GAUSSIAN SUM PARTICLE FILTERING FOR DYNAMIC STATE SPACE MODELS

Jayesh H. Kotecha and Petar M. Djurić

Department of Electrical and Computer Engineering
State University of New York at Stony Brook, Stony Brook, NY 11794
jkotecha@ece.sunysb.edu, djuric@ece.sunysb.edu

ABSTRACT

For dynamic systems, sequential Bayesian estimation requires updating of the filtering and predictive densities. For nonlinear and non-Gaussian models, sequential updating is not as straightforward as in the linear Gaussian model. In this paper, densities are approximated as finite mixture models as is done in the Gaussian sum filter. A novel method is presented, whereby sequential updating of the filtering and posterior densities is performed by particle based sampling methods. The filtering method has combined advantages of Gaussian sum and particle based filters and simulations show that the presented filter can outperform both methods.

1. INTRODUCTION

Many problems in statistical signal processing can be written in the form of the so called Dynamic State Space (DSS) model [1]. The signal of interest $\{\mathbf{x}_n; n \in \mathbb{N}\}, \mathbf{x} \in \mathbb{R}^{m_x}$, is an unobserved (hidden) Markov process of initial distribution $p(\mathbf{x}_0)$ represented by the distribution $p(\mathbf{x}_n|\mathbf{x}_{n-1})$. The states $\{\mathbf{x}_n; n \in \mathbb{N}\}$ are not observed directly, and hence are called the hidden states of the DSS. We shall use 'signal' and 'state' interchangeably. The observations $\{\mathbf{y}_n; n \in \mathbb{N}\}, \mathbf{y} \in \mathbb{R}^{m_y}$, are conditionally independent given the state process $\{\mathbf{x}_n; n \in \mathbb{N}\}$ and represented by the distribution $p(\mathbf{y}_n|\mathbf{x}_n)$. Alternatively the model can be written as

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

where $\mathbf{u}_n$ 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 $\mathbf{x}_n$. Observations of the system are functions of the signal usually distorted by noise.

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

- the marginal posterior distribution of the state at time n given all the observations up to time n referred to as the filtering distribution $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ and
- the prediction distribution of $\mathbf{x}_{n+1}$ given all the observations up to time n, $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})$.

When the model is linear with Gaussian noise, the filtering and prediction densities are Gaussian, and the Kalman filter provides the mean and covariance sequentially, which is the optimal Bayesian solution [2]. For most nonlinear models and non-Gaussian noise problems, closed form analytic expression for the posterior densities do not exist. Numerical solutions often require high dimensional integrations that are not practical to implement. The extended Kalman filter can be applied, however in a few limited cases. An interesting method is the Gaussian sum (GS) filter [2], whereby the posterior densities are approximated as a finite Gaussian mixture. Recently, particle based sampling filters have been used to update the posterior distributions [3],[4],[5]. A density is represented by a weighted set of samples from the density, which are propagated through the dynamic system to sequentially update the posterior densities. These methods will be collectively called sequential importance sampling (SIS) filters.

In this paper, we present a novel approach to update the posterior densities called Gaussian sum particle (GSP) filtering. As in the Gaussian sum filters, the filtering and prediction densities are approximated as finite Gaussian mixtures, but the sequential update is carried out using sampling based methods. For a nonlinear model, the Gaussian sum filter uses a bank of extended Kalman filters running in parallel, wherein the filter equations are obtained by linearizing around the current state. This introduces errors in the filtering process, which can cause the filter to diverge. Another important difficulty presents itself when the process covariance is large, causing the number of mixands to grow exponentially [6]. In the GSP, an update using particle methods improves upon the above approximation, thereby giving better performance. The number of mixands can be kept constant using *residual random resampling*. The presented simulations show that the GSP filter exhibits better performance than the GS and SIS filters.

2. GAUSSIAN MIXTURE APPROXIMATION

Closed form expressions for the filtering and prediction densities do not exist in general and therefore, approximations in the form of finite Gaussian mixtures will be developed. Two important theorems are recalled below from [2] for this purpose. Let $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the normal density of a random vector $\mathbf{x}$ where the m-vector $\boldsymbol{\mu}$ is the mean, and the covariance is the nonsingular matrix $\boldsymbol{\Sigma}$. The following lemma indicates that any given density can be approximated as a Gaussian mixture. For a proof see [2], page 213.

Lemma 1 *Any probability density $p(\mathbf{x})$ can be approximated as closely as desired by a density of the form*

$$p_G(\mathbf{x}) = \sum_{i=1}^{G} w_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{2}$$

for some integer G, positive scalars w_i with $\sum_{i=1}^{G} w_i = 1$, m-vectors $\boldsymbol{\mu}_i$ and positive definite matrices $\boldsymbol{\Sigma}_i$, so that $\int_{\mathbb{R}^m} |p(\mathbf{x}) -$

This work was supported by the National Science Foundation under Award Nos. CCR-9903120 and CCR-0082607.

$p_G(\mathbf{x})|d\mathbf{x} \leq \epsilon$ *for any given* ϵ*.*

Numerical methods can be used to obtain the mixing weights, means and covariance matrices, which minimize a given norm such as the one stated in Lemma 1 [7]. However, more practical approaches take into account the class of densities to be approximated.

To obtain insight in the nature of the approximations involved, we recall the following theorem (for a proof see [2], page 197).

Theorem 1 *In model 1, let the noise vectors* $\mathbf{v}_n$ *and* $\mathbf{u}_n$ *be white, Gaussian noises with zero mean and covariance matrices* $\mathbf{R}_n$ *and* $\mathbf{Q}_n$*, respectively. If* $p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1})$*, then for fixed* $\mathbf{h}_n(\cdot)$*,* $\boldsymbol{\mu}_{n|n-1}$ *and* $\mathbf{R}_n$

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{y}_{0:n}) &= c_n p(\mathbf{x}_n|\mathbf{y}_{0:n-1}) p(\mathbf{y}_n|\mathbf{x}_n) \\ &\longrightarrow \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n}) \end{aligned} \tag{3}$$

uniformly in $\mathbf{x}_n$ *and* $\mathbf{y}_n$ *as* $\boldsymbol{\Sigma}_{n|n-1} \longrightarrow 0$*, where* c_n *is a normalizing constant. Also if,* $p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n})$*, then for fixed* $\mathbf{f}(\cdot)$*,* $\boldsymbol{\mu}_{n|n}$ *and* $\mathbf{y}_{0:n}$*,*

$$\begin{aligned} p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) &= \int p(\mathbf{x}_{n+1}|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n}) d\mathbf{x}_n \\ &\longrightarrow \mathcal{N}(\mathbf{x}_{n+1}; \boldsymbol{\mu}_{n+1|n}, \boldsymbol{\Sigma}_{n+1|n}) \end{aligned} \tag{4}$$

as $\boldsymbol{\Sigma}_{k|k} \longrightarrow 0$*. In the above expressions, the mean and covariances are obtained using the extended Kalman filter equations, where the subscript* $n|n-1$ *indicates the parameter estimate at time* n *given data* $\mathbf{y}_{0:n-1}$*.*

The theorem indicates that approximations can be obtained for the filtering and prediction densities if the covariance matrices $\boldsymbol{\Sigma}_{n|n-1}$ and $\boldsymbol{\Sigma}_{n|n}$ are small. Taking this into account, if the densities are modeled as a finite Gaussian mixture with small covariance matrices, then parallel updates as shown in the above theorem will yield good approximations to the updated densities. Hence, the Gaussian sum (GS) filter results in a bank of parallel extended Kalman filters, under the assumption that the covariance matrices are small. The GS filter assumes that the noise processes involved are Gaussian, and the extended Kalman filter equations are formed by linearizing the process and observation equations. In practical applications, where the number of mixands in the approximation of the prediction and filtering densities is not large, divergence may still occur as a result of the linearizations. Another problem occurs when the covariance of the mixands grows, which causes all the mixands to collapse, resulting in only one distinct trajectory. The covariance of the mixands grows especially when the process noise is large compared to the covariance of the mixands. To combat this problem, in [6] it has been suggested to approximate the Gaussian noise process as a finite Gaussian mixture itself. However, this results in an exponentially growing number of mixands.

3. GAUSSIAN SUM PARTICLE (GSP) FILTERING

The GS filter assumes that the noise processes are Gaussian, however for the GSP filter this assumption can be relaxed. Update of the filtering and prediction densities is done using particles, which allows for the observation noise to be non-Gaussian. Non-Gaussian process noise densities are approximated as finite Gaussian mixtures following theorem 1. In the following, the GSP filter time and measurement update algorithms are presented with the assumption that the density of the non-Gaussian process noise $\mathbf{u}_n$ is approximated as a finite Gaussian mixture. The observation noise $\mathbf{v}_n$ is assumed Gaussian, however the extension to non Gaussian noise is straightforwardly deduced by approximating the noise as a finite Gaussian mixture. Thus we have

$$p(\mathbf{u}_n) = \sum_{k=1}^{K} \alpha_k \mathcal{N}(\mathbf{u}_n; \tilde{\boldsymbol{\mu}}_{nk}, \tilde{\boldsymbol{\Sigma}}_{nk}). \tag{5}$$

The linearizations involved in the extended Kalman filter are not invoked, and a more accurate approximation to the updated densities can be achieved using particles [8].

For the DSS model 1, suppose that the density $p(\mathbf{x}_0)$ is expressed as a Gaussian mixture. Given that, we would like to obtain the filtering and prediction densities recursively and approximate them as Gaussian mixtures.

3.1. Time update

Assume that at time n, we have

$$p(\mathbf{x}_n|\mathbf{y}_{0:n}) = \sum_{g=1}^{G} w_{ng} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{ng}, \boldsymbol{\Sigma}_{ng}). \tag{6}$$

With $p(\mathbf{x}_n|\mathbf{y}_{0:n})$ expressed as a Gaussian mixture, we would like to obtain the predictive density $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n})$ and approximate it as a Gaussian mixture. We have

$$\begin{aligned} & p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) = \int p(\mathbf{x}_{n+1}|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{0:n}) d\mathbf{x}_n \\ & = \int \textstyle\sum_{k=1}^{K} \alpha_k \mathcal{N}(\mathbf{x}_{n+1}; \mathbf{f}(\mathbf{x}_n) + \tilde{\boldsymbol{\mu}}_{(n+1)k}, \tilde{\boldsymbol{\Sigma}}_{(n+1)k}) \\ & \textstyle\sum_{g=1}^{G} w_{ng} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{ng}, \boldsymbol{\Sigma}_{ng}) d\mathbf{x}_n \\ & = \textstyle\sum_{g=1}^{G} \sum_{k=1}^{K} \alpha_k w_{ng} \\ & \int \mathcal{N}(\mathbf{x}_{n+1}; \mathbf{f}(\mathbf{x}_n) + \tilde{\boldsymbol{\mu}}_{(n+1)k}, \tilde{\boldsymbol{\Sigma}}_{(n+1)k}) \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{ng}, \boldsymbol{\Sigma}_{ng}) d\mathbf{x}_n \end{aligned} \tag{7}$$

where equations (6) and (5) have been used to obtain the expression. Upon inspection of the expression in the integral, we see that the nonlinearity of the process equation makes the integration quite intractable. The integral can be approximated as a Gaussian following theorem 1 as is done in the GS filter. Then the time update algorithm is presented below.

For clarity of notation define $g' = g + (k-1)K$ and $G' = GK$. Thus in the above equation we have G' mixands and references to g' implies references to the respective g and k, since they are uniquely mapped.

1. For $g = 1, \ldots, G$, obtain samples from $\mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{ng}, \boldsymbol{\Sigma}_{ng})$ and denote them as $\{\mathbf{x}_{ng}^{(j)}\}_{j=1}^{M}$.
2. For $g' = 1, \ldots, G'$, $j = 1, \ldots, M$ obtain samples from $\mathcal{N}(\mathbf{x}_{n+1}; \mathbf{f}(\mathbf{x}_n = \mathbf{x}_{ng}^{(j)}) + \tilde{\boldsymbol{\mu}}_{(n+1)k}, \tilde{\boldsymbol{\Sigma}}_{(n+1)k})$ and denote them as $\{\mathbf{x}_{(n+1)g'}^{(j)}\}_{j=1}^{M}$.
3. For $g' = 1, \ldots, G'$, the weights for each mixand are updated as
$$\bar{w}_{(n+1)g'} = \frac{w_{ng}\alpha_k}{\sum_{k=1}^{K} \sum_{g=1}^{G} w_{ng}\alpha_k}.$$
4. For $g' = 1, \ldots, G$, $\{\mathbf{x}_{(n+1)g'}^{(j)}\}_{j=1}^{M}$ are distributed as Gaussian samples, obtain mean $\bar{\boldsymbol{\mu}}_{(n+1)g'}$ and covariance $\bar{\boldsymbol{\Sigma}}_{(n+1)g'}$ by taking sample means and covariances.

The time updated (prediction) density can now be approximated as

$$p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) = \sum_{g'=1}^{G'} \bar{w}_{(n+1)g'} \mathcal{N}(\mathbf{x}_{n+1}; \bar{\boldsymbol{\mu}}_{(n+1)g'}, \bar{\boldsymbol{\Sigma}}_{(n+1)g'}). \tag{8}$$

Inspection of equation (7) shows that, the number of mixands at each time update step have increased from G to G'. As in the GS filter this can result in an exponentially growing number of mixands [6]. In order to keep the number of mixands constant, we introduce the use of residual random resampling, which throws away trajectories that have insignificant weights. Resampling is performed after the measurement step explained below.

3.2. Measurement update

With $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n+1})$ expressed as a Gaussian mixture, we would like to obtain the filtering density $p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n+1})$ and approximate it as a Gaussian mixture. After receiving the $n+1$-th observation $\mathbf{y}_{n+1}$, we update the filtering density as follows:

$$\begin{aligned} &p(\mathbf{x}_{n+1}|\mathbf{y}_{n+1}) = C_{n+1} p(\mathbf{y}_{n+1}|\mathbf{x}_{n+1}) p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n}) \\ &= C_{n+1} \sum_{g'=1}^{G'} \bar{w}_{(n+1)g'} \mathcal{N}(\mathbf{x}_{n+1}; \bar{\boldsymbol{\mu}}_{(n+1)g'}, \bar{\boldsymbol{\Sigma}}_{(n+1)g'}) \\ &\cdot p(\mathbf{y}_{n+1}|\mathbf{x}_{n+1}) \end{aligned} \tag{9}$$

where C_{n+1} is a normalizing constant and equation (8) has been used. Using theorem 1, each term on the right hand side given by $\mathcal{N}(\mathbf{x}_{n+1}; \bar{\boldsymbol{\mu}}_{(n+1)g'}, \bar{\boldsymbol{\Sigma}}_{(n+1)g'}) p(\mathbf{y}_{n+1}|\mathbf{x}_{n+1})$ can be approximated as a Gaussian. This allows for the following update algorithm :

1. For $g' = 1, \ldots, G'$, obtain samples from the distribution $\mathcal{N}(\mathbf{x}_{n+1}; \bar{\boldsymbol{\mu}}_{(n+1)g'}, \bar{\boldsymbol{\Sigma}}_{(n+1)g'})$, and denote them as $\{\mathbf{x}_{(n+1)g'}^{(j)}\}_{j=1}^{M}$.
2. As in importance sampling, obtain the respective weights by $\delta_{(n+1)g'}^{(j)} = p(\mathbf{y}_{n+1}|\mathbf{x}_{n+1} = \mathbf{x}_{(n+1)g'}^{(j)})$, $j = 1, \ldots, M$.
3. Since $\mathcal{N}(\mathbf{x}_{n+1}; \bar{\boldsymbol{\mu}}_{(n+1)g'}, \bar{\boldsymbol{\Sigma}}_{(n+1)g'}) p(\mathbf{y}_{n+1}|\mathbf{x}_{n+1})$ is approximated by a Gaussian, the weighted samples obtained and denoted as $\{\mathbf{x}_{(n+1)g'}^{(j)}, \delta_{(n+1)g'}^{(j)}\}_{j=1}^{M}$ approximately represent a Gaussian. The mean and covariance are estimated by

$$\boldsymbol{\mu}_{(n+1)g'} = \frac{\sum_{j=1}^{M} \delta_{(n+1)g'}^{(j)} \mathbf{x}_{(n+1)g'}^{(j)}}{\sum_{j=1}^{M} \delta_{(n+1))g'}^{(j)}}$$

$$\begin{aligned} &\boldsymbol{\Sigma}_{(n+1)g'} = \\ &\frac{\sum_{j=1}^{M} \delta_{(n+1)g'}^{(j)} (\mathbf{x}_{(n+1)g'}^{(j)} - \boldsymbol{\mu}_{(n+1)g'})(\mathbf{x}_{(n+1)g'}^{(j)} - \boldsymbol{\mu}_{(n+1)g'})^T}{\sum_{j=1}^{M} \delta_{(n+1)g'}^{(j)}}. \end{aligned} \tag{10}$$

4. Update the weights as

$$\tilde{w}_{(n+1)g'} = \bar{w}_{ng'} \frac{\sum_{j=1}^{M} \delta_{(n+1)g'}^{(j)}}{\sum_{g'=1}^{G} \sum_{j=1}^{M} \delta_{(n+1)g'}^{(j)}}, \quad g' = 1, \ldots, G'. \tag{11}$$

5. Normalize the weights as

$$w_{(n+1)g'} = \frac{\tilde{w}_{(n+1)g'}}{\sum_{g'=1}^{G'} \tilde{w}_{(n+1)g'}}. \tag{12}$$

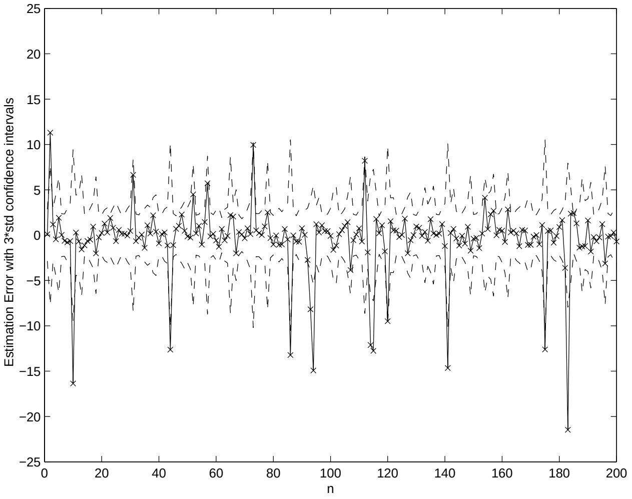

Fig. 1. Error plot of the estimate along with 3σ confidence intervals for the GSP with resampling filter.

The updated filtering density can now be represented as

$$\begin{aligned} &p(\mathbf{x}_{n+1}|\mathbf{y}_{0:n+1}) = \\ &\sum_{g'=1}^{G'} w_{(n+1)g'} \mathcal{N}(\mathbf{x}_{n+1}; \boldsymbol{\mu}_{(n+1)g'}, \boldsymbol{\Sigma}_{(n+1)g'}). \end{aligned} \tag{13}$$

6. Resampling is performed to reduce the number of mixands from G' to G, as explained in the following section.

3.3. Gaussian resampling for the GSP filter

In many cases it may happen that the weights of all but one mixand may become insignificant. In such a case, the GS becomes essentially the extended Kalman filter and the filtering density needs to be reexpressed as a Gaussian mixture with more meaningful weights. Typically, it is advantageous practically, to keep the number of mixands constant. For the GSP, we suggest using a method called residual random sampling [5] which applies a correction mechanism, so that all the mixands have significant weights, see [8] for details. The correction mechanism discards the trajectories with insignificant weights. To keep the number of mixands constant, mixands with significant weights are *duplicated*. When the trajectories are duplicated, the weights are also proportionally redistributed.

3.4. Inference

The Gaussian sum approximation lends an advantage in that estimation of the hidden state and the error covariance becomes straightforward. From equation(13), the estimate of $\mathbf{x}_n$, $\hat{\mathbf{x}}_n = E(\mathbf{x}_n|\mathbf{y}_{0:n})$ and the error covariance $\hat{\boldsymbol{\Sigma}}_n = E(\mathbf{x}_n - \hat{\mathbf{x}}_n)(\mathbf{x}_n - \hat{\mathbf{x}}_n)^T$ can be approximated as

$$\begin{aligned} \hat{\mathbf{x}}_n &= \sum_{i=1}^{G} w_{ni} \boldsymbol{\mu}_{ni} \\ \hat{\boldsymbol{\Sigma}}_n &= \sum_{i=1}^{G} w_{ni} (\boldsymbol{\Sigma}_{ni} + (\mathbf{x}_n - \boldsymbol{\mu}_{ni})(\mathbf{x}_n - \boldsymbol{\mu}_{ni})^T). \end{aligned} \tag{14}$$

4. SIMULATION RESULTS

One important choice to be made is that of the number of mixands G. Although theoretical results suggest that a large G is required, the choice will in general depend on the particular problem. Simulation results of the GS filter on some examples in [6], [2] suggest that the GS filter works satisfactorily even when G is surprisingly small (say $G = 6$). In simulations of the GSP presented here similar observations have been made. The GSP filter is applied to the univariate non-stationary growth model. The DSS equations for this model can be written as:

$$\begin{aligned} x_n &= \alpha x_{n-1} + \beta \frac{x_{n-1}}{1+x_{n-1}^2} + \gamma \cos(1.2(n-1)) + u_n, \\ y_n &= x_n^2/20 + v_n, \quad n = 1, \ldots, N \end{aligned} \qquad (15)$$

where data were generated using $x_0 = 0.1$ and $u_n \sim \mathcal{N}(1,0)$ $\forall n$ and $\alpha = 0.5, \beta = 25$ and $\gamma = 8$. This is highly nonlinear in both the process and observation equations. Notice the term in the process equation which is independent of x_n but varies with time n, which can be interpreted as time varying noise. The likelihood $p(y_n|x_n)$ has bimodal nature when $y_n > 0$, but when $y_n < 0$, it is unimodal. The bimodality makes the problem more difficult to address using conventional methods. In this example we would like to illustrate the application of these filters in an interesting scenario of heavy tailed non-Gaussian noise in a highly nonlinear model. The GSP filter is well suited to tackle these problems, because heavy tailed densities can be modeled as a Gaussian mixture [9]. For the UNGM model, the noise is now distributed as a Gaussian mixture given by

$$p(u_n) = \epsilon \mathcal{N}(u; 0, \sigma_{u1}^2) + (1-\epsilon)\mathcal{N}(u; 0, \sigma_{u2}^2).$$

By varying ϵ and the variances, heavy tailed densities can be modeled quite well. We show results where $\epsilon = 0.8$, $\sigma_{u1}^2 = 0.1$ and $\sigma_{u2}^2 = 1$.

Comparisions are made between the GS filter and the GSP filter. A large number of simulations were performed where both filters were used for state estimation. For the present example we have $G = 16$. The number of particles chosen for each mixand update was $M = 100$. Resampling was performed with the threshold chosen as $w_{thresh} = 0.001$. All the mixands in the prior density $p(\mathbf{x}_0)$ are distributed as $\mathcal{N}(0,1)$. Figure 1 shows the prediction error along with 3σ confidence intervals for the GSP filter. Clearly, the filter shows good performance for this simulation run and similar observations were made in other simulations. In Table 1, we show the mean square error (MSE) for 10 random simulations, where MSE is defined by

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{x}_n)^2, \qquad (16)$$

where $\hat{x}_n$ is an estimate of x_n. In the table, we also show the sample average

$$V = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{x}_n)^2 \qquad (17)$$

for each simulation run for the two filters. The parameter V can be interpreted as an estimate of σ_v^2 and it indicates how well the filter has been able to deal with the nonlinearities of the problem. The closer $\frac{V}{\sigma_v^2}$ is to 1, the better the performance of the filter for that simulation run.

Note that the GSP filter outperforms the GS filter significantly according to the two metrics for this example. The non-Gaussian noise is easily accomodated in this problem and increasing number of mixands are reduced by resampling. This example illustrates the potential of the proposed filter to address a large number of nonlinear problems with non-Gaussian noise.

Simulation	MSE		V	
Number	GS	GSP	GS	GSP
1	94.60	6.72	491.60	1.18
2	75.11	5.30	131.55	1.34
3	39.77	4.44	13.87	1.50
4	105.49	6.80	1163.63	1.46
5	85.66	7.46	137.91	1.34
6	81.32	2.72	459.54	1.19
7	89.27	1.00	138.06	1.71
8	137.50	9.06	1595.93	1.92
9	61.31	4.46	152.63	1.12
10	187.01	9.71	6138.90	1.49

Table 1. MSE and V parameters defined in equations (16) and (17) for 10 simulation runs for the GS and GSP filters.

5. CONCLUSION

Updating the filtering and prediction densities as finite Gaussian mixtures using particle based approach has the advantages of easy implementation and better performance. The GSP filter combines the principles of the conventional Gaussian sum filtering and particle based filtering methods to obtain better approximations for the finite Gaussian mixture. Simulations show that some of the limitations of the GS filter are overcome by the GSP, which lead to better performance.

6. REFERENCES

[1] P.J. Harrison and C.F. Stevens, “Bayesian forecasting (with discussion),” *Journal of the Royal Statistical Society , Ser B*, vol. 38, pp. 205–247, 1976.

[2] B.D.O Anderson and J.B.Moore, *Optimal Filtering*, Prentice Hall Inc., Englewood Ciffs, New Jersey 07632, 1979.

[3] N. Gordon, D. Salmond, and C. Ewing, “Bayesian state estimation for tracking and guidance using the bootstrap filter,” *Journal of Guidance, Control and Dynamics*, vol. 18, no. 6, pp. 1434–1443, Nov-Dec. 1995.

[4] J. H. Kotecha and P.M. Djurić, “Sequential Monte Carlo sampling detector for Rayleigh fast-fading channels,” *International Conference on Acoustics, Speech and Signal Processing*, 2000.

[5] J.S. Liu and R. Chen, “Monte Carlo methods for dynamic systems,” *Journal of American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[6] D.L. Alspach and H.W. Sorenson, “Nonlinear Bayesian Estimation using Gaussian Sum Approximation,” *IEEE Transactions Auto. Cont.*, vol. 17, no. 4, pp. 439–448, 1972.

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

[8] J. H. Kotecha and P.M. Djurić, “Gaussian sum particle filtering,” *submitted to Transactions on Signal Processing*, Feb 2001.

[9] D. Sengupta and S. Kay, “Efficient estimation of parameters for non-Gaussian autoregressive processes,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 6, pp. 785–794, June 1989.