Gaussian Particle Filtering

Jayesh H. Kotecha and Petar M. Djurić, *Senior Member, IEEE*

***Abstract*—Sequential Bayesian estimation for *nonlinear* dynamic state-space models involves recursive estimation of filtering and predictive distributions of unobserved time varying signals based on noisy observations. This paper introduces a new filter called the Gaussian particle filter[1]. It is based on the particle filtering concept, and it approximates the posterior distributions by single Gaussians, similar to Gaussian filters like the extended Kalman filter and its variants. It is shown that under the Gaussianity assumption, the Gaussian particle filter is asymptotically optimal in the number of particles and, hence, has much-improved performance and versatility over other Gaussian filters, especially when nontrivial nonlinearities are present. Simulation results are presented to demonstrate the versatility and improved performance of the Gaussian particle filter over conventional Gaussian filters and the lower complexity than known particle filters. The use of the Gaussian particle filter as a building block of more complex filters is addressed in a companion paper.**

***Index Terms*—Dynamic state space models, extended Kalman filter, Gaussian mixture, Gaussian mixture filter, Gaussian particle filter, Gaussian sum filter, Gaussian sum particle filter, Monte Carlo filters, nonlinear non-Gaussian stochastic systems, particle filters, sequential Bayesian estimation, sequential sampling methods, unscented Kalman filter.**

I. Introduction

Nonlinear filtering problems arise in many fields including statistical signal processing, economics, statistics, biostatistics, and engineering such as communications, radar tracking, sonar ranging, target tracking, and satellite navigation. The problem consists of estimating a possibly dynamic state of a nonlinear stochastic system, based on a set of noisy observations. Many of these problems can be written in the form of the so-called dynamic state space (DSS) model [3]. The DSS model represents the time-varying dynamics of an unobserved state variable $\mathbf{x}_n$, as the distribution $p(\mathbf{x}_n \mid \mathbf{x}_{n-1})$, where n indicates time (or any other physical parameter). The observations $\mathbf{y}_n$ in the application are usually noisy and distorted versions of $\mathbf{x}_n$. The distribution $p(\mathbf{y}_n \mid \mathbf{x}_n)$ represents the observation equation conditioned on the unknown state variable $\mathbf{x}_n$, which is to be estimated. Alternatively, the model can be written as

$$\begin{aligned} \mathbf{x}_n &= \mathbf{f}(\mathbf{x}_{n-1}, \mathbf{u}_n) \quad \text{(process equation)} \\ \mathbf{y}_n &= \mathbf{h}(\mathbf{x}_n, \mathbf{v}_n) \quad \text{(observation equation)} \end{aligned} \tag{1}$$

where $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are some known functions, and $\mathbf{u}_n$ and $\mathbf{v}_n$ are random noise vectors of given distributions. The process equation represents a system evolving with time n, where the system is represented by the hidden state $\mathbf{x}_n$, and the prior knowledge of the initial state is given by the probability distribution $p(\mathbf{x}_0)$. Our aim is to learn more about the unknown state variables, given the observations as time evolves.

We denote by $\mathbf{x}_{0:n}$ and $\mathbf{y}_{0:n}$ the signal and observations up to time n, respectively, i.e., $\mathbf{x}_{0:n} \equiv \{\mathbf{x}_0, \ldots, \mathbf{x}_n\}$ and $\mathbf{y}_{0:n} \equiv \{\mathbf{y}_0, \ldots, \mathbf{y}_n\}$. In a Bayesian context, our aim is to estimate *recursively in time*

- the filtering distribution $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ at time n given all the observations up to time n;
- the predictive distribution $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})$ at time n given all the observations up to time n.

From these distributions, an estimate of the state can be determined for any performance criterion suggested for the problem. The filtering distribution or the marginal posterior of the state at time n can be written as

$$p(\mathbf{x}_n \mid \mathbf{y}_{0:n}) = C_n p(\mathbf{x}_n \mid \mathbf{y}_{0:n-1})\, p(\mathbf{y}_n \mid \mathbf{x}_n) \tag{2}$$

where C_n is the normalizing constant given by

$$C_n = \left(\int p(\mathbf{x}_n \mid \mathbf{y}_{0:n-1})\, p(\mathbf{y}_n \mid \mathbf{x}_n)\, d\mathbf{x}_n \right)^{-1}.$$

Furthermore, the predictive distribution can be expressed as

$$p(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n}) = \int p(\mathbf{x}_{n+1} \mid \mathbf{x}_n)\, p(\mathbf{x}_n \mid \mathbf{y}_{0:n})\, d\mathbf{x}_n. \tag{3}$$

When the model is linear with Gaussian noise and the prior knowledge about $\mathbf{x}_0$ given by $p(\mathbf{x}_0)$ is Gaussian, the filtering and predictive distributions are Gaussian, and the Kalman filter provides the mean and covariance sequentially, which is the optimal Bayesian solution [4]. However, for most nonlinear models and non-Gaussian noise problems, closed-form analytic expression for the posterior distributions do not exist in general. Numerical solutions often require high-dimensional integrations that are not practical to implement. As a result, several approximations that are more tractable have been proposed.

A class of filters called *Gaussian filters* provide Gaussian approximations to the filtering and predictive distributions, examples of which include the extended Kalman filter (EKF) and its variations [4]–[8]. Equivalently, each recursive update propagates only the mean and covariance of the densities. The justifications are that under this assumption, only the mean and covariance need to be tracked and that given just the mean and covariance, the Gaussian maximizes the entropy of the random variable, i.e., it is the least informative distribution. The applicability of Gaussian filters to nonlinear problems depends on the nature of the nonlinearities and has to be established on a

Manuscript received July 5, 2001; revised March 4, 2003. This work was supported by the National Science Foundation under Awards CCR-0082607 and CCR-9903120. The associate editor coordinating the review of this paper and approving it for publication was Prof. Zhi Ding.

J. H. Kotecha is with the Department of Electrical and Computer Engineering, University of Wisconsin at Madison, Madison, WI, 53705 USA (e-mail: jkotecha@ece.wisc.edu).

P. M. Djurić is with the Department of Electrical and Computer Engineering, State University of New York at Stony Brook, Stony Brook, NY 11794 USA (e-mail: djuric@ece.sunysb.edu).

Digital Object Identifier 10.1109/TSP.2003.816758

[1] Part of this work has appeared in [1] and [2].

case-by-case basis. For some general guidelines, see, for example, [5] and [9]. Other approaches that propagate approximations of the first two moments of the densities include [10]–[12]. These filters, including the EKF, have been successfully implemented in some problems, but in others, they diverge or provide poor approximations. This is especially emphasized when the model is highly nonlinear or when the posterior distributions are multimodal. In such cases, however, significant improvements are possible. Efforts to improve on the EKF have led to the new filters recently, like the unscented Kalman filter (UKF) by Julier *et al.* [13] and similar filters proposed by Ito *et al.* [14], which use deterministic sets of points in the space of the state variable to obtain more accurate approximations to the mean and covariance than the EKF. However, note the following.

- The improvement of these filters over the EKF can be significant, depending on the problem. However, divergence can still occur in some nonlinear problems. An example of such undesirable divergence is shown in the simulations.
- More importantly, in problems where the above filters do not diverge, improvement in the estimates of the mean and covariance are desirable. The aim is not only to minimize the mean square error but also to provide accurate estimates of the covariance that are a measure of the confidence in the estimates.
- Finally, the versatility of the filter can be improved if the restrictive assumption of additive Gaussian noise made in the EKF like filters is removed.

There have also been other attempts to propagate filtering densities, most of them based on Gaussian sum filters (GSFs) [15], where the posterior distributions are approximated as finite Gaussian mixtures (GMs). The GM approximation is generally more accurate, especially for multimodal systems. Other methods evaluate the required densities over grids [7], [16]–[19], but they are computationally intense especially for high-dimensional problems.

Recently, importance sampling-based filters have been used to update the posterior distributions [20]–[23]. There, a distribution is represented by a weighted set of samples (or particles) from the distribution, which are propagated through the dynamic system using importance sampling to sequentially update the posterior distributions. These methods are collectively called sequential importance sampling (SIS) filters, or particle filters, and provide optimal results asymptotically in the number of particles. However, a major disadvantage of particle filters is the computational complexity (in a serial implementation), a large part of which comes from a procedure called resampling. Particle filters are, however, amenable to parallel implementation, which provides possibilities for processing signals sampled at high sampling rates.

In this paper, we introduce a new Gaussian filter called the Gaussian particle filter (GPF). Essentially, the GPF approximates the posterior mean and covariance of the unknown state variable using importance sampling. The proposed filter is analyzed theoretically and studied by computer simulations. Comparisons are made with the EKF, UKF, and the standard sequential importance sampling resampling (SISR) filters. It is important to note that unlike the EKF, the assumption of additive Gaussian noise can be relaxed for the GPF and can, in general, be *non-Gaussian* and *nonadditive*. The GPF has improved performance compared with the EKF and UKF, as demonstrated by simulations. It is shown analytically that the estimates of the unknowns converge asymptotically with probability one to the minimum mean square estimates (given that the Gaussian assumption holds true). The GPF is quite similar to the SIS filter by the fact that importance sampling is used to obtain particles. However, unlike the SIS filters, resampling is not required in the GPF. This results in a reduced complexity of the GPF as compared with the SIS with resampling and is a major advantage.

The GPF only propagates the mean and covariance; however, note that the importance sampling procedure makes it simple to propagate higher moments as well. This gives rise to a new class of filters, where the posterior distributions are approximated by distributions other than Gaussian simply by propagating the required moments (or functions thereof).

An example of the above is given in a companion paper [24], where we introduce three types of Gaussian sum particle filters (GSPFs), which are built from banks of GPFs, and develop a general framework for Bayesian inference for nonlinear and non-Gaussian additive noise DSS models using Gaussian mixtures.

To facilitate readability of the paper, we provide a list of abbreviations used in the sequel.

List of Abbreviations:

BOT	Bearings-only tracking.
DSS	Dynamic state space.
EKF	Extended Kalman filter.
EM	Expectation–maximization.
GS	Gaussian sum.
GM	Gaussian mixture.
GMF	Gaussian mixture filter.
GMM	Gaussian mixture model.
GPF	Gaussian particle filter.
GSF	Gaussian sum filter.
GSPF	Gaussian sum particle filter.
MMSE	Minimum mean square error.
MSE	Mean square error.
SIS	Sequential importance sampling.
SISR	Sequential importance sampling with resampling.
UKF	Unscented Kalman filter.
VLSI	Very large scale integration.

II. GAUSSIAN PARTICLE FILTERING

The GPF approximates the filtering and predictive distributions in (2) and (3) by Gaussian densities using the particle filtering methodology [20], [22], [25]. The basic idea of Monte Carlo methods is to represent a distribution $p(\mathbf{x}_n)$ of a random variable $\mathbf{x}_n$ by a collection of samples (particles) from that distribution. M particles $\mathcal{X} = \{\mathbf{x}_n^{(1)}, \ldots, \mathbf{x}_n^{(M)}\}$ from a so-called importance sampling distribution $\pi(\mathbf{x}_n)$ (which satisfies certain conditions; see [25] for details) are generated. The particles are then weighted as $w^{(j)} = (p(\mathbf{x}_n^{(j)}))/(\pi(\mathbf{x}_n^{(j)}))$. If $W = \{w^{(1)}, \ldots, w^{(M)}\}$, then the set $\{\mathcal{X}, W\}$ represents samples from the posterior distribution $p(\mathbf{x}_n)$. Monte Carlo integration suggests then that the estimate of

$$E_p\left(g\left(\mathbf{x}_n\right)\right) = \int g\left(\mathbf{x}_n\right) p\left(\mathbf{x}_n\right) d\mathbf{x}_n$$

can be computed as

$$\hat{E}_p\left(g\left(\mathbf{x}_n\right)\right) = \frac{\sum_j w^{(j)} g\left(\mathbf{x}_n^{(j)}\right)}{\sum_j w^{(j)}}. \tag{4}$$

Using the Strong Law of Large Numbers, it can be shown that

$$\hat{E}_p\left(g\left(\mathbf{x}_n\right)\right) \longrightarrow E_p\left(g\left(\mathbf{x}_n\right)\right) \tag{5}$$

almost surely as $M \rightarrow \infty$; see, for example, [25]. The posterior distribution can be approximated as

$$p\left(\mathbf{x}_n\right) d\mathbf{x}_n = P\left(d\mathbf{x}_n\right) \approx \frac{\sum_{j=1}^{M} w^{(j)} \delta_{x_n^{(j)}}\left(d\mathbf{x}_n\right)}{\sum_{j=1}^{M} w^{(j)}} \tag{6}$$

where $\delta_{\mathbf{x}_n}(d\mathbf{x}_n)$ is the Dirac delta function.

In the sequel, the density of a Gaussian random variable $\mathbf{x}$ is written as

$$\begin{aligned}\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= (2\pi)^{-m/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)\end{aligned}$$

where the m-dimensional vector $\boldsymbol{\mu}$ is the mean, and the covariance is the positive definite matrix $\boldsymbol{\Sigma}$. Assume that at time $n = 1$ we have $p(\mathbf{x}_1 \mid \mathbf{y}_0) = \mathcal{N}(\mathbf{x}_1; \bar{\boldsymbol{\mu}}_1, \bar{\boldsymbol{\Sigma}}_1)$, where $\bar{\boldsymbol{\mu}}_1$ and $\bar{\boldsymbol{\Sigma}}_1$ are chosen based on prior information. As new measurements are received, measurement and time updates are performed to obtain the filtering and predictive distributions as discussed in the following sections.

A. Measurement Update

After receiving the nth observation $\mathbf{y}_n$, the filtering distribution in (2) is given by

$$\begin{aligned}p\left(\mathbf{x}_n \mid \mathbf{y}_{0:n}\right) =& C_n p\left(\mathbf{y}_n \mid \mathbf{x}_n\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{0:n-1}\right) \\ \approx& C_n p\left(\mathbf{y}_n \mid \mathbf{x}_n\right) \mathcal{N}\left(\mathbf{x}_n; \bar{\boldsymbol{\mu}}_n, \bar{\boldsymbol{\Sigma}}_n\right).\end{aligned} \tag{7}$$

The GPF measurement update step approximates the above density as a Gaussian, i.e.,

$$\hat{p}(\mathbf{x}_n | \mathbf{y}_{0:n}) = \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n). \tag{8}$$

In general, analytical expressions for the mean $\boldsymbol{\mu}_n$ and covariance $\boldsymbol{\Sigma}_n$ of $p(\mathbf{x}_n \mid \mathbf{y}_{0:n})$ are not available. However, for the GPF update, Monte Carlo estimates of $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$ can be computed from the samples $\mathbf{x}_n^{(i)}$ and their weights, where the samples are obtained from an importance sampling function $\pi(\mathbf{x}_n \mid \mathbf{y}_{0:n})$. This allows for the measurement update algorithm in Table I.

The updated filtering distribution is now approximated as $p(\mathbf{x}_n \mid \mathbf{y}_{0:n}) \approx \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$.

TABLE I
GPF—MEASUREMENT UPDATE ALGORITHM

GPF - Measurement update algorithm.

1. Draw samples from the importance function $\pi(\mathbf{x}_n|\mathbf{y}_{0:n})$ and denote them as $\{\mathbf{x}_n^{(j)}\}_{j=1}^{M}$.
2. Obtain the respective weights by

$$\bar{w}_n^{(j)} = \frac{p(\mathbf{y}_n|\mathbf{x}_n^{(j)})\mathcal{N}(\mathbf{x}_n = \mathbf{x}_n^{(j)}; \bar{\boldsymbol{\mu}}_n, \bar{\boldsymbol{\Sigma}}_n)}{\pi(\mathbf{x}_n^{(j)}|\mathbf{y}_{0:n})}. \tag{9}$$

3. Normalize the weights as

$$w_n^{(j)} = \bar{w}_n^{(j)} / \sum_{j=1}^{M} \bar{w}_n^{(j)}. \tag{10}$$

4. Estimate the mean and covariance by

$$\begin{aligned}\boldsymbol{\mu}_n &= \textstyle\sum_{j=1}^{M} w_n^{(j)} \mathbf{x}_n^{(j)} \\ \boldsymbol{\Sigma}_n &= \textstyle\sum_{j=1}^{M} w_n^{(j)} (\boldsymbol{\mu}_n \quad \mathbf{x}_n^{(j)})(\boldsymbol{\mu}_n \quad \mathbf{x}_n^{(j)})^H.\end{aligned} \tag{11}$$

GPF - Time update algorithm.

1. Draw samples from $\mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ and denote them as $\{\mathbf{x}_n^{(j)}\}_{j=1}^{M}$.
2. For $j = 1, \ldots, M$, sample from $p(\mathbf{x}_{n+1}|\mathbf{x}_n = \mathbf{x}_n^{(j)})$ to obtain $\{\mathbf{x}_{n+1}^{(j)}\}_{j=1}^{M}$.
3. Compute the mean $\bar{\boldsymbol{\mu}}_{n+1}$ and covariance $\bar{\boldsymbol{\Sigma}}_{n+1}$ as

$$\begin{aligned}\bar{\boldsymbol{\mu}}_{n+1} &= \tfrac{1}{M} \textstyle\sum_{j=1}^{M} \mathbf{x}_{n+1}^{(j)} \\ \bar{\boldsymbol{\Sigma}}_{n+1} &= \tfrac{1}{M} \textstyle\sum_{j=1}^{M} (\bar{\boldsymbol{\mu}}_{n+1} \quad \mathbf{x}_{n+1}^{(j)})(\bar{\boldsymbol{\mu}}_{n+1} \quad \mathbf{x}_{n+1}^{(j)})^H.\end{aligned} \tag{12}$$

GPF - Time update algorithm - Alternate.

1. Retain weighted samples of $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ obtained in the measurement update and denote them as $\{\mathbf{x}_n^{(j)}, w_n^{(j)}\}_{j=1}^{M}$.
2. For $j = 1, \ldots, M$, sample from $p(\mathbf{x}_{n+1}|\mathbf{x}_n = \mathbf{x}_n^{(j)})$ to obtain $\{\mathbf{x}_{n+1}^{(j)}\}_{j=1}^{M}$.
3. Compute the mean $\bar{\boldsymbol{\mu}}_{n+1}$ and covariance $\bar{\boldsymbol{\Sigma}}_{n+1}$ as

$$\begin{aligned}\bar{\boldsymbol{\mu}}_{n+1} &= \tfrac{1}{M} \textstyle\sum_{j=1}^{M} w_n^{(j)} \mathbf{x}_{n+1}^{(j)} \\ \bar{\boldsymbol{\Sigma}}_{n+1} &= \tfrac{1}{M} \textstyle\sum_{j=1}^{M} w_n^{(j)} (\bar{\boldsymbol{\mu}}_{n+1} \quad \mathbf{x}_{n+1}^{(j)})(\bar{\boldsymbol{\mu}}_{n+1} \quad \mathbf{x}_{n+1}^{(j)})^H.\end{aligned} \tag{13}$$

Now, we recall a standard theorem of importance sampling, following which we provide a corollary that underscores the improvements obtained by the GPF in comparison to other Gaussian filters.

Theorem 1: Assume that at time n, (the analytical form of) $p(\mathbf{x}_n \mid \mathbf{y}_{0:n-1})$ is known up to a proportionality constant. On receiving the nth observation $\mathbf{y}_n$, the GPF measurement updates the filtering distribution using Monte Carlo integration defined in (4). Then, the rth moment (or the rth central moment) of $p(\mathbf{x}_n \mid \mathbf{y}_{0:n})$ estimated using (4) converges almost surely to $E(\mathbf{x}_n^r \mid \mathbf{y}_{0:n})$ [or to $E((\mathbf{x}_n - E(\mathbf{x}_n \mid \mathbf{y}_{0:n}))^r | \mathbf{y}_{0:n})$].

Proof: From (10) and (11), the rth moment is estimated as

$$\begin{aligned}\hat{E}\left(\mathbf{x}_n^r \mid \mathbf{y}_{0:n}\right) =& \frac{\sum_{i=1}^M \left(\mathbf{x}_n^{(i)}\right)^r \bar{w}_n^{(i)}}{\sum_{i=1}^M \bar{w}_n^{(i)}} \\ =& \frac{\sum_{i=1}^M \left(\mathbf{x}_n^{(i)}\right)^r \frac{p\left(\mathbf{y}_n|\mathbf{x}_n^{(i)}\right)p\left(\mathbf{x}_n^{(i)}|\mathbf{y}_{0:n-1}\right)}{\pi\left(\mathbf{x}_n^{(i)}|\mathbf{y}_{0:n}\right)}}{\sum_{i=1}^M \frac{p\left(\mathbf{y}_n|\mathbf{x}_n^{(i)}\right)p\left(\mathbf{x}_n^{(i)}|\mathbf{y}_{0:n-1}\right)}{\pi\left(\mathbf{x}_n^{(i)}|\mathbf{y}_{0:n}\right)}} \\ \rightarrow& \frac{\int \mathbf{x}_n^r \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1})}{\pi(\mathbf{x}_n|\mathbf{y}_{0:n})} \cdot \pi\left(\mathbf{x}_n \mid \mathbf{y}_{0:n}\right) d\mathbf{x}_n}{\int \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n-1})}{\pi(\mathbf{x}_n|\mathbf{y}_{0:n})} \cdot \pi\left(\mathbf{x}_n|\mathbf{y}_{0:n}\right) d\mathbf{x}_n} \\ =& \frac{\int \mathbf{x}_n^r p\left(\mathbf{y}_n \mid \mathbf{x}_n\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{0:n-1}\right) d\mathbf{x}_n}{\int p\left(\mathbf{y}_n \mid \mathbf{x}_n\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{0:n-1}\right) d\mathbf{x}_n} \\ =& \int \mathbf{x}_n^r p\left(\mathbf{x}_n \mid \mathbf{y}_{0:n}\right) d\mathbf{x}_n\end{aligned}$$

where the convergence is with probability one as $M \to \infty$ [see (5)] and is due to the Strong Law of Large Numbers. A similar proof holds for the rth central moment. ■

Corollary 1: Assume that at time n, $p(\mathbf{x}_n \mid \mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}_n; \bar{\boldsymbol{\mu}}_n, \bar{\boldsymbol{\Sigma}}_n)$. On receiving the nth observation $\mathbf{y}_n$, the GPF measurement updates the filtering distribution, as shown by the algorithm above. Then, $\boldsymbol{\mu}_n$ computed in (11) converges to the MMSE estimate of $\mathbf{x}_n$ almost surely as $M \to \infty$. In addition, the MMSE estimate given by $\boldsymbol{\Sigma}_n$ in (11) converges to the true MMSE estimate almost surely as $M \to \infty$.

Proof: The corollary follows straightforwardly from Theorem 1 since the MMSE estimate of $\mathbf{x}_n$ is given by $E_p(\mathbf{x}_n \mid \mathbf{y}_{0:n})$, and the MMSE is $E_p((\mathbf{x}_n - E_p(\mathbf{x}_n \mid \mathbf{y}_{0:n}))^2 \mid \mathbf{y}_{0:n})$. ■

The above corollary shows that, given the validity of the Gaussian approximation, the GPF provides the MMSE estimate asymptotically during the measurement update, which is clearly not true for the EKF and UKF. Hence, the GPF is expected to perform better than the EKF and UKF, which is validated by simulations.

1) Choice of $\pi(\cdot)$: The choice of the importance sampling function $\pi(\cdot)$ depends on the problem; see [23] and [25] for details. For the GPF, a simple choice for $\pi(\cdot)$ is $p(\mathbf{x}_n \mid \mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}_n; \bar{\boldsymbol{\mu}}_n, \bar{\boldsymbol{\Sigma}}_n)$ since samples from this density can be easily obtained. Alternatively, instead of generating completely new samples from $\mathcal{N}(\mathbf{x}_n; \bar{\boldsymbol{\mu}}_n, \bar{\boldsymbol{\Sigma}}_n)$ samples obtained in the time update step (presented in the next section) in step 2 can be used. However, this choice can be inadequate in some applications. Another choice is $\mathcal{N}(\mathbf{x}_n; \bar{\boldsymbol{\mu}}_{n|n}, \bar{\boldsymbol{\Sigma}}_{n|n})$, where $\bar{\boldsymbol{\mu}}_{n|n}$ and $\bar{\boldsymbol{\Sigma}}_{n|n}$ are obtained from the measurement update step of the EKF or from the unscented Kalman filter [26].

B. Time Update

Assume that at time n it is possible to draw samples from $p(\mathbf{x}_{n+1} \mid \mathbf{x}_n)$. With $p(\mathbf{x}_n \mid \mathbf{y}_{0:n})$ approximated as a Gaussian, we would like to obtain the predictive distribution $p(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n})$ and approximate it as a Gaussian. We recall (3), that is

$$\begin{aligned}p\left(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n}\right) &= \int p\left(\mathbf{x}_{n+1} \mid \mathbf{x}_n\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{0:n}\right) d\mathbf{x}_n \\ &\simeq \int p\left(\mathbf{x}_{n+1} \mid \mathbf{x}_n\right) \mathcal{N}\left(\mathbf{x}_n; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\right) d\mathbf{x}_n.\end{aligned}$$

A Monte Carlo approximation for the predictive distribution is given by

$$p\left(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n}\right) \approx \frac{1}{M} \sum_{i=1}^M p\left(\mathbf{x}_{n+1} \mid \mathbf{x}_n^{(i)}\right) \tag{14}$$

where $\mathbf{x}_n^{(i)}$ are particles from $\mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$. Following this observation, M samples from $p(\mathbf{x}_{n+1} \mid \mathbf{x}_n^{(i)})$, $i = 1, \ldots, M$ are obtained and are denoted as $\mathbf{x}_{n+1}^{(i)}$, from which the mean and covariance of $p(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n})$ is computed as

$$\begin{aligned}\bar{\boldsymbol{\mu}}_{n+1} =& \frac{1}{M} \sum_{i=1}^M \mathbf{x}_{n+1}^{(i)} \\ \bar{\boldsymbol{\Sigma}}_{n+1} =& \frac{1}{M} \sum_{i=1}^M \left(\bar{\boldsymbol{\mu}}_{n+1} - \mathbf{x}_{n+1}^{(i)}\right) \left(\bar{\boldsymbol{\mu}}_{n+1} - \mathbf{x}_{n+1}^{(i)}\right)^H.\end{aligned}$$

The GPF time update approximates $p(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n})$ as a Gaussian, or

$$p\left(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n}\right) \approx \mathcal{N}\left(\mathbf{x}_{n+1}; \bar{\boldsymbol{\mu}}_{n+1}, \bar{\boldsymbol{\Sigma}}_{n+1}\right).$$

Alternatively, we can use the weighted samples of $p(\mathbf{x}_n \mid \mathbf{y}_{0:n})$ obtained in the measurement update to get the following Monte Carlo approximation of $p(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n})$

$$\begin{aligned}p\left(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n}\right) &= \int p\left(\mathbf{x}_{n+1} \mid \mathbf{x}_n\right) p\left(\mathbf{x}_n \mid \mathbf{y}_{0:n}\right) d\mathbf{x}_n \\ &\simeq \frac{1}{M} \sum_{i=1}^M w_n^{(i)} p\left(\mathbf{x}_{n+1} \mid \mathbf{x}_n^{(i)}\right).\end{aligned}$$

Following this observation, M samples from $p(\mathbf{x}_{n+1} \mid \mathbf{x}_n^{(i)})$, $i = 1, \ldots, M$ are obtained and denoted as $\mathbf{x}_{n+1}^{(i)}$, from which the mean and covariance of $p(\mathbf{x}_{n+1} \mid \mathbf{y}_{0:n})$ is computed as

$$\begin{aligned}\bar{\boldsymbol{\mu}}_{n+1} =& \frac{1}{M} \sum_{i=1}^M w_n^{(i)} \mathbf{x}_{n+1}^{(i)} \\ \bar{\boldsymbol{\Sigma}}_{n+1} =& \frac{1}{M} \sum_{i=1}^M w_n^{(i)} \left(\bar{\boldsymbol{\mu}}_{n+1} - \mathbf{x}_{n+1}^{(i)}\right) \left(\bar{\boldsymbol{\mu}}_{n+1} - \mathbf{x}_{n+1}^{(i)}\right)^H.\end{aligned}$$

The GPF time update steps are summarized in Table I.

Similar to Theorem 1, it can be shown that as $M \to \infty$ and given the observations until time n, $\bar{\boldsymbol{\mu}}_{n+1}$ converges almost surely to the MMSE estimate of $\mathbf{x}_{n+1}$.

III. Discussion

From Corollary 1 (and its counterpart in the time update), we may deduce that the GPF provides better approximations to the integrations involved in the update processes than other Gaussian filters in the presence of severe nonlinearities. Two conclusions can be obtained from this observation. One is that divergence can be avoided by using the GPF unlike other Gaussian filters. The other is that the GPF provides more accurate estimates of the mean and covariance (confidence intervals) as a function of the number of particles than the other Gaussian filters. Simulations results in Section IV show that in general, the GPF has better performance than the EKF and UKF when the mean square errors are compared.

The EKF and its variants assume that the process and observation noise processes are *additive* and Gaussian. However, as long as the posterior distributions can be approximated as Gaussians, these assumptions can be relaxed for the GPF, and the proposed filter can be applied for a nonadditive and non-Gaussian noise DSS model. The measurement and time updates indicate that this is possible if $p(\mathbf{y}_n \mid \mathbf{x}_n)$ required in the weight update (9) can be evaluated and that a sample from $p(\mathbf{x}_{n+1} \mid \mathbf{x}_n)$ can be obtained.

As with most Gaussian filters, it is possible that bias accumulation may occur in certain applications. Particle filters with resampling also have bias due to resampling; see [27] for details. However, this phenomenon has not occurred in our simulation examples. A theoretical analysis of bias accumulation due to the approximations involved and finite samples is of interest but is left for future work.

SIS filters essentially obtain particles and their weights from the posterior distributions in a recursive manner. However, a phenomenon called *sample degeneration* occurs wherein only a few particles representing the distribution have significant weights. A procedure called *resampling* [22], [28] has been introduced to mitigate this problem, but it may give limited results and may be computationally expensive. Since the GPF approximates posterior distributions as Gaussians, particle resampling is not required. This results in a computational advantage of the GPF over SIS filters. Removal of resampling has another important advantage in implementation. Resampling is a nonparallel operation in an otherwise parallel SIS algorithm; hence, the GPF is more amenable for fully parallel implementation in VLSI.

The GPF propagates only the mean and covariance of the posterior densities. However, Theorem 1 shows that all moments can be estimated using importance sampling. This suggests a natural extension, wherein even higher moments are propagated. Alternatively, it may be motivating to approximate the posterior distributions by distributions other than Gaussian, which may be more accurate. This results in a new and much richer class of filters following our methodology. An example is given in the companion paper [24], where Gaussian mixtures approximate the distributions. Other distributions can also be considered, for example, the Student-t distribution, which has a heavier tail than the Gaussian; this may be motivating in certain applications. The mathematic tractability of propagating Gaussians has motivated researchers in the past to provide Gaussian approximations to posterior densities. However, the framework provided here makes it possible to obtain better approximations, albeit at the cost of computational power. We do not elaborate more on this new class of filters in this paper since the resulting filtering algorithms are straightforward extensions of ones provided in this paper.

IV. Simulation Results

The GPF proposed in this paper was applied to numerical examples, and here, we present the results for two models: the univariate nonstationary growth model, which is popular in econometrics and has been used previously in [17], [20], and [29], and the bearing only tracking model, which is of interest in defense applications [20], [30].

A. Univariate Nonstationary Growth Model (UNGM)

We choose this model because it is highly nonlinear and is bimodal in nature. The DSS equations for this model can be written as

$$\begin{aligned} x_n &= \alpha x_{n-1} + \beta \frac{x_{n-1}}{1+x_{n-1}^2} + \gamma \cos(1.2(n-1)) + u_n \\ y_n &= \frac{x_n^2}{20} + v_n, \quad n = 1, \ldots, N \end{aligned}$$

where $v_n \sim \mathcal{N}(0, \sigma_v^2)$, and the distribution of u_n is specified below. Data were generated using $x_0 = 0.1$, $\sigma_v^2 = 1$, $\alpha = 0.5, \beta = 25, \gamma = 8$, and $N = 500$. Notice the term in the process equation that is independent of x_n but varies with time n and that it can be interpreted as time-varying noise. The likelihood $p(y_n \mid x_n)$ has bimodal nature when $y_n > 0$ and when $y_n < 0$, it is unimodal. The bimodality makes the problem more difficult to address using conventional methods.

We compare the estimation and prediction performance of the EKF, UKF, GPF, and SIS filters based on the following metrics. MSEx_f is defined by

$$\text{MSE}x_f = \frac{1}{N} \sum_{n=1}^{N} (x_n - \hat{x}_n)^2 \tag{15}$$

where $\hat{x}_n = E(x_n \mid y_{0:n})$, and it is obtained from the filtering distribution. MSEx_p is defined similarly with $\hat{\hat{x}}_n = E(x_n \mid y_{0:n-1})$ and

$$\text{MSE}x_p = \frac{1}{N} \sum_{n=1}^{N} \left(x_n - \hat{\hat{x}}_n\right)^2 \tag{16}$$

where the estimate is obtained from the predictive distribution. MSEy_p is the mean square error computed from the predictive distribution of y_n, i.e., $p(y_n \mid y_{0:n-1})$. The MMSE estimate of y_n is given by $\hat{y}_n = E(y_n \mid y_{0:n-1})$ or

$$\begin{aligned} \hat{y}_n &= \int y_n p\left(y_n \mid y_{0:n-1}\right) dy_n \\ &= \iint y_n p\left(y_n \mid x_n\right) p\left(x_n \mid y_{0:n}\right) dx_n dy_n. \end{aligned}$$

For this example, we obtain

$$\hat{y}_n = \int \frac{x_n^2}{20} p\left(x_n \mid y_{0:n}\right) dx_n.$$

When the ratio $(x_n^2/20)/\sigma_v^2$ is small, the bimodality of the problem is more severe, and we expect to see improved performance of the GPF over that of the EKF in the presence of this high nonlinearity. The process noise $u_n \sim \mathcal{N}(0, \sigma_u^2)$, where $\sigma_u^2 = 1$. The initial distribution was $p(x_0) \sim \mathcal{N}(0, 1)$. For both GPF and SIS, the IS function was given by $p(x_n \mid y_{0:n-1})$ (in SIS terminology, the IS function is written as $p(x_n \mid x_{n-1})$). The applied SIS filter was the one from [31], where resampling was performed at every step using the systematic resampling scheme [32]. For the EKF, we get $\hat{y}_n = (\mu_n^2 + \sigma_n^2)/20$. For the GPF, since we draw particles from $p(x_n \mid y_{0:n})$ in the measurement update, we obtain a Monte Carlo estimate for $\hat{y}_n$.

A large number of simulations were performed to compare the four filters. In Figs. 1 and 2, we show plots for the first 100

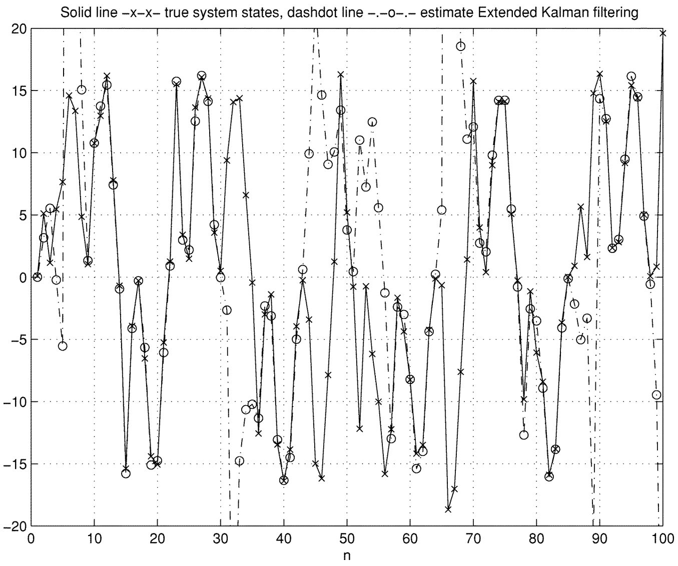

Fig. 1. Plot of the true state and estimate of the EKF.

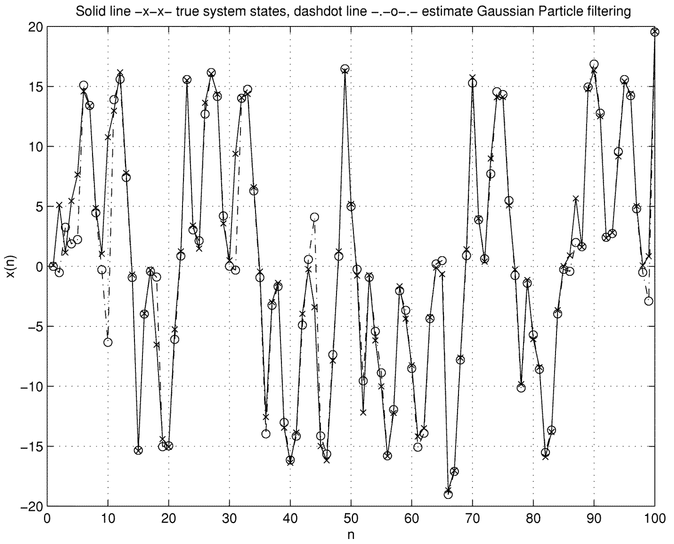

Fig. 2. Plot of the true state and estimate of the GPF.

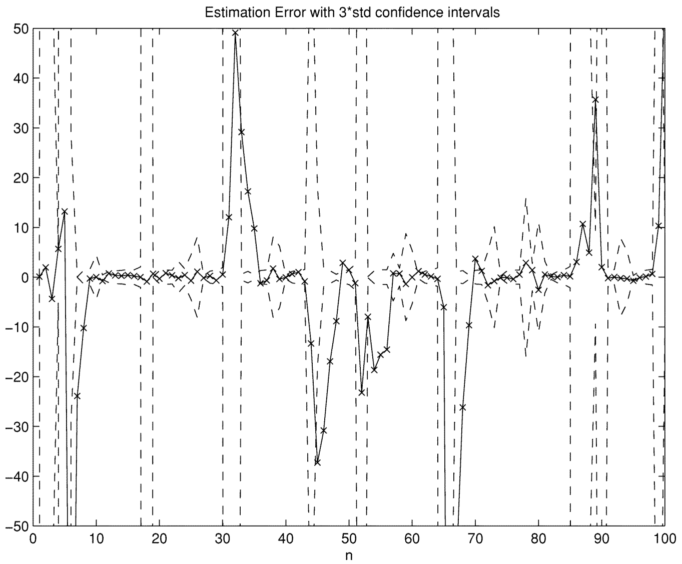

Fig. 3. Plot of the prediction error and $3\sigma_{\rm err}$ interval for the EKF.

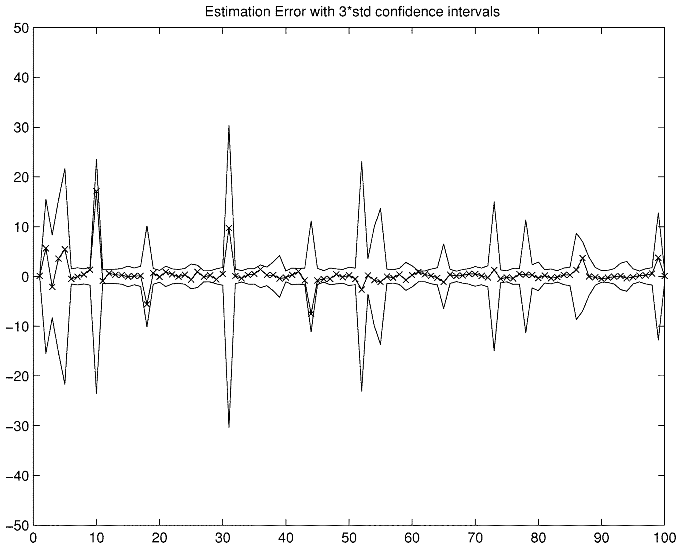

Fig. 4. Plot of the prediction error and $3\sigma_{\rm err}$ interval for the GPF.

states and the estimates obtained using the EKF and GPF with $M = 100$, respectively, for a single simulation. Note the tendency of the EKF to track the opposite mode of the bimodality, especially when $(x_n^2/20)/\sigma_v^2$ is small. This behavior was observed in general for most simulation runs.

In Figs. 3 and 4, we plot the error $x_n - \hat{x}_n$ and the $3\hat{\sigma}_{\rm err}$ intervals, where $\hat{\sigma}_{\rm err}$ was the estimated standard deviation of the prediction error. As expected, the errors lie mostly within this interval for the GPF, which is not the case for the EKF. In addition, the values of $\hat{\sigma}_{\rm err}$ for the EKF are much higher, pointing to the occurrence of divergence of the filter. All of the above observations were made in most of the simulation runs. Clearly, the GPF outperforms the EKF significantly for this highly nonlinear example.

Fig. 5 shows the MSEx_f for 50 random realizations with $M = 100$ particles. In Fig. 7, the average MSEx_f is plotted for $M = 20$, 100, and 1000 particles. Even for a small number of particles $M = 20$, the GPF and SISR filters perform significantly better than the EKF and UKF, whereas the MSE for the GPF is marginally higher than that of SISR. An increase in the number of particles reduced the MSE even further for the GPF and SISR filters; however, the change in performance is negligible as the number of particles was increased from 100. As observed from the figures, for $M = 100$ and 1000, there is insignificant difference in the MSEs for the GPF and the SISR. Similar behavior was observed for the MSEx_p (see Fig. 6) and the MSEy_p metrics.

A comparison of the computation times is also shown in Fig. 7 for simulations implemented on a 450-MHz Intel Pentium III processor using MATLAB. Note that as expected, the computation time for GPF and SISR filters is much higher than the EKF and UKF. However, as noted in the Section III, these times indicate those obtained on a serial computer and much reduction in computation times can be expected for the GPF and SISR when implemented in parallel. For the EKF, UKF, and GPF, a

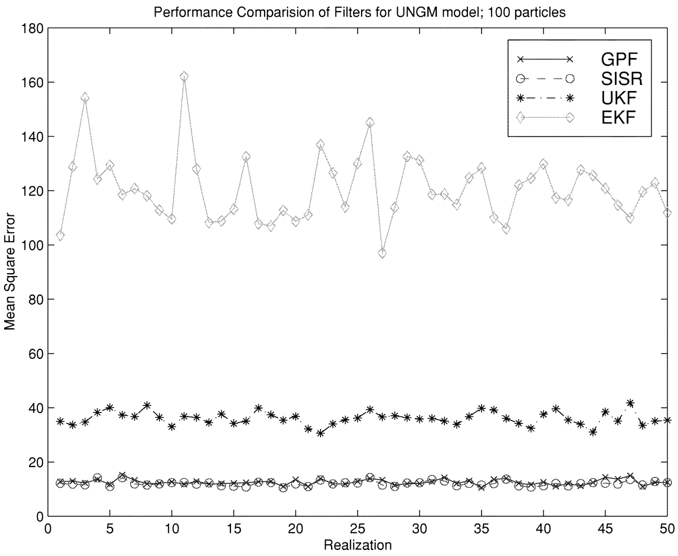

Fig. 5. Performance comparison of EKF, UKF, GPF, and SISR filters; MSEx_f is plotted for 50 random realizations. $M = 100$ for both GPF and SISR filters.

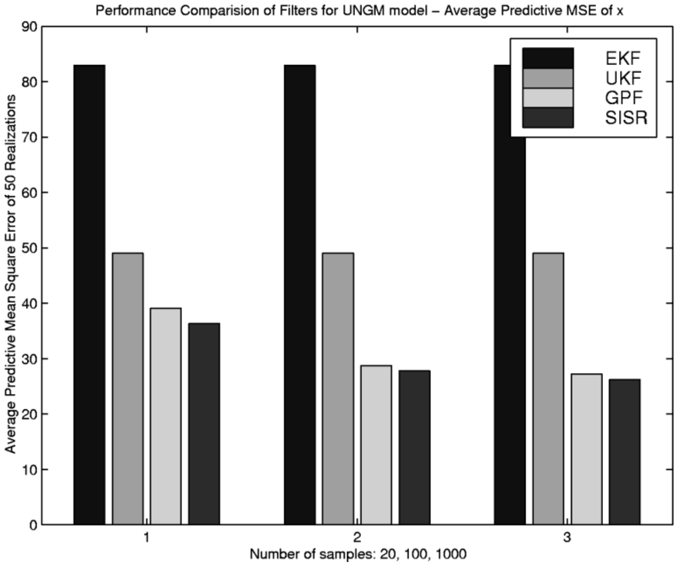

Fig. 6. Performance comparison of EKF, UKF, GPF, and SISR filters. Average MSEx_p of 50 random realizations. M=20, 100, and 1000 for both GPF and SISR filters.

clear performance-computational time tradeoff can be seen in the figure. More importantly, the GPF has much less of a computation time than the SISR, and the difference increases as the number of particles increases. This is due to the additional resampling required in the SISR filter, which has computational complexity of $O(N)$ for the systematic resampling scheme used here.

B. Bearings Only Tracking (BOT)

The bearings-only model is well motivated and arises in defense applications and in engineering problems. The bearings only tracking problem considers tracking an object moving in a 2-D space. The measurements taken by the sensor at fixed intervals to track the object are the bearings or angles (subject to noise) with respect to the sensor. The range of the object, that is, the distance from the sensor, is not measured.

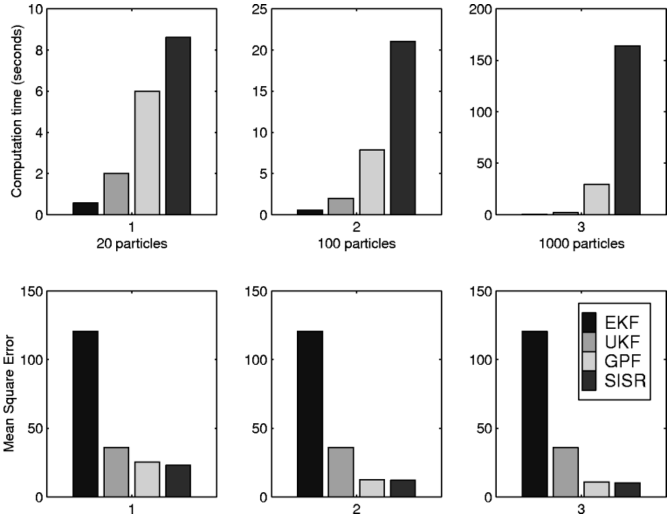

Fig. 7. Performance comparison of EKF, UKF, GPF, and SISR filters. Computation time (per realization) and average MSEx_f of 50 random realizations. $M = 20$, 100, and 1000 for both GPF and SISR filters.

We employ a change of notation here to better facilitate understanding of the mechanics of the problem. Let the sensor be stationary and located at the origin in the x—y plane. The object moves in the x–y plane according to the following process model:

$$\mathbf{x}_n = \mathbf{\Phi}\mathbf{x}_{n-1} + \mathbf{\Gamma}\mathbf{w}_n \quad n = 1, \ldots, N$$

where $\mathbf{x}_n = [x_n, \quad v_{xn}, \quad y_n, \quad v_{yn}]^T$, $\mathbf{w}_n = [w_{xn}, \quad w_{yn}]^T$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ \mathbf{\Gamma} = \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix}.$$

Here, x and y denote the Cartesian coordinates of the target v_x, and v_y denote the velocities in the x and y directions, respectively. The system noise is a zero mean Gaussian white noise, that is, $\mathbf{w}_n \sim N(\mathbf{0}, \sigma_w^2 \mathbf{I}_2)$, where $\mathbf{I}_2$ is the 2×2 identity matrix. The initial state $\mathbf{x}_0$ describes the targets initial position and velocity. A prior for the initial state $p(\mathbf{x}_0)$ also needs to be specified for the model; we assume $\mathbf{x}_0 \sim N(\boldsymbol{\mu}_0, \mathbf{P}_0)$.

The measurements consist of the true bearing of the target corrupted by a Gaussian error term. The measurement equation can be written as

$$z_n = \tan^{-1}\left(\frac{y_n}{x_n}\right) + v_n \tag{17}$$

where the measurement noise is a zero mean Gaussian white noise, that is, $v_n \sim N(0, \sigma_v^2)$.

It is important to note that from (17), we obtain no information about the range of the object from the measurement. Thus, given a series of measurements, we can only detect in general the shape of the trajectory but not its location or distance from the point of measurement. In particular, if we consider the likelihood associated with a single measurement z_n as a function of x_n and y_n, that is $p(z_n \mid x_n, y_n)$, it is clear that the likelihood consists of a modal ridge along the line $y_n = \tan(z_n)x_n$. The prior $p(\mathbf{x}_0)$ describes our knowledge about the position and

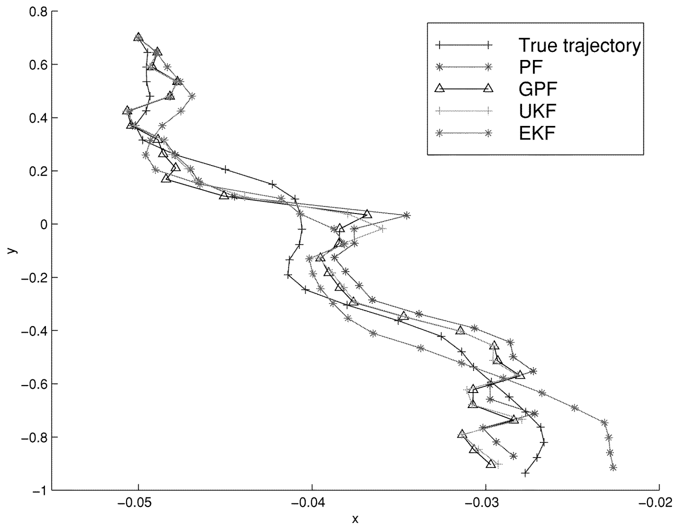

Fig. 8. Tracking of a moving target in two dimensions with the EKF, UKF, GPF, and SISR filters.

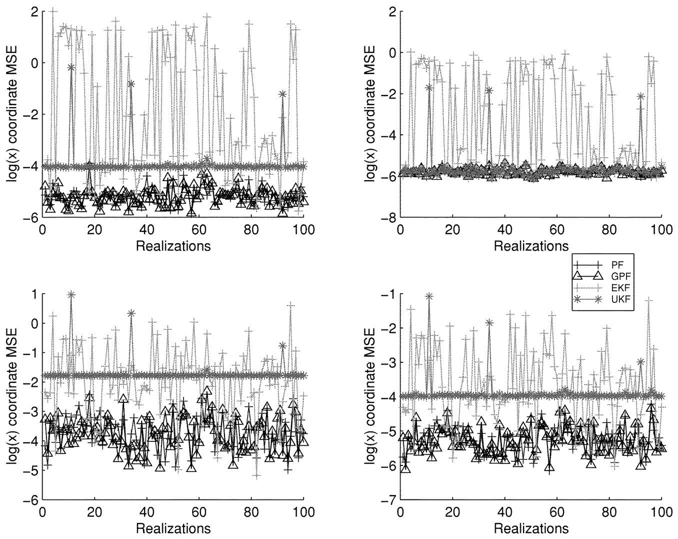

Fig. 9. Performance comparision of EKF, UKF, GPF, and SISR filters. MSEs for position and velocities in x and y directions. M $= 1000$ for both GPF and SISR filters.

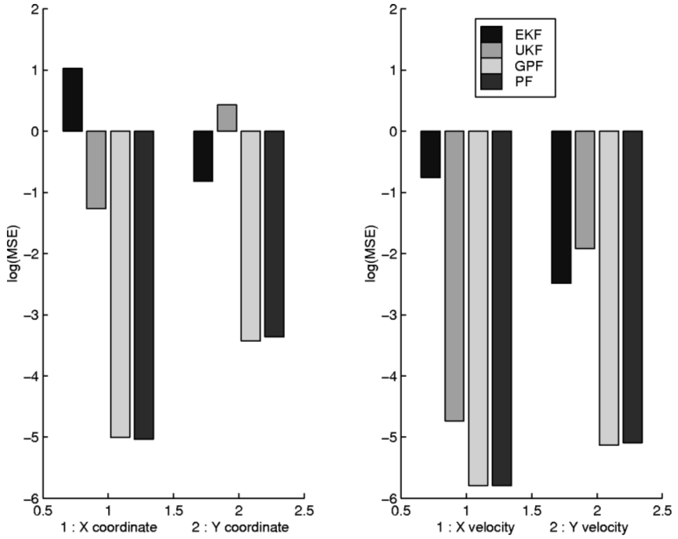

Fig. 10. Performance comparision of EKF, UKF, GPF, and SISR filters. Average MSEs for position and velocities in x and y directions. M $= 1000$ for both GPF and SISR filters.

velocity of the target at the initial stage. This prior and the likelihood of the series of measurements are combined to obtain an estimate of the trajectory.

A target trajectory and associated measurements over 24 time steps was generated with initial state vector $\mathbf{x}_0 = [-0.05, \quad 0.001, \quad 0.7, \quad -0.055]^T$. The process and measurement noise were as specified above with $\sigma_w = 0.001$ and $\sigma_v = 0.005$. Note that for this particular trajectory, in the initial stages, the bearing of the target with respect to the observer remains almost constant. As the object passes the observer, the change in the angle is large. For the prior, the parameters assumed were $\boldsymbol{\mu}_0 = [-0.05, \quad 0.001, \quad 0.7, \quad -0.055]^T$ (which is the same as the initial vector) and

$$\mathbf{P}_0 = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0.1 & 1 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}.$$

The EKF, UKF, GPF, and SISR filters were applied to get an estimate of the trajectory. The number of particles used for the GPF and SISR filters was $M = 1000$. Systematic random resampling was used for the SISR filter. Observe that due to the nonlinearity of the observation equation, it is not straightforward to sample from the optimal importance function. Hence, we use the prior $p(\mathbf{x}_n \mid z_{0:n-1})$ and $p(\mathbf{x}_n \mid \mathbf{x}_{n-1})$ as the IS function for the GPF and SISR, respectively.

Fig. 8 displays a representative trajectory and the tracking obtained by all four filters. Fig. 9 shows the MSE for 100 random realizations for all the state variables. The EKF diverged in more than 50 realizations, whereas the UKF diverged for three realizations. Fig. 10 shows the average MSE obtained for same 100 random realizations for all the state variables on a logarithmic scale. Fig. 11 displays a trajectory where the EKF and UKF both diverged. In cases where divergence was not observed, the MSEs of EKF and UKF were much higher compared to the MSEs of the GPF and SISR filters. On an average, the UKF performed better than the EKF; however, the MSE of the v_{xn} was much higher for the UKF. Comparison of the GPF and SISR filters shows that the GPF has marginally better performance then the SISR filter.

In another experiment, all the above simulations were replicated for a varying number of particles. For $M < 500$, the GPF and SISR filters diverged, implying that a greater number of particles was required for stability. It was observed that increasing M from 500 to 1000 improved the performance. Increasing M further gave minor improvements.

Computation times are shown in Fig. 12. As expected, the EKF and UKF have very small computation time compared with the particle filters. However, the high use of computational power is justified for the GPF and SISR since in most cases, the target is tracked well. Again, since resampling is required in the SISR, it is more computationally expensive, and a large difference is observed in the computation times of the GPF and SISR.

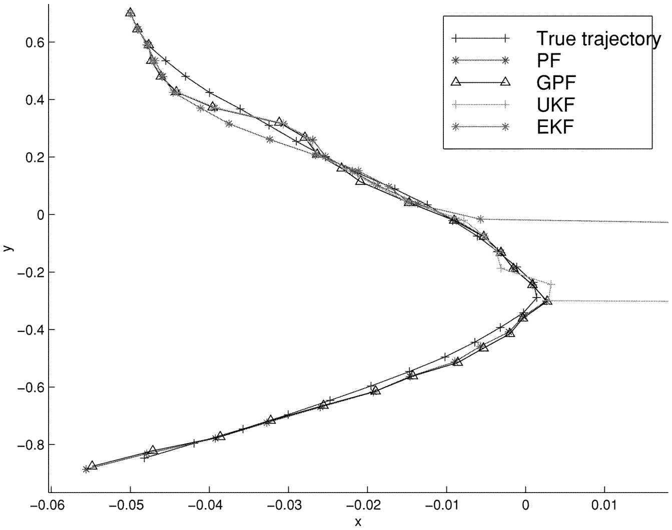

Fig. 11. Example of divergence of the EKF and UKF in the bearings-only tracking example, whereas the GPF and SISR filters do not diverge.

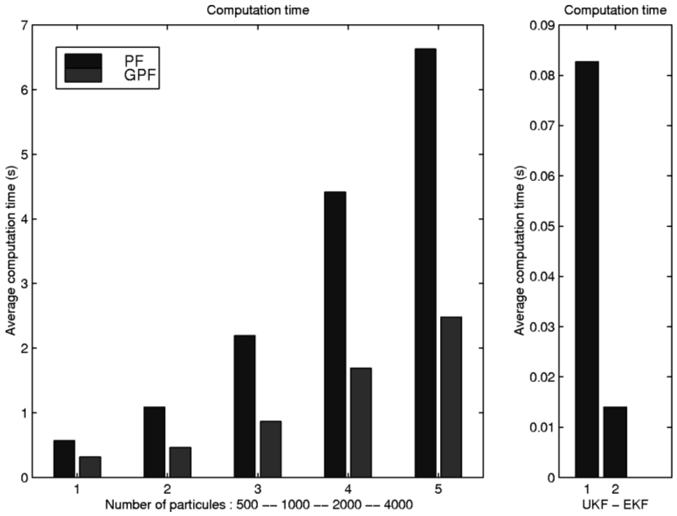

Fig. 12. Performance comparison of EKF, UKF, GPF, and SISR filters; computation time (per realization) versus number of particles for both GPF and SISR filters.

V. Conclusion

The Gaussian particle filter provides much better performance than the EKF and UKF. Moreover, the additive Gaussian noise assumption can be relaxed without any modification to the filter algorithm. The update of the filtering and predictive distributions as Gaussians by particle-based approaches has the advantages of easier implementation than standard SIS algorithms and better performance than algorithms that are also based on Gaussian approximations. The GPF is more computationally expensive than the EKF. However, the parallelizibility of the GPF and the absence of resampling makes it convenient for VLSI implementation and, hence, feasible for practical real-time applications. In a companion paper [24], we exploit the GPF to build Gaussian sum particle filters that can be used for models where the the filtering and predictive distributions cannot be approximated successfully with a single Gaussian distribution and for models with non-Gaussian noise. The latter models can be approximated as a weighted bank of Gaussian noise models, and hence, the GPF developed here can also be used to solve the non-Gaussian noise problem.

References

[1] J. H. Kotecha and P. M. Djurić, "Gaussian particle filtering," in *Proc. Workshop Statistical Signal Process.*, Singapore, Aug. 2001.

[2] ——, "Gaussian sum particle filtering for dynamic state space models," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Salt Lake City, UT, May 2001.

[3] P. J. Harrison and C. F. Stevens, "Bayesian forecasting (with discussion)," *J. R. Statist. Soc. B*, vol. 38, pp. 205–247, 1976.

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

[5] M. S. Grewal and A. P. Andrews, *Kalman Filtering*, 2nd ed. New York: Wiley, 2001.

[6] S. F. Schmidt, "Advances in control systems," in *Applications of State Space Methods to Navigation Problems*, C. T. Leondes, Ed. New York: Academic, 1966.

[7] H. W. Sorenson, *Recursive Estimation for Nonlinear Dynamic Systems*. New York: Marcel Dekker, 1988.

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

[9] Y. Bar-Shalom and Xiao-Rong Li, *Estimation and Tracking*. Norwell, MA: Artech House, 1993.

[10] S. Frühwirth-Schnatter, "Applied state-space modeling of non-Gaussian time series using integration based Kalman filtering," *Statist. Comput.*, vol. 4, pp. 259–269, 1994.

[11] C. J. Masreliez, "Approximate non-Gaussian filtering with linear state and observation relations," *IEEE Trans. Automat. Contr.*, vol. 20, pp. 107–110, 1975.

[12] M. West, P. J. Harrison, and H. S. Migon, "Dynamic generalized linear models and Bayesian forecasting, (with discussion)," *J. Amer. Statist. Assoc.*, vol. 80, pp. 73–97, 1985.

[13] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proc. Amer. Contr. Conf.*, Seattle, WA, June 1995, pp. 1628–1632.

[14] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. Automat. Contr.*, vol. 45, pp. 910–927, May 2000.

[15] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximation," *IEEE Trans. Automat. Contr.*, vol. 17, pp. 439–448, July 1972.

[16] R. S. Bucy, "Bayes theorem and digital realization for nonlinear filters," *J. Astronaut. Sci.*, vol. 80, pp. 73–97, 1969.

[17] G. Kitagawa, "Non-Gaussian state-space modeling of nonstationary time series," *J. Amer. Statist. Assoc.*, vol. 82, no. 400, pp. 1032–1063, 1987.

[18] S. C. Kramer and H. W. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximations," *Automatica*, vol. 24, pp. 789–801.

[19] A. Pole and M. West, "Efficient numerical integration in dynamic models," Dept. Statist., Univ. Warwick, Warwick, U.K., Res. Rep., 136, 1988.

[20] N. Gordon, D. Salmond, and C. Ewing, "Bayesian state estimation for tracking and guidance using the bootstrap filter," *J. Guidance, Contr., Dyn.*, vol. 18, no. 6, pp. 1434–1443, Nov.-Dec. 1995.

[21] J. H. Kotecha and P. M. Djurić, "Sequential monte carlo sampling detector for Rayleigh fast-fading channels," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2000.

[22] J. S. Liu and R. Chen, "Monte Carlo methods for dynamic systems," *J. Amer. Statist. Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.

[23] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[24] J. H. Kotecha and P. M. Djurić, "Gaussian sum particle filtering," *IEEE Trans. Signal Processing*, vol. 51, pp. 2603–2613, Oct. 2003.

[25] J. Geweke, "Bayesian inference in econometrics models using Monte Carlo integration," *Econometrica*, vol. 33, no. 1, pp. 1317–1339, 1989.

[26] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Automat. Contr.*, vol. 45, pp. 477–482, Mar. 2000.

[27] C. Berzuini, N. G. Best, W. R. Gilks, and C. Larizza, "Dynamic conditional independence models and Markov chain Monte Carlo," *J. Amer. Statist. Assoc.*, vol. 92, no. 440, pp. 1403–1412, Dec. 1997.

[28] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2000.

[29] E. R. Beadle and P. M. Djurić, "A fast weighted Bayesian bootstrap filter for nonlinear model state estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, pp. 338–342, Jan. 1997.

[30] D. Avitzour, "Stochastic simulation Bayesian approach to multitarget tracking," *Proc. Inst. Elect. Eng.—Radar Sonar Navigat.*, vol. 142, no. 2, pp. 41–44, Apr. 1995.

[31] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Proc. Inst. Elect. Eng. F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[32] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *Proc. Inst. Elect. Eng.—Radar, Sonar, Navig.*, vol. 146, no. 1, pp. 2–7, Feb. 1999.

Jayesh H. Kotecha received the B.E. degree in electronics and telecommunications from the College of Engineering, Pune, India, in 1995 and the M.S. and Ph.D. degrees from the State University of New York at Stony Brook in 1996 and 2001, respectively.

Since January 2002, he has been with the University of Wisconsin, Madison, as a post-doctoral researcher. His research interests are primarily in the fields of communications, signal processing, and information theory. Presently, he is working in statistical signal processing, adaptive signal processing, and particle filters and physical layer aspects in multi-input multi-ouput communication systems, including channel modeling, tranceiver design, space-time coding, and capacity analysis.

Peter M. Djurić (SM'99) received the B.S. and M.S. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1981 and 1986, respectively, and the Ph.D. degree in electrical engineering from the University of Rhode Island, Kingston, in 1990.

From 1981 to 1986, he was Research Associate with the Institute of Nuclear Sciences, Vinca, Belgrade, Yugoslavia. Since 1990, he has been with the State University of New York at Stony Brook, where he is Professor with the Department of Electrical and Computer Engineering. He works in the area of statistical signal processing, and his primary interests are in the theory of modeling, detection, estimation, and time series analysis and its application to a wide variety of disciplines, including telecommunications, bio-medicine, and power engineering.

Prof. Djuric has served on numerous Technical Committees for the IEEE and SPIE and has been invited to lecture at universities in the United States and overseas. He was Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and currently, he is the Treasurer of the IEEE Signal Processing Conference Board and Area Editor of Special Issues of the IEEE SIGNAL PROCESSING MAGAZINE. He is also Vice Chair of the IEEE Signal Processing Society Committee on Signal Processing—Theory and Methods and a Member of the American Statistical Association and the International Society for Bayesian Analysis.