbol{y}_k - \boldsymbol{y}^{(j)}_{k|k-1})^T[\boldsymbol{S}^{(j)}_{k|k-1}]^{-1} - (\boldsymbol{y}_k - \boldsymbol{y}^{(j)}_{k|k-1})(\boldsymbol{y}_k - \boldsymbol{y}^{(j)}_{k|k-1})^T[\boldsymbol{S}^{(j)}_{k|k-1}]^{-1} \Bigg) \end{aligned} \tag{48}$$

where n_y is the dimension of $\boldsymbol{y}_k$.

Proof: See the Appendix.

Interestingly, the recursions shown above turn out to be the equations of the ACM filter [22], [29], [31]. In light of this, we recognize that the proposed algorithm is, in fact, an appropriate combination of the ACM filter and the SIS–PF. For this reason, we have referred to the proposed algorithm as the ACM–PF.

For the choice of the importance distribution $q(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^2, \boldsymbol{y}_{1:k})$, it is common to either choose the prior $p(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^2)$, or the OID $p(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^2, \boldsymbol{y}_{1:k})$ for sampling $\boldsymbol{x}_k^2$. The computationally attractive prior is readily given by (2), and the OID for (1)-(3) in its most general form is analytically intractable. Fortunately, there are two important classes of model for which we can derive a finite dimensional approximation of the OID. They are as follows:

- If $\boldsymbol{x}_k^2$ is a discrete-valued random process and the set forming the range of $\boldsymbol{x}_k^2$ is $\mathcal{X}^2$, i.e., $\boldsymbol{x}_k^2 \in \mathcal{X}^2$, then an approximation of the OID is given by

$$q(\boldsymbol{x}_k^2 = m|\boldsymbol{x}_{1:k-1}^2, \boldsymbol{y}_{1:k}) = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k-1}^2, \boldsymbol{x}_k^2 = m, \boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_k^2 = m|\boldsymbol{x}_{k-n:k-1}^2)}{\sum_{n\in\mathcal{X}^2} p(\boldsymbol{y}_k|\boldsymbol{x}_{1:k-1}^2, \boldsymbol{x}_k^2 = n, \boldsymbol{y}_{1:k-1})p(\boldsymbol{x}_k^2 = n|\boldsymbol{x}_{k-n:k-1}^2)}. \quad (49)$$

- If $\boldsymbol{H}(\boldsymbol{x}_k^2) = \boldsymbol{H}\boldsymbol{x}_k^2$, $\boldsymbol{M}(\boldsymbol{x}_k^2) = \boldsymbol{M}$ and (2) is equal to a Gaussian pdf, i.e., $p(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^2) = \mathcal{N}(\boldsymbol{x}_k^2; \boldsymbol{A}^2\boldsymbol{x}_{k-1}^2, \boldsymbol{Q}_k^2)$, the associated suboptimal importance distribution is in the form of

$$q(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^2, \boldsymbol{y}_{1:k}) = \sum_{j=1}^{N} \overline{p}_j \mathcal{N}(\boldsymbol{x}_k^2; \overline{\boldsymbol{x}}_k^{2,(j)}, \overline{\boldsymbol{P}}_k^{2,(j)}) \quad (50)$$

where

$$\overline{p}_j = \frac{p_j \mathcal{N}(\boldsymbol{y}_k; \boldsymbol{H}\boldsymbol{A}^2\boldsymbol{x}_{k-1}^2, \boldsymbol{H}\boldsymbol{Q}_k^2\boldsymbol{H}^T + \boldsymbol{S}_{k|k-1}^{(j)})}{\sum_{m=1}^{N} p_m \mathcal{N}(\boldsymbol{y}_k; \boldsymbol{H}\boldsymbol{A}^2\boldsymbol{x}_{k-1}^2, \boldsymbol{H}\boldsymbol{Q}_k^2\boldsymbol{H}^T + \boldsymbol{S}_{k|k-1}^{(m)})} \quad (51)$$

$$\overline{\boldsymbol{x}}_k^{2,(j)} = \boldsymbol{A}^2\boldsymbol{x}_{k-1}^2 + \boldsymbol{W}_k^{(j)}(\boldsymbol{y}_k - \boldsymbol{M}\boldsymbol{x}_{k|k-1}^1 - \boldsymbol{H}\boldsymbol{A}^2\boldsymbol{x}_{k-1}^2 - \overline{\boldsymbol{e}}_k^{(j)}) \quad (52)$$

$$\overline{\boldsymbol{P}}_k^{2,(j)} = \boldsymbol{Q}_k^2 - \boldsymbol{W}_k^{(j)}\boldsymbol{H}\,\boldsymbol{Q}_k^{2\,T} \quad (53)$$

and $\boldsymbol{W}_k^{(j)} = \boldsymbol{Q}_k^2\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{Q}_k^2\boldsymbol{H}^T + \boldsymbol{S}_{k|k-1}^{(j)})^{-1}$.

The above suboptimal importance distributions exploit the information in the most recent observation $\boldsymbol{y}_k$. Hence, it incorporates additional information into the proposal of new particles $\boldsymbol{x}_k^{2,(i)}$, and thereby, improves the efficiency of the ACM-PF.

Equations (33), (38), (2) and an appropriately chosen importance distribution $q(\boldsymbol{x}_k^2|\boldsymbol{x}_{1:k-1}^2, \boldsymbol{y}_{1:k})$ are the main ingredients of the ACM–PF. Although the adopted approximation (37) may introduce some bias in the estimates, we find through extensive simulations that the proposed

combination of the ACM filter and the SIS–PF does indeed render an effective alternative to the aforementioned MKF. The algorithm for the ACM-PF is summarized in Table 1.

D. The Degeneracy Problem

In practice, all the aforementioned algorithms suffer from *the Degeneracy problem* [11]. That is, after a few iterations, all but a few particles possess insignificant weights. As a consequence, the PF yields biased estimates with large variance. Therefore, to mitigate this problem and to design a workable PF, we introduce the resampling of particles [10], [14]. The basic idea is to discard particles with weak importance weights and to multiply ones with significant importance weights. In practice, it is not necessary to implement resampling at every time step k. Hence, we only introduce resampling whenever the effective sample size [19]: $\widehat{N}_{eff} = 1/\sum_{i=1}^{N_p}(\tilde{w}_k^{(i)})^2$ is below the threshold $N_{th} = 0.8N_p$. Since stratified resampling introduces minimum variance in the class of unbiased schemes, we use the former in the resampling of the particles.

III. SIMULATIONS

Two experiments were conducted to test the performance of the proposed ACM–PF. The first addresses the blind detection problem for an impulsive flat fading channel. The second studies the tracking of a maneuvering target in the presence of glint noise [32]. For an additional experiment, see [33], where we addressed the filtering problem for a time–varying autoregressive signal observed under non–Gaussian noise.

1) Example 1: The Rayleigh flat fading channel h_k is modeled by a second order autoregressive AR(2) model [16]. In usual state space form, the transmission of differentially encoded BPSK (DBPSK) symbols over a complex impulsive fading channel can be written as

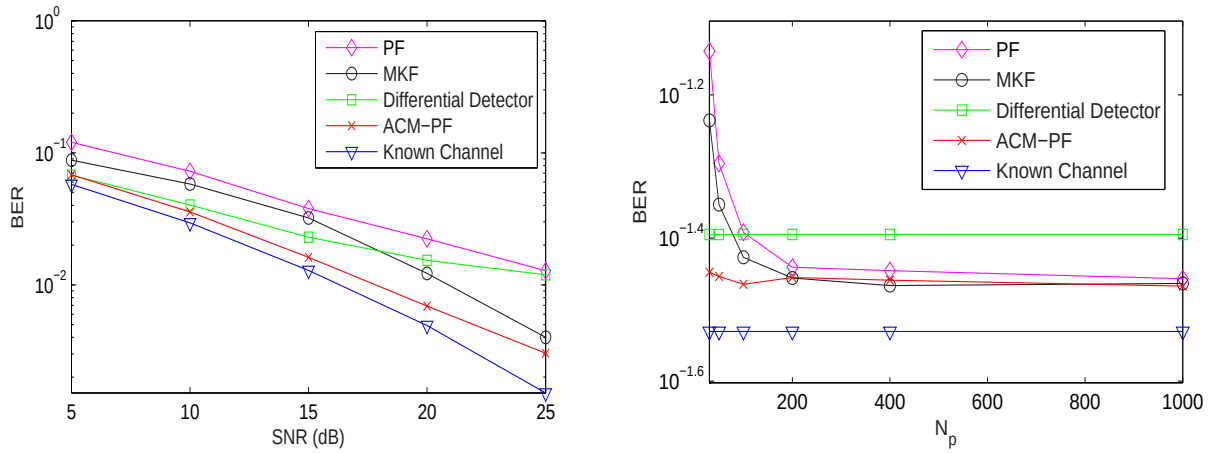
$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{g}w_k \tag{54}$$

$$s_k = d_k s_{k-1} \tag{55}$$

$$y_k = \boldsymbol{g}^T \boldsymbol{x}_k s_k + e_k \tag{56}$$

where

$$\boldsymbol{F} = \begin{bmatrix} \gamma_1 & \gamma_2 \\ 1 & 0 \end{bmatrix}, \qquad \boldsymbol{g} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$



(a) BER versus SNR.

(b) BER versus N_p at a SNR of 10 dB.

Fig. 1. Obtained results are averaged over 60000 DBPSK symbols. In Fig. 1(a), each PF based detector was implemented with $N_p = 30$ particles and used the appropriate prior as the importance distribution. We have $\epsilon = 0.1$ and $\sigma_I^2/\sigma_n^2 = 100$.

d_k is the BPSK symbol, s_k is the DBPSK symbol and $\boldsymbol{x}_k = [h_k \;\; h_{k-1}]^T$. To model impulsive noise, we follow [1], [31], and assume that e_k has a two–term Gaussian mixture distribution of the form

$$p(e_k) \;\; = \;\; (1-\epsilon)\,\mathcal{CN}(e_k; 0, \sigma_n^2) + \epsilon\,\mathcal{CN}(e_k; 0, \sigma_I^2). \tag{57}$$

Observe that $\mathcal{CN}(e_k; 0, \sigma_n^2)$ corresponds to the probability density function (pdf) of the ambient background noise whereas $\mathcal{CN}(e_k; 0, \sigma_I^2)$ corresponds to the pdf of the impulsive noise component. ϵ represents the probability that an impulse will occur.

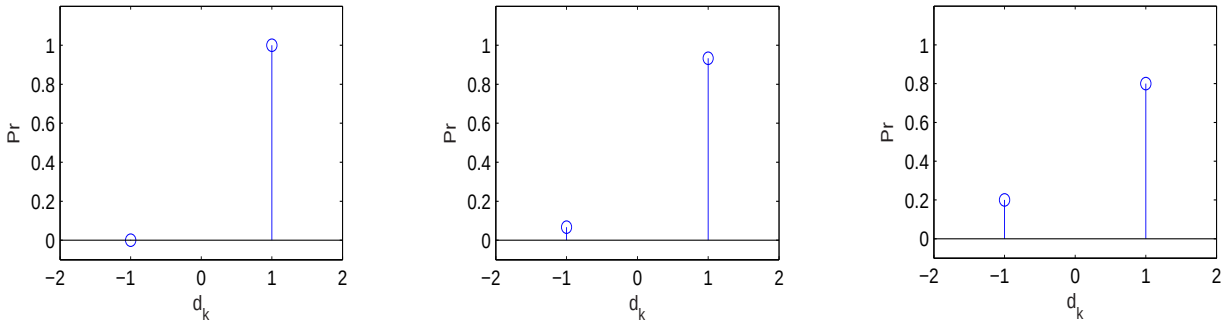
Before we make any additional progress, let us reformulate (54) to (56) in terms of *real* stochastic processes. For reasons detailed below, we rewrite (54) to (56) as

$$\tilde{\boldsymbol{x}}_k \;\; = \;\; \tilde{\boldsymbol{F}}\tilde{\boldsymbol{x}}_{k-1} + \tilde{\boldsymbol{w}}_k \tag{58}$$

$$s_k \;\; = \;\; d_k s_{k-1} \tag{59}$$

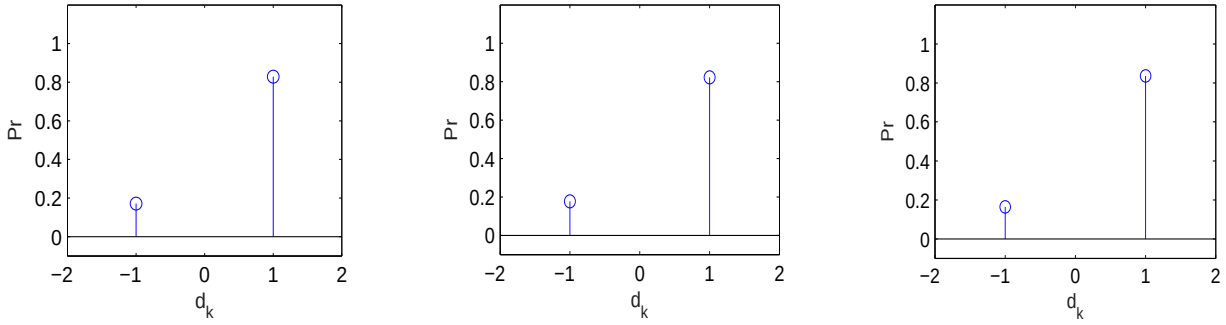
$$\tilde{\boldsymbol{y}}_k \;\; = \;\; \boldsymbol{H}(s_k)\tilde{\boldsymbol{x}}_k + \tilde{\boldsymbol{e}}_k \tag{60}$$

where $\tilde{\boldsymbol{x}}_k = [\mathrm{Re}\{h_k\} \;\; \mathrm{Im}\{h_k\} \;\; \mathrm{Re}\{h_{k-1}\} \;\; \mathrm{Im}\{h_{k-1}\}]^T$, $\tilde{\boldsymbol{y}}_k = [\mathrm{Re}\{y_k\} \;\; \mathrm{Im}\{y_k\}]^T$, $\boldsymbol{H}(s_k) =$



(a) SIS–PF estimate (b) MKF estimate (c) ACM–PF estimate

Fig. 2. Estimate of $p(d_{100}|\tilde{\boldsymbol{y}}_{1:100})$ via the SIS–PF, MKF and the ACM–PF. To generate each estimate, each PF used $N_p = 30$ particles and the appropriate prior as the importance distribution.



(a) SIS–PF estimate (b) MKF estimate (c) ACM–PF estimate

Fig. 3. Estimate of $p(d_{100}|\tilde{\boldsymbol{y}}_{1:100})$ via the SIS–PF, MKF and the ACM–PF. To generate each estimate, each PF used $N_p = 2000$ particles and the appropriate prior as the importance distribution.

$[s_k \boldsymbol{I}_{2\times 2} \;\; \boldsymbol{0}_{2\times 2}]$, $\tilde{\boldsymbol{e}}_k = [\mathrm{Re}\{e_k\} \;\; \mathrm{Im}\{e_k\}]^T$,

$$\tilde{\boldsymbol{F}} = \begin{bmatrix} \gamma_1 \boldsymbol{I}_{2\times 2} & \gamma_2 \boldsymbol{I}_{2\times 2} \\ \boldsymbol{I}_{2\times 2} & \boldsymbol{0}_{2\times 2} \end{bmatrix}$$

and $\tilde{\boldsymbol{w}}_k = [\mathrm{Re}\{w_k\} \;\; \mathrm{Im}\{w_k\} \;\; 0 \;\; 0]^T$ with covariance matrix

$$\tilde{\boldsymbol{Q}}_k = \begin{bmatrix} \mathrm{diag}([\frac{\mathbb{E}[|\mathrm{w_k}|^2]}{2} \;\; \frac{\mathbb{E}[|\mathrm{w_k}|^2]}{2}]^{\mathrm{T}}) & \boldsymbol{0}_{2\times 2} \\ \boldsymbol{0}_{2\times 2} & \boldsymbol{0}_{2\times 2} \end{bmatrix}. \tag{61}$$

In this form, all the stochastic processes of interest are real. More importantly is that the new DSSM falls into the framework of (1) to (3), therefore, it is amenable to the application of the standard PF, the MKF, and the proposed ACM–PF.

For the simulations, the SNR (dB) was defined as

$$SNR_{dB} = 10\log_{10}\left(\frac{1}{\mathbb{E}[|e_k|^2]}\right) \tag{62}$$

where $\mathbb{E}[|e_k|^2] = (1-\epsilon)\sigma_n^2 + \epsilon\sigma_I^2$, $\epsilon = 0.1$, and $\sigma_I^2/\sigma_n^2 = 100$. The carrier frequency was set to $f_c = 2.4$ GHz, the symbol rate to $1/T = 7353$ symbols/sec, and the speed of the vehicle to $v = 150$ km/h so that the resulting time-Doppler fading is given by $f_d T = 0.045$.

The implemented algorithms are the simple differential detector (DD), the PF, the MKF, and the ACM–PF. For the DD, the detected BPSK symbol is given by $\widehat{d}_k = \mathrm{sign}(\mathrm{Re}\{y_k y_{k-1}^*\})$. For each PF, we use the appropriate prior as the importance distribution. To lower bound these algorithms, we also implemented a detector that has perfect knowledge of the considered Rayleigh fading channel.

Fig. 1(a) shows the bit error rate (BER) vs. signal to noise ratio (SNR) averaged over 60000 DBPSK symbols. Between the PF based detectors, it can be seen that the ACM–PF significantly outperforms the scheme based on the SIS–PF and the MKF. More significant is that for practical values of the SNR, the attained BER is very close to that which is achieved by the receiver with perfect knowledge of the considered Rayleigh fading channel.

In Fig. 1(b), we show the BER vs. N_p for an SNR of 10 dB. Here, the ACM–PF only uses 30 particles to achieve a BER of $10^{-1.45}$, while in the case of the MKF, 200 particles are needed to achieve a comparable level of performance– a significant seven–fold increase in N_p.

It is gratifying to note that for a high number of particles, the SIS–PF, MKF and the ACM–PF all approach the same level of performance. Since the SIS–PF and the MKF approach the optimal Bayesian filter for $N_p \to \infty$, it is seen that the error introduced by (37) has a minimal effect on the performance of the ACM–PF for this example, i.e., in this case, Condition 1 is valid and a moment–matched Gaussian pdf is a good approximation of the mixture.

This result is further supported by Figs 2 and 3, which show estimates of the $p(d_k|\tilde{\boldsymbol{y}}_{1:k})$ as rendered by the SIS–PF, the MKF and the ACM–PF for $N_p = 30$ and $N_p = 2000$ particles, respectively. The time index k is arbitrarily set to $k = 100$ and the true value of the bit is a "1".

For $N_p = 30$, there are noticeable differences between the estimated posteriors, but the bulk of

N_p/Algorithm	PF	MKF	ACM–PF
30	0.44	2.48	8.40
50	0.65	4.01	14.05
100	1.17	7.8	28.98
200	2.22	15.54	57.86
400	4.27	31.08	117.12
1000	11.52	79.33	307.32

Fig. 4. Average CPU time (sec.) to process 1000 DBPSK symbols for various number of particles.

the probabilities are correctly centered on the true value of d_{100}. For a larger number of particles, i.e., $N_p = 2000$, the estimated posteriors are nearly identical to each other. Between Figs 2(c) and 3(c), we observe little change in $\hat{p}(d_{100}|\tilde{\boldsymbol{y}}_{1:100})$ as rendered by the ACM–PF. For both a small and a large number of particles, the ACM–PF in this example provides a reasonable approximation of $p(d_{100}|\tilde{\boldsymbol{y}}_{1:100})$. This is unlike the MKF and the SIS–PF, which for $N_p = 30$, admits a poorer approximation of $p(d_{100}|\tilde{\boldsymbol{y}}_{1:100})$. For the large N_p case, the fact these distributions are similar is again due to Condition 1 being valid in this case, and illustrates that the error introduced by the approximation (37) is negligible.

Finally, let us consider computational complexity of the MKF and the ACM–PF. For non–optimized algorithms implemented in MATLAB on a standard 2.4 GHz PC, the average CPU time to process 1000 DBPSK symbols for various number of particles is summarized in Fig. 4. It is shown that the computational complexity of the ACM–PF is higher than that of the MKF for the same number of particles. However, although the MKF needs less time to process 1000 symbols, the ACM–PF performs more efficiently than the MKF, i.e., in order for the MKF to achieve a BER that is comparable to those of the ACM–PF, the MKF cannot be implemented with less than 200 particles. Since the ACM–PF with 30 particles is computationally cheaper than the MKF with 200 particles, it can be concluded that the ACM–PF is a viable alternative to the MKF. We can construct a similar argument between the ACM–PF and the standard SIS–PF, i.e., in order for the SIS–PF to achieve a BER that is comparable to those of the ACM–PF, the SIS–PF cannot be implemented with less than 1000 particles.

2) Example 2: We consider the problem of tracking a maneuvering target in the presence of glint noise. The location of the sensor is $(x_s, y_s) = (0, 0)$. The state is given by $\boldsymbol{x}_k = [d_{x,k} \;\; \dot{d}_{x,k} \;\; d_{y,k} \;\; \dot{d}_{y,k}]^T$ where $d_{x,k}$ and $d_{y,k}$ are respectively, the position of the target in the x–direction and y–direction. An observation consists of bearing and range measurements. Therefore, the corresponding DSSM is given by

$$\mathbf{x}_k = A(r_k)\mathbf{x}_{k-1} + \mathbf{w}_k \tag{63}$$

$$r_k \sim p(r_k|r_{k-1}) \tag{64}$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k) + \mathbf{e}_k \tag{65}$$

where

$$h_k(\mathbf{x}_k) = \begin{bmatrix} \tan^{-1}\left(\dfrac{d_{y,k} - y_S}{d_{x,k} - x_S}\right) \\ \sqrt{(d_{x,k} - x_S)^2 + (d_{y,k} - y_S)^2} \end{bmatrix} \tag{66}$$

and $\boldsymbol{w}_k$ is the sequence of zero–mean white Gaussian process noise with covariance

$$\boldsymbol{Q}_k = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} \sigma_v^2. \tag{67}$$

For this experiment, $\sigma_v = 10$ and $T = 1$. The current maneuver is determined by a three state Markov process r_t; the associated transition probabilities $p_{m,n} = p(r_k = n|r_{k-1} = m)$ correspond to a sojourn time of 5 seconds. At any time t, either

$$\mathbf{A}(1) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{68}$$

$A(2) = \mathbf{A}_{ct}(0.1)$ or $A(3) = \mathbf{A}_{ct}(-0.1)$ where

$$\mathbf{A}_{ct}(\Omega) = \begin{bmatrix} 1 & \frac{\sin \Omega T}{\Omega} & 0 & -\frac{1-\cos \Omega T}{\Omega} \\ 0 & \cos \Omega T & 0 & -\sin \Omega T \\ 0 & \frac{1-\cos \Omega T}{\Omega} & 1 & \frac{\sin \Omega T}{\Omega} \\ 0 & \sin \Omega T & 0 & \cos \Omega T \end{bmatrix}. \tag{69}$$

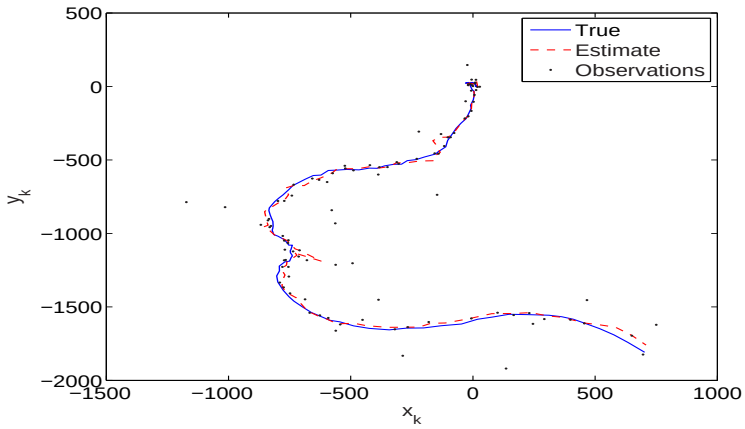


Fig. 5. True and estimated trajectory of ACM–PF for $M = 100$ observations, $N_p = 25$ particles and the prior was used for the importance distribution.

N_p/Algorithm	EKF	MKF	ACM–PF
25	98	75	22
50	98	35	13
100	98	9	2
200	98	2	2
400	98	2	1
1000	98	0	0

Fig. 6. Comparison of the EKF, the MKF, and the ACM–PF in terms of the number of times a target was lost, i.e., $|d_{x,k} - \hat{d}_{x,k}| > 500$ or $|d_{y,k} - \hat{d}_{y,k}| > 500$.

Where $A(1)$ corresponds to a constant trajectory, and $A(1)$ and A(2) correspond to a coordinated right– and left–hand turn, respectively. To model glint noise, we make the assumption that [30], [32]

$$p(\boldsymbol{e}_k) \quad = \quad (1-\epsilon)\mathcal{N}(\boldsymbol{e}_k; \boldsymbol{0}, \boldsymbol{R}) + \epsilon\mathcal{N}(\boldsymbol{e}_k; \boldsymbol{0}, \kappa\boldsymbol{R}) \tag{70}$$

where $\epsilon = 0.2$, $\kappa = 100$ and

$$\boldsymbol{R} = \mathrm{diag}(\sqrt{(3)}[10 \;\; 0.1]^{\mathrm{T}}). \tag{71}$$

Note that a conventional particle filter implementation is not appropriate for this example. This

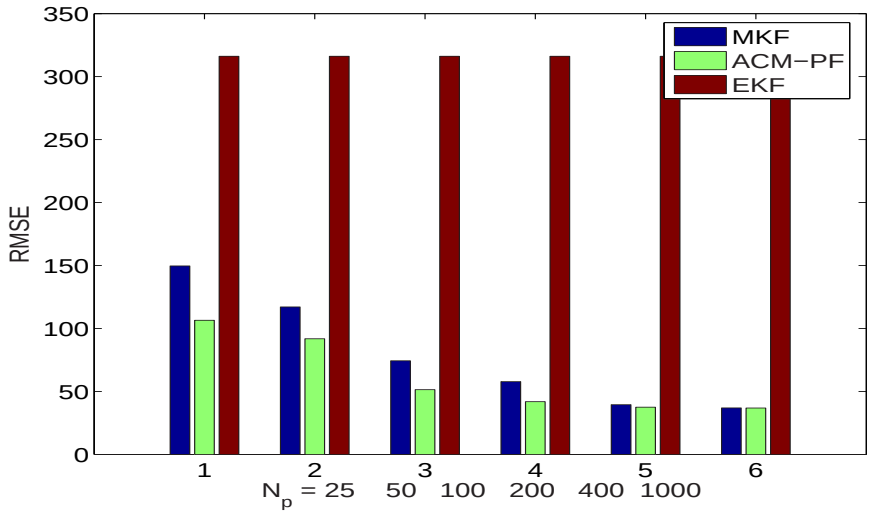


Fig. 7. RMSE after removing of the lost tracks for each filter. The prior was used as the importance distribution with $N_p = 25, 50, 100, 200, 400$ and 1000. For each value of N_p, the left–hand bar represents the MKF, the center bar the ACM–PF, and the right–hand bar shows results for the EKF.

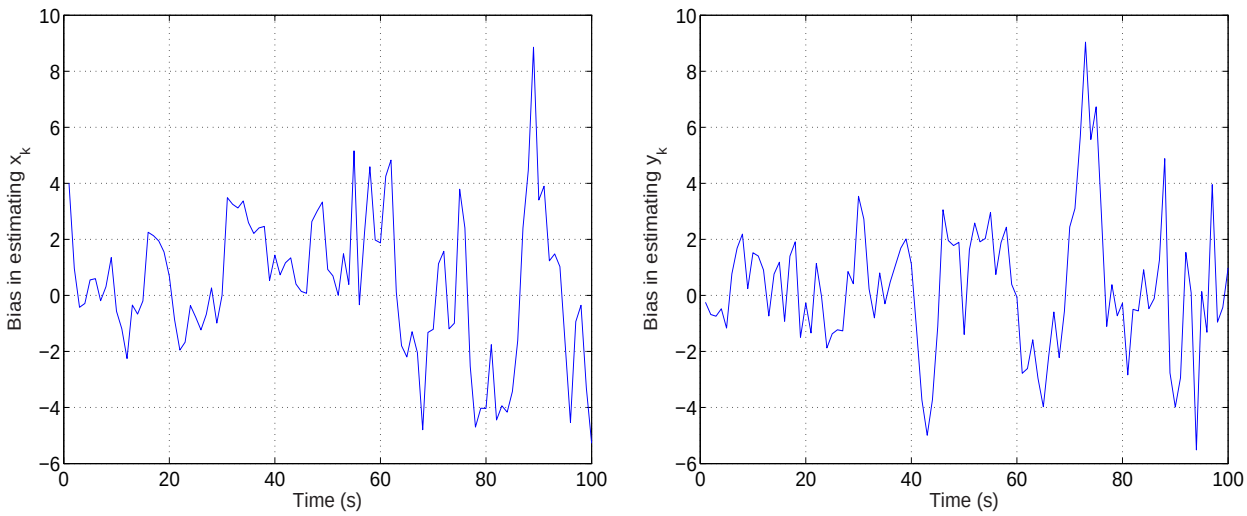
is because the state equations change structure for each type of maneuver under consideration. This renders the implementation of the PF to be very awkward. Therefore in the simulation results, we compare the performance of the MKF and the ACM-PF with that of the extended Kalman filter (EKF).

For the initialization of the target trajectories, we assume that r_0 is a discrete uniform random variable and that $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{x}_0; \mathbf{0}, \boldsymbol{P}_0)$ where $\boldsymbol{P}_0 = \mathrm{diag}([250 \;\; 16 \;\; 250 \;\; 16]^{\mathrm{T}})$.

Now, observe that (65) is a nonlinear measurement equation. Before we can apply the MKF or the ACM–PF we must first linearize $h_k(\mathbf{x}_k)$ about $\mathbf{x}_{k|k-1}$. The quality of the estimates depend on the accuracy of the linearization. As we shall see, the ACM–PF accurately tracks the target. Thus, in conjunction with appropriate linearizations, we can extend the applicability of the ACM–PF to even more general DSSM's. Some experimental results follow.

In Fig. 5, we show a typical and estimated trajectory of $(d_{x,k}, d_{y,k})$ for $M = 100$ observations with the ACM–PF. Clearly, the ACM–PF faithfully tracks the true trajectory of $(d_{x,k}, d_{y,k})$. Here, the prior was used as the importance distribution.

In Fig. 6, we provide a comparison between the MKF and the ACM–PF, in terms of the number of times that a target was lost. For comparison, we also implemented a EKF that has



(a) Average bias for $d_{x,k}$ for $M = 100$ observations.

(b) Average bias for $d_{y,k}$ for $M = 100$ observations.

Fig. 8. Average bias in the estimate of the x and y coordinates for $N_p = 400$, $M = 100$, and $L = 100$ Monte Carlo runs.

perfect knowledge of the maneuver, and the possible occurrence of an impulse at any time step k. To gather statistically meaningful results, we carried out $L = 100$ Monte Carlo (MC) simulations and deemed a target lost if $|d_{x,k} - \hat{d}_{x,k}| > 500$ or $|d_{y,k} - \hat{d}_{y,k}| > 500$. Unlike the EKF or the MKF using a small number of particles, i.e. $N_p \leq 100$, the ACM–PF loses track of its target for a smaller number of runs.

Now, we will compute the root mean square error (RMSE) of each algorithm, after removing the lost tracks. More precisely,

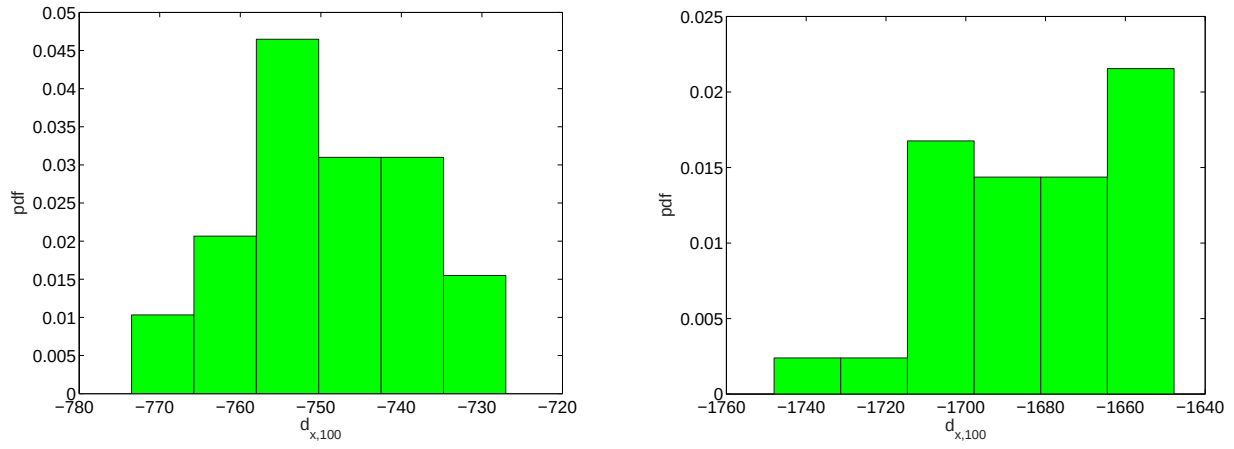
$$RMSE^2 = \frac{1}{ML} \sum_{k=1}^{M} \sum_{i=1}^{L^*} \left((d^i_{x,k} - \hat{d}^i_{x,k})^2 + (d^i_{y,k} - \hat{d}^i_{y,k})^2 \right) \tag{72}$$

where $\hat{d}^i_{x,k}$ and $\hat{d}^i_{y,k}$ are, respectively, the estimate of $d^i_{x,k}$ and $d^i_{y,k}$ for the i-th MC simulation. Fig 7 summarizes the results. Clearly, the EKF performs poorest. Between the MKF and the ACM–PF, the latter outperforms the former. For the considered scenario, the ACM–PF gives the smallest RMSE in this case.

It is also interesting to calculate the bias in the estimates as obtained from the ACM–PF. Fig 8(a) and 8(b) shows the average bias for the x and y coordinate at each time step for $M = 100$ observations based on 100 MC runs. As indicated in the figures, the bias in both coordinates is close to zero for the majority of the time.

Figures 6 and 7 indicate that as N_p becomes large, the performance of the MKF and the ACM–PF become very close to each other. Again, as in the first example, we see that the error introduced by (37) in this case is minimal. From Fig. 8, we can also note that the bias introduced by (37) is relatively small.

To further demonstrate the effect of the ACM approximation (37) on this example and to shed further light on the behaviour of the ACM-PF method, we show approximations of the posterior distributions $p(d_{x,100}|y_{1:100})$ rendered by the MKF and the ACM-PF, for the case of $N_p = 30$ particles in Figure 9. The time index $k = 100$ was chosen arbitrarily. The true value for $d_{x,100} = -1670.6$m. For this small number of particles, we can see that the distribution obtained by the MKF is badly biased away from the true value, whereas that yielded by the ACM-PF is roughly centered on the true value. This improved performance of the ACM-PF may be attributed to the fact we need not sample the indicator variables I_k whcih are required for the MKF. The behaviour of the distributions $p(d_{y,100}|y_{1:100})$ (not shown) corresponds to that of $p(d_{x,100}|y_{1:100})$ for the two methods.

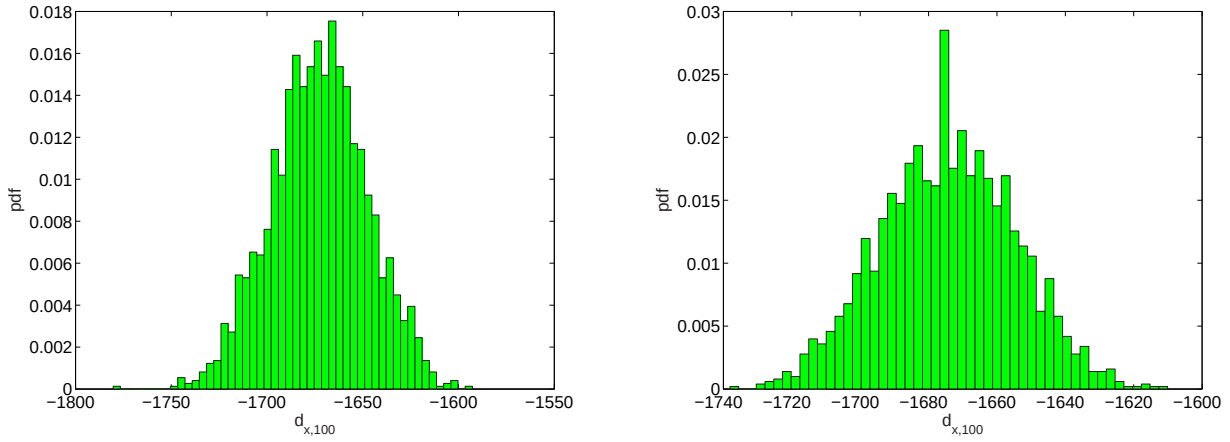


(a) MKF (b) ACMPF

Fig. 9. Estimate of $p(d_{x,100}|y_{1:100})$ via MKF and the ACM-PF. Each filter uses $N_p = 30$. The true value is $d_{x,100} = -1670.6$m.

In Figure 10, we show the same scenario for the large number of particles case, $N_p = 2000$. Here, we see the distributions produced by the two methods are very close to each other. This indicates that for this specific case, the ACM-PF behaves like the optimal Bayesian filter as $N_p \rightarrow \infty$. Again, the distributions for $p(d_{y,100}|y_{1:100})$ which are also not shown indicate corresponding

behaviour. As in the first example, the favourable performance of the ACM-PF is due to Condition 1 being satisfied.



(a) MKF (b) ACMPF

Fig. 10. Estimate of $p(d_{x,100}|y_{1:100})$ via MKF and the ACM-PF. Each filter uses $N_p = 2000$. The true value is $d_{x,100} = -1670.6$m.

Finally, let us consider computational complexity of the MKF and the ACM–PF. For non–optimized algorithms implemented in MATLAB on a standard 3.0 GHz PC, the average CPU time to process $M = 100$ observations for various number of particles is summarized in Fig. 11. For the same number of particles, the computational complexity of the ACM–PF is higher than that of the MKF. However, although the MKF needs less time to process $M = 100$ observations, the ACM–PF performs more efficiently than the MKF. Indeed, even with $N_p = 400$ particles, the MKF yields a RMSE that is 20% larger than that of the ACM–PF. Also, for $N_p = 25, 50, 100$, recall that the ACM–PF outperformed the MKF in terms of the RMSE and the number of times a target was lost. For the tracking of a maneuvering target in the presence of glint noise, the ACM–PF renders an effective alternative to the EKF and the well known MKF.

IV. Conclusion

In this paper, we have proposed a novel filter for a class of DSSM's whose parameters evolve in time according to some known transition distribution and whose measurement noise is distributed according to a mixture of Gaussians. The proposed method called the ACM–PF is an efficient combination of the ACM filter and the SIS–PF. Extensive simulations showed that the ACM–PF

N_p/Algorithm	MKF	ACM–PF
25	1.7575	2.4688
50	3.4720	4.8807
100	6.9057	9.6800
200	13.8125	19.3148
400	27.8249	38.6258
1000	71.0164	96.7361

Fig. 11. Average CPU time (sec.) to process $M = 100$ observations for various number of particles.

successfully addresses the blind detection problem for an impulsive flat fading channel and the tracking of a maneuvering target in the presence of glint noise. For computationally constrained signal processing, we have demonstrated that the ACM–PF outperforms other state–of–the–art filtering algorithms, namely, the SIS–PF and the well known MKF for the simulation examples considered. The success of the proposed method in these cases depends on the predicted posterior pdf which is modelled as a mixture of Gaussians being approximated by a single Gaussian distribution.

V. Acknowledgements

This paper was funded in large part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors would like to recognize the contributions of the reviewers, whose comments and suggestions have resulted in a considerably improved version of the paper.

References

[1] R. G. Albano, L. Franchina, and S. A. Kosmopoulos, "Bit error performance evaluation of double-differential QPSK in faded channels characterized by Gaussian plus impulsive noise and doppler effects," *IEEE Trans. Veh. Technol.*, vol. 49, no. 1, pp. 148–158, Jan. 2000.

[2] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[3] C. Andrieu and A. Doucet, "Particle filtering for partially observed Gaussian state space models," *J. Royal Statist. Soc. B*, vol. 64, pp. 827–836, 2002.

[4] 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.

[5] R. Chen, X. Wang, and J. Liu, "Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering," *IEEE Trans. Inform. Theory*, vol. 46, no. 6, pp. 2079–2094, 2000.

[6] M. Daly and J. P. Reilly, "Blind Deconvolution using Bayesian Methods with Application to the Dereverberation of Speech," in *Proc. IEEE ICASSP*, May 2004, pp. 1009–1012.

[7] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry*. Meboo Publishing USA, 2005.

[8] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *J. Royal Statistic Soc*, vol. 30, no. B, pp. 1–38, 1977.

[9] P. M. Djuricˋ, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle Filtering," *IEEE Signal Processing Mag.*, vol. 20, no. 5, pp. 19–38, 2003.

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

[11] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, Dec. 2000.

[12] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 613–624, Mar. 2001.

[13] W. Fong, S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing with application to audio signal enhancement," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 438–449, Feb. 2002.

[14] N. Gordon, D. Salmond, and A. Smith, "Novel approach to non-linear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

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

[16] Y. Huang and P. M. Djurić, "A blind particle filtering detector of signals transmitted over flat fading channels," *IEEE Trans. Signal Processing*, vol. 52, no. 7, pp. 1891–1900, 2004.

[17] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, Series D, pp. 35–45, 1960.

[18] C. Komninakis, C. Fragouli, A. H. Sayed, and R. D. Wesel, "Multi-input Multi-output fading channel tracking and equalization using Kalman estimation," *IEEE Trans. Signal Processing*, vol. 50, no. 5, pp. 1065–1076, May 2002.

[19] J. S. Liu, *Monte Carlo Methods in Scientific Computing*. New York: Springer-Verlag, 2001.

[20] J. S. Liu and R. Chen, "Mixture Kalman filters," *J. Roy. Statistic. Soc. B*, vol. 62, pp. 493–508, 2000.

[21] Z. Liu, X. Ma, and G. B. Giannakis, "Space-time coding and Kalman filtering for time-selective fading channels," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 183–186, Feb. 2002.

[22] C. J. Masreliez, "Approximate non-Gaussian filtering with linear state and observation relations," *IEEE Trans. Automat. Contr.*, vol. 20, no. 1, pp. 107–110, Jan. 1975.

[23] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 8, pp. 1344–1348, 2006.

[24] S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian approaches to Gaussian mixture modeling," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 11, pp. 1133–1142, 1998.

[25] T. Schon, F. Gustafsson, and P. J. Nordlund, "Marginalized Particle Filters for Mixed Linear/Nonlinear State–Space Models," *IEEE Trans. Signal Processing*, vol. 53, no. 7, pp. 2279–2289, 2005.

[26] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, pp. 465–479, 1971.

[27] J. J. Verbeek, N. Vlassis, and B. Krose, “Efficient greedy learning of Gaussian mixture models,” *Neural Comput*, vol. 15, pp. 469–485, 2003.

[28] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill, “Particle methods for Bayesian modeling and enhancement of speech signals,” *IEEE Trans. Speech Audio Processing*, vol. 10, no. 3, pp. 173–185, Mar. 2002.

[29] R. Vijayan and H. V. Poor, “Nonlinear techniques for interference suppression in spread-spectrum systems,” *IEEE Trans. Commun.*, vol. 38, no. 7, pp. 1060–1065, 1990.

[30] X. Wang and R. Chen, “Adaptive Bayesian multiuser detection for synchronous CDMA with Gaussian and impulsive noise,” *IEEE Trans. Signal Processing*, vol. 47, no. 7, pp. 2013–2028, 2000.

[31] X. Wang and H. V. Poor, “Joint channel estimation and symbol detection in Rayleigh flat-fading channels with impulsive noise,” *IEEE Commun. Lett.*, vol. 1, no. 1, pp. 19–21, Jan. 1997.

[32] W. R. Wu, “Maximum likelihood identification of glint noise,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 1, pp. 41–51, Jan. 1996.

[33] D. Yee, J. P. Reilly, and T. Kirubarajan, “Approximate Conditional Mean Particle Filter,” in *Proc. IEEE Workshop on Statistical Signal Processing*, 2005.

Table 1: Approximate Conditional Mean Particle filter (ACM–PF)

1) Initialization: For $i = 1, \ldots, N_p$, we initialize the particles, $\boldsymbol{x}_0^{2,(i)} \sim p(\boldsymbol{x}_0^2)$, $\boldsymbol{x}_{0|0}^{1,(i)} = \hat{\boldsymbol{x}}_0^1$, $\boldsymbol{P}_{0|0}^{1,(i)} = \hat{\boldsymbol{P}}_0^1$ and set $w_0^{(i)} = \frac{1}{N_p}$.
2) New particles: For $i = 1, \ldots, N_p$, set $\tilde{\boldsymbol{x}}_{k-n:k-1}^{2,(i)} = \boldsymbol{x}_{k-n:k-1}^{2,(i)}$, $\tilde{\boldsymbol{x}}_{k-1|k-1}^{1,(i)} = \boldsymbol{x}_{k-1|k-1}^{1,(i)}$, $\tilde{\boldsymbol{P}}_{k-1|k-1}^{1,(i)} = \boldsymbol{P}_{k-1|k-1}^{1,(i)}$.
   - Proposals: Draw $\tilde{\boldsymbol{x}}_k^{2,(i)} \sim q(\boldsymbol{x}_k^2 | \tilde{\boldsymbol{x}}_{1:k-1}^{2,(i)}, \boldsymbol{y}_{1:k})$.
   - ACM prediction: Compute $\tilde{\boldsymbol{x}}_{k|k-1}^{1,(i)}$, $\tilde{\boldsymbol{P}}_{k|k-1}^{1,(i)}$ using (43), and (44), respectively.
   - ACM update: Compute $\tilde{\boldsymbol{x}}_{k|k}^{1,(i)}$, $\tilde{\boldsymbol{P}}_{k|k}^{1,(i)}$ using (41), and (42), respectively.
3) Calculate Importance Weights: For $i = 1, \ldots, N_p$, evaluate the importance weights up to a normalizing constant

$$w_k^{(i)} \propto \frac{p(\boldsymbol{y}_k | \boldsymbol{x}_{1:k}^{2,(i)}, \boldsymbol{y}_{1:k-1}) p(\boldsymbol{x}_k^{2,(i)} | \boldsymbol{x}_{k-n:k-1}^{2,(i)})}{q(\boldsymbol{x}_k^{2,(i)} | \boldsymbol{x}_{1:k-1}^{2,(i)}, \boldsymbol{y}_k)} w_{k-1}^{(i)}$$

   and normalize importance weights to yield $\tilde{w}_k^{(i)}$.
4) If $\widehat{N}_{eff} < N_{th}$,
   - Resample $\{\tilde{\boldsymbol{x}}_{k-n+1:k}^{2,(i)}\}_{i=1}^{N_p}$, $\{\tilde{\boldsymbol{x}}_{k|k}^{1,(i)}\}_{i=1}^{N_p}$, $\{\tilde{\boldsymbol{P}}_{k|k}^{1,(i)}\}_{i=1}^{N_p}$ w.r.t importance weights to obtain $\{\boldsymbol{x}_{k-n+1:k}^{2,(i)}\}_{i=1}^{N_p}$, $\{\boldsymbol{x}_{k|k}^{1,(i)}\}_{i=1}^{N_p}$, $\{\boldsymbol{P}_{k|k}^{1,(i)}\}_{i=1}^{N_p}$ and set $w_k^{(i)} = \frac{1}{N_p}$ for $i = 1, \ldots, N_p$.

   otherwise
   - Set $\tilde{\boldsymbol{x}}_{k-n+1:k}^{2,(i)} = \boldsymbol{x}_{k-n+1:k}^{2,(i)}$, $\tilde{\boldsymbol{x}}_{k|k}^{1,(i)} = \boldsymbol{x}_{k|k}^{1,(i)}$, and $\tilde{\boldsymbol{P}}_{k|k}^{1,(i)} = \boldsymbol{P}_{k|k}^{1,(i)}$ for $i = 1, \ldots, N_p$.
5) Estimates: Compute $\widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$, $\widehat{cov}_{p(\boldsymbol{x}_k^1|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^1]$, $\widehat{\mathbb{E}}_{p(\boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^2]$, and $\widehat{cov}_{p(\boldsymbol{x}_k^2|\boldsymbol{y}_{1:k})}[\boldsymbol{x}_k^2]$ using (34), (35), (19), and (20), respectively.
6) Set $k = k + 1$, and go back to step 2.

Remark 1: We point out that the ACM filter (41)–(44) has a structure that is similar to the KF (24)–(27). Indeed, it can be shown that the ACM filter reduces to the KF when $\boldsymbol{e}_k$ is Gaussian distributed. Therefore, it follows that the ACM–PF reduces to the MKF if $\boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{e}_k; \overline{\boldsymbol{e}}_{\boldsymbol{k}}, \boldsymbol{R}_k)$ where $\overline{\boldsymbol{e}}_{\boldsymbol{k}}$ and $\boldsymbol{R}_k$ are, respectively, the mean and covariance of $\boldsymbol{e}_k$.

APPENDIX

The forthcoming derivation of $g_k(\boldsymbol{x})$ and $G_k(\boldsymbol{x})$ will make use of the following lemma [7]:

Lemma 1: For dimensionally conforming matrix valued functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$, it can be shown that

$$\nabla_{\boldsymbol{x}}\left[f^T(\boldsymbol{x})g(\boldsymbol{x})\right] = \nabla_{\boldsymbol{x}}\left[f(\boldsymbol{x})\right]g(\boldsymbol{x}) + \nabla_{\boldsymbol{x}}\left[g(\boldsymbol{x})\right]f(\boldsymbol{x}). \tag{73}$$

where the notation $\nabla_{\boldsymbol{x}}$ denotes the gradient operator with respect to $\boldsymbol{x}$.

For ease of notation, let us derive $g_k(\boldsymbol{x})$ and $G_k(\boldsymbol{x})$ for a general GMM $p(\boldsymbol{x})$.

Proof: Given

$$p(\boldsymbol{x}) \;=\; \sum_{j=1}^{N} p_j \mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)}), \tag{74}$$

it can be shown that $g(\boldsymbol{x}) = -p(\boldsymbol{x})^{-1}\nabla_{\boldsymbol{x}} p(\boldsymbol{x})$ equals

$$g(\boldsymbol{x}) \;=\; p(\boldsymbol{x})^{-1}\sum_{j=1}^{N} p_j \mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})[\boldsymbol{P}^{(j)}]^{-1}(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)}). \tag{75}$$

In order to derive $G(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} g(\boldsymbol{x})^T$, we will repeatedly make use of (73). To begin, let us expand $G(\boldsymbol{x})$ as follows:

$$G(\boldsymbol{x}) \;=\; \sum_{j=1}^{N} p_j \nabla_{\boldsymbol{x}}\left[p(\boldsymbol{x})^{-1}\mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})^T[\boldsymbol{P}^{(j)}]^{-1}\right]. \tag{76}$$

Now, let $f(\boldsymbol{x}) = p(\boldsymbol{x})^{-1}\mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})$ and $g(\boldsymbol{x}) = (\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})^T[\boldsymbol{P}^{(j)}]^{-1}$ in (73) so that

$$\begin{aligned} G(\boldsymbol{x}) = \sum_{j=1}^{N} p_j \Bigg(& \nabla_{\boldsymbol{x}}\left[p(\boldsymbol{x})^{-1}\mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})\right](\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})^T[\boldsymbol{P}^{(j)}]^{-1} \\ & + \nabla_{\boldsymbol{x}}\left[(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})^T[\boldsymbol{P}^{(j)}]^{-1}\right] p(\boldsymbol{x})^{-1}\mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})\Bigg). \end{aligned} \tag{77}$$

Viewing (77), it is clear that we must evaluate $\nabla_{\boldsymbol{x}}\left[p(\boldsymbol{x})^{-1}\mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})\right]$ and $\nabla_{\boldsymbol{x}}\left[(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})^T [\boldsymbol{P}^{(j)}]^{-1}\right]$. By using (73) again, but with $f(\boldsymbol{x}) = p(\boldsymbol{x})^{-1}$ and $g(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})$, it can be shown that

$$\nabla_{\boldsymbol{x}}\left[p(\boldsymbol{x})^{-1}\mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})\right] \;=\; \frac{\mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})}{p(\boldsymbol{x})}\left(g(\boldsymbol{x}) - [\boldsymbol{P}^{(j)}]^{-1}(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})\right) \tag{78}$$

while

$$\nabla_{\boldsymbol{x}}\left[(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})^T[\boldsymbol{P}^{(j)}]^{-1}\right] \;=\; [\boldsymbol{P}^{(j)}]^{-1}. \tag{79}$$

Finally, by substituting (78) and (79) into (77), we obtain after several mathematical manipulations of the resulting expression:

$$G(\boldsymbol{x}) = \sum_{j=1}^{N} p_j \frac{\mathcal{N}(\boldsymbol{x}; \overline{\boldsymbol{x}}^{(j)}, \boldsymbol{P}^{(j)})[\boldsymbol{P}^{(j)}]^{-1}}{p(\boldsymbol{x})} \times \left(\boldsymbol{I}_{n\times n} + \boldsymbol{P}^{(j)} g(\boldsymbol{x})(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})^T [\boldsymbol{P}^{(j)}]^{-1} - (\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})(\boldsymbol{x} - \overline{\boldsymbol{x}}^{(j)})^T [\boldsymbol{P}^{(j)}]^{-1} \right). \quad (80)$$