Metrol. Meas. Syst., Vol. XX (2013), No. 1, pp. 65–76.

PAN
POLSKA AKADEMIA NAUK

METROLOGY AND MEASUREMENT SYSTEMS
Index 330930, ISSN 0860-8229
www.metrology.pg.gda.pl

M&S

STATE ESTIMATION BASED ON GENERALIZED GAUSSIAN DISTRIBUTIONS

Xifeng Li, Yongle Xie

University of Electronic Science and Technology of China, School of Automation Engineering, Chengdu, 611731, China
(✉ yonglexie@hotmail.com. +86 137 0808 3375, *xifenglee@126.com)*

Abstract

This paper presents a novel strategy of particle filtering for state estimation based on Generalized Gaussian distributions (GGDs). The proposed strategy is implemented with the Gaussian particle pilter (GPF), which has been proved to be a powerful approach for state estimation of nonlinear systems with high accuracy and low computational cost. In our investigations, the distribution which gives the complete statistical characterization of the given data is obtained by exponent parameter estimation for GGDs, which has been solved by many methods. Based on GGDs, an extension of GPF is proposed and the simulation results show that the extension of GPF has higher estimation accuracy and nearly equal computational cost compared with the GPF which is based on Gaussian distribution assumption.

Keywords: Generalized Gaussian distributions, state estimation, Gaussian particle pilter, nonlinear systems.

1. Introduction

Possibly the most important problem arising in measurement activities is the representation and maintenance of uncertainty. The state of a system, whether measured or estimated, is rarely known exactly because (a) measuring instruments and processes have limited precision, and/or (b) estimates of dynamic systems are based on process models that fail to include all governing parameters. The uncertainty associated with a state estimate can be represented most generally by a probability distribution incorporating all information about the state. However, measurements of an evolving system generally imply that the number of parameters necessary to specify the state probability distribution will increase to infinity [1]. Considering the constraints of available resources, an approximation state estimation must be generated. The most common approach is to maintain a fixed number of moments of the state distribution. The most successful approach for fixed-moments estimation is the Kalman filter [2]. For linear systems, the Kalman filter provides the optimal solution for maintaining a consistent estimate of the first two moments of the state distribution: the mean and variance. However, the requirement that all measurement and process models be linear is rarely satisfied in nontrivial applications.

In order to apply the mechanics of the Kalman filter to nonlinear problems, based on the concept of sequential importance sampling and the use of Bayesian theory, particle filtering is especially useful in dealing with nonlinear and non-Gaussian problems. The underlying principle of the methodology is the approximation of relevant distributions with random measures composed of particles (samples from the space of the unknowns) and their associated weights [3]. Recently, a special class of particle filters called Gaussian particle filter (GPF) that approximates the posterior distributions by single Gaussians has been introduced [4]. Although in their derivation it is assumed that all the relevant distributions are Gaussian, as is done with some other filters including the extended Kalman filter and its

Article history: received on Jun. 04, 2012; accepted on Jan. 09, 2013; available online on Mar. 15, 2013; DOI: 10.2478/mms-2013-0006.

variants, they are distinguishable in that the updating of the filtering and predictive distributions is accomplished by propagating particles. This entails advantages of easier implementation than is the case with the standard particle filters and improved performance over other Gaussian based approximation filters.

However, all results derived from GPF are based on Gaussian assumption, especially for its propagating particles. According to the Central Limit Theorem, the limit behaviour of the stochastic variable is Gaussian. So the true Probability density function (PDF) in an actual situation is an approximation of Gaussian (or Gaussian-like) as a result of insufficient sampling points [5].

In this paper, a review of some the most popular tracking (state estimation) algorithms is shown. Several methods of parameter estimation for generalized Gaussian probability density functions are also presented. We employ Generalized Gaussian distributions (GGDs) to perform the particle propagation in a Gaussian particle filter. The effectiveness of the use of GGDs in place of the Gaussian assumption in GPF is elucidated.

The method presented in this paper has two contributions. The first one is to improve the estimation accuracy of the conventional Gaussian particle filter. The second contribution is to maintain the nearly equal computational cost compared with the GPF. Two nonlinear system models are presented to illustrate our results, simulation results are provided to verify our analysis. This paper is organized in the following order. In Section 2, the most popular state estimation algorithms are outlined, and the method of parameter estimation for generalized Gaussian probability density functions is presented in Section 3. The GGDs based Gaussian particle filter technique is proposed in Section 4. Our method is validated by the simulations from two cases in Section 5. Important conclusions are drawn in Section 6.

2. State estimation methods – from the Kalman filter to particle filtering

A large portion of the theory on measurement model is about measurement data (signals) and systems that are represented by state-space and observation equations, that is, equations of the form:

$$x_t = f_t(x_{t-1}, u_t), \tag{1}$$

$$y_t = g_t(x_t, v_t), \tag{2}$$

where y_t is a vector of observations, x_t is a state vector, $g_t(x_t, v_t)$ is a measurement function, $f_t(x_{t-1}, u_t)$ is a system transition function, u_t and v_t are noise vectors, and the subscript t denotes a time index. The first equation is known as state equation, and the second, as measurement equation. The standard assumptions are that the analytic forms of the functions and the distributions of the two noises are known. Based on the observation y_t and the assumptions, the objective is to estimate x_t recursively.

The approach that has been investigated the most and that has been frequently used in practice is the Kalman filter. The Kalman filter is optimal in the important case when the equations are linear and the noises are independent, addition, and Gaussian. In this situation, the distributions of interest (filtering, predictive, or smoothing) are also Gaussian and the Kalman filter can compute them exactly without approximations. For scenarios where the models are nonlinear or the noise is non-Gaussian, various approximation methods have been proposed. Among these methods, the extended Kalman filter is perhaps the most prominent of all [6].

The particle filtering method has become an important alternative to the extended Kalman filter. With particle filtering, continuous distributions are approximated by discrete random measures, which are composed of weighted particles, where the particles are samples of the unknown states from the state space, and the particle weights are “probability masses” computed by using Bayes’ theory. In the implementation of particle filtering, importance sampling plays a crucial role and, since the procedure is designed for sequential use, the method is also called sequential importance sampling. The advantage of particle filtering over other methods is in that the exploited approximation does not involve linearization around current estimates but rather approximation in the representation of the desired distributions by discrete random measures.

2.1. Particle filtering

As already pointed out, the main task of sequential measurement signal processing is the estimation of the state x_t recursively from the observations y_t. In general, three probability distribution functions are of interest, and they are the filtering distribution $p(x_t \mid y_{0:t})$; the predictive distribution $p(x_{t+l} \mid y_{0:t}), l \geq 1$; and the smoothing distribution $p(x_t \mid y_{0:T}), T > t$. All the information about x_t regarding filtering, prediction, or smoothing is captured by these distributions, respectively, and so the main goal is their tracking, which is obtaining $p(x_t \mid y_{0:t})$ from $p(x_{t-1} \mid y_{0:t-1})$, $p(x_{t+l} \mid y_{0:t})$ from $p(x_{t+l-1} \mid y_{0:t})$, or $p(x_t \mid y_{0:T})$ from $p(x_{t+1} \mid y_{0:T})$. The algorithms that exactly track these distributions are known as optimal algorithms. In many practical situations, however, the optimal algorithms are impossible to implement, primarily because the distribution updates require integrations that cannot be performed analytically or summations that are impossible to carry out due to the curse of dimensionality.

For the joint a posteriori distribution of $x_0, x_1, ..., x_t$, in case of independent noise samples which are assumed through the article, we can write:

$$p(x_{0:t} \mid y_{0:t}) \propto p(x_0 \mid y_0) \prod_{k=1}^{t} p(y_k \mid x_k) p(x_k \mid x_{k-1}). \tag{3}$$

It is straightforward to show that a recursive formula for obtaining $p(x_{0:t} \mid y_{0:t})$ from $p(x_{0:t-1} \mid y_{0:t-1})$ is given by:

$$p(x_{0:t} \mid y_{0:t}) = \frac{p(y_t \mid x_t) p(x_t \mid x_{t-1})}{p(y_t \mid y_{0:t-1})} p(x_{0:t-1} \mid y_{0:t-1}). \tag{4}$$

Since the transition from $p(x_{0:t-1} \mid y_{0:t-1})$ to $p(x_{0:t} \mid y_{0:t})$ is often analytically intractable, it is necessary to resort to approaches that are based on approximations.

In particle filtering, the distributions are approximated by discrete random measures defined by particles and weights assigned to the particles. If the distribution of interest is $p(x)$ and its approximating random measure is:

$$\chi = \{x^{(j)}, w^{(j)}\}_{j=1}^{M}, \tag{5}$$

where $x^{(j)}$ are the particles, $w^{(j)}$ are their weights, and M is the number of particles used in the approximation, χ approximates the distribution $p(x)$ by:

$$p(x) = \sum_{j=1}^{M} w^{(j)} \delta(x - x^{(j)}), \tag{6}$$

where $\delta(x)$ is the Dirac delta function.

The next important concept used in particle filtering is the principle of importance sampling. Suppose we want to approximate a distribution $p(x)$ with a discrete random measure. If we can generate the particles from $p(x)$, each of them will be assigned a weight equal to $1/M$. When direct sampling from $p(x)$ is intractable, one can generate particles $x^{(j)}$ from a distribution $\pi(x)$, known also as importance function, and assign (nonnormalized) weights according to:

$$w^{*(j)} = \frac{p(x)}{\pi(x)}, \tag{7}$$

which upon normalization become:

$$w^{(j)} = \frac{w^{*(j)}}{\sum_{m=1}^{M} w^{*(j)}}. \tag{8}$$

Suppose that the posterior distribution $p(x_{0:t-1} \mid y_{0:t-1})$ is approximated by the discrete random measure $\chi_{t-1} = \{x_{0:t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^{M}$. Note that the trajectories of particles $x_{0:t-1}^{(j)}$ can be considered particles of $p(x_{0:t-1} \mid y_{0:t-1})$. Given the discrete random measures χ_{t-1} and the observation y_t, the objective is to exploit χ_{t-1} in obtaining χ_t. Sequential importance sampling methods achieve this by generating particles $x_t^{(j)}$ and appending them to $x_{0:t-1}^{(j)}$ to form $x_{0:t}^{(j)}$, and updating the weights $w_t^{(j)}$ so that χ_t allows accurate estimates of the unknown of interest at time t.

If we use an importance function that can be factored as:

$$\pi(x_{0:t} \mid y_{0:t}) = \pi(x_t \mid x_{0:t-1}, y_{0:t})\pi(x_{0:t-1} \mid y_{0:t-1}) \tag{9}$$

and if:

$$x_{0:t-1}^{(j)} \sim \pi(x_{0:t-1} \mid y_{0:t-1}) \tag{10}$$

and:

$$w_{t-1}^{(j)} \propto \frac{p(x_{0:t-1}^{(j)} \mid y_{0:t-1})}{\pi(x_{0:t-1}^{(j)} \mid y_{0:t-1})} \tag{11}$$

we can augment the trajectory $x_{0:t-1}^{(j)}$ with $x_t^{(j)}$, where:

$$x_t^{(j)} \sim \pi(x_t \mid x_{0:t-1}^{(j)}, y_{0:t}) \tag{12}$$

and easily associated with it an updated weight $w_t^{(j)}$ obtained according to:

$$w_t^{(j)} \propto \frac{p(y_t \mid x_t^{(j)}) p(x_t^{(j)} \mid x_{t-1}^{(j)})}{\pi(x_t^{(j)} \mid x_{0:t-1}^{(j)}, y_{0:t})} w_{t-1}^{(j)}. \tag{13}$$

The sequential importance sampling algorithm can thus be implemented by performing the following two steps for every t.

1) Draw particles $x_t^{(j)} \sim \pi(x_t \mid x_{0:t-1}^{(j)}, y_{0:t})$, where $j = 1,2,\ldots,M$.
2) Compute the weights of $w_t^{(j)}$ according to (13).

The importance function plays a very important role in the performance of the particle filter. This function must have the same support as the probability distribution that is being approximated. In general, the closer the importance function to that distribution, the better the approximation.

In the literature, the two most frequently used importance functions are the prior and the optimal importance function. The implementations of particle filters with prior importance functions are much easier than those with optimal importance functions because the prior importance functions have lower computational cost.

A major problem with particle filtering is that the discrete random measure degenerates quickly, that is to say, all the particles except for a very few are assigned negligible weights.

The degeneracy implies that the performance of the particle filter will deteriorate. Degeneracy, however, can be reduced by using good importance sampling functions and resampling.

Resampling is a scheme that eliminates particles with small weights and replicates particles with large weights. In principle, it is carried out as follows:

Step 1. Draw M particles, $x_t^{*(j)}$ from the discrete distribution χ_t.

Step 2. Let $x_t^{(j)} = x_t^{*(j)}$, and assign equal weights $1/M$ to the particles.

In summary, the procedure of particle filtering involves the following steps:

Step 1. Description of the problem by a discrete state-space model as in (1) and (2).

Step 2. Selection of proposal function for particle generation.

Step 3. Derivation of a proposal functions for the weight update.

Additional issues are the choice of resampling algorithm and the schedule for resampling.

2.2. Gaussian Particle Filtering

A major disadvantage of particle filters is the computational complexity, a large part of which comes from resampling. The Gaussian particle filter (GPF) is quite similar to the particle filter by the fact that importance sampling is used to obtain particles. However, unlike the particle filters, resampling is not required in the GPF, and the GPF only propagates the mean and covariance of the posterior distributions. This results in a reduced complexity of the GPF as compared with the particle filter with resampling and is a major advantage.

The GPF measurement updates and time updates steps are summarized as follows:

GPF – measurement update algorithm.

1) Draw samples from the importance function $\pi(x_t \mid y_{0:t})$ and denote them as $\{x_t^{(j)}\}_{j=1}^{M}$.
2) Obtain the respective weights by:

$$\overline{w}_t^{(j)} = \frac{p(y_t \mid x_t^{(j)}) N(x_t = x_t^{(j)}; \overline{\mu}_t, \overline{\Sigma}_t)}{\pi(x_t^{(j)} \mid y_{0:n})}, \tag{14}$$

where $\overline{\mu}_t$ is the mean, and the covariance is the positive definite matrix $\overline{\Sigma}_t$. $N(x_t = x_t^{(j)}; \overline{\mu}_t, \overline{\Sigma}_t)$ is the Gaussian distribution.

3) Normalize the weight as:

$$w_t^{(j)} = \frac{\overline{w}_t^{(j)}}{\sum_{j=1}^{M} \overline{w}_t^{(j)}}. \tag{15}$$

4) Estimate the mean and covariance by :

$$\begin{aligned} \mu_t &= \sum_{j=1}^{M} w_t^{(j)} x_t^j, \\ \Sigma_t &= \sum_{j=1}^{M} w_t^{(j)} (\mu_t - x_t^j)(\mu_t - x_t^j)^H. \end{aligned} \tag{16}$$

GPF – time update algorithm.

1) Draw samples from $N(x_t; \mu_t, \Sigma_t)$ and denote them as $\{x_t^{(j)}\}_{j=1}^{M}$.
2) For $j = 1,\ldots,M$, sample from $p(x_{t+1} \mid x_t = x_t^j)$ to obtain $\{x_{t+1}^{(j)}\}_{j=1}^{M}$.
3) Compute the mean $\overline{\mu}_{t+1}$ and covariance $\overline{\Sigma}_{t+1}$ as:

$$\bar{\mu}_{t+1} = \frac{1}{M}\sum_{j=1}^{M} x_{t+1}^{j},$$
$$\bar{\Sigma}_{t+1} = \sum_{j=1}^{M} (\bar{\mu}_{t+1} - x_{t+1}^{j})(\bar{\mu}_{t+1} - x_{t+1}^{j})^{H}. \tag{17}$$

3. Parameter estimation for Generalized Gaussian distributions

Precisely modeling unknown Probability density functions (PDFs) of data are very important for the efficient design of modern signal processing and controlling systems. The parametric family of densities coming from the Generalized Gaussian distribution (GGD) are known to model successfully a large number of different signals and systems [7–12], where the GGD model has been shown to provide , in most cases, a fairly accurate representation of the unknown distributions.

For any zero-mean ($\mu = 0$) signal $x \in R$, the PDF of a Generalized Gaussian distribution with standard deviation σ is defined as [7]:

$$p_v(x) = \frac{v \cdot \eta(v,\sigma)}{2 \cdot \Gamma(\frac{1}{v})} \exp\{-[\eta(v,\sigma) \cdot |x|]^{v}\}, \tag{18}$$

in which:

$$\eta(v,\sigma) = \frac{1}{\sigma}\left[\frac{\Gamma(3/v)}{\Gamma(1/v)}\right]^{1/2} \tag{19}$$

and $\Gamma(v)$ is the Gamma function, *i.e.*, $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt, z > 0$. Since $\Gamma(n) = (n-1)!$, when v =1 a Laplacian law is obtained, while v = 2, yields a Gaussian distribution. As the limit case, for $v \to 0$, $p_v(x)$ becomes an impulse function, yet having flat tails and thus nonzero σ^2 variance, whereas for $v \to \infty$, $p_v(x)$ approaches a uniform distribution having variance σ^2 as well. The shape parameter v rules the exponential rate of decay: the larger the v, the flatter the PDF; the smaller the v, the more peaked the PDF. Fig. 1 shows the trend of the GGD function for different values of the v.

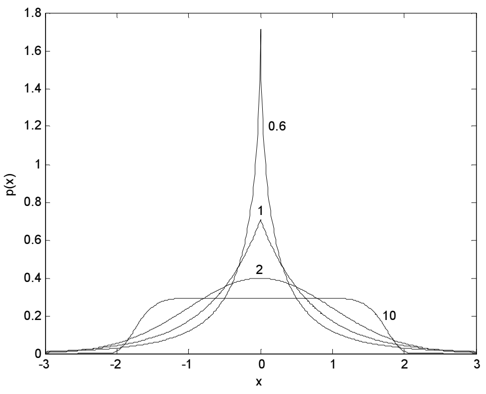

Fig. 1. The probability density function of the generalized Gaussian plotted for different values of the shape. parameter v = 0.6, 1, 2 and 10, respectively. All distributions are normalized to unit variance (σ^2= 1) and have zero mean.

Many methods exist for parameter estimation of a GGD. The state of the art is summarized in the work by Varanasi and Aazhang [13], which focuses on evaluation of accuracy of estimates for both large and small samples, and compares classic statistical methods, like the moment method, the maximum likelihood (ML), and the moment estimator. The ML method turned out to be the most efficient for $v < 1$. Its theoretical formulation [13, 10], and numerical solution, however, make it unsuitable for real-time applications.

Here, we use the scheme first proposed by Mallat [14] to implement the parametric estimation of GGD. Mallat's method substantially falls into the category of moment methods, being based on matching the moments of the data set with those of the assumed distribution, in order to find out its variance value and shape factor. For a GGD, the ratio of mean absolute value to standard deviation is a steadily increasing function of the *v*:

$$F_M(v) = \frac{\Gamma(2/v)}{\sqrt{\Gamma(1/v)\cdot\Gamma(3/v)}}. \tag{20}$$

Given a group of N i.i.d. zero-mean random variables, $\{x_1, x_2, ..., x_N\}$, following a GGD $p_v(x)$, let $m_1 = \frac{1}{N}\sum_{i=1}^{N} |x_i|$ be the estimate of the mean absolute value and $\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N} x_i^2$ the sample variance. The parameter v is estimated by inverting (20), that is by solving:

$$\hat{v} = F_M^{-1}\left(\frac{m_1}{\hat{\sigma}}\right). \tag{21}$$

4. GGD based Gaussian particle filtering – an extension of the GPF

Applying GGD to improve the performance of the Gaussian particle filter is rather routine if we have got an estimate of *v* from (21). From the description in Section 2, we give an extension algorithm of the Gaussian particle filter as follows. The density function chosen is the only difference in algorithm between the extension of the GPF and the GPF; in the former, it is GGD and in the latter, it is Gaussian density.

The extension algorithm of the GPF – measurement update algorithm.

1) Draw samples from the importance function $\pi(x_t \mid y_{0:t})$ and denote them as $\{x_t^{(j)}\}_{j=1}^{M}$.
2) Obtain the respective weights by:

$$\overline{w}_t^{(j)} = \frac{p(y_t \mid x_t^{(j)})\, p_v(x_t = x_t^{(j)}; \overline{\mu}_t, \overline{\Sigma}_t)}{\pi(x_t^{(j)} \mid y_{0:n})}, \tag{22}$$

where the $\overline{\mu}_t$ is the mean, and the covariance is the positive definite matrix $\overline{\Sigma}_t$. $p_v(x)$ is the Generalized Gaussian distribution.

3) Normalize the weight as:

$$w_t^{(j)} = \frac{\overline{w}_t^{(j)}}{\sum_{j=1}^{M} \overline{w}_t^{(j)}}. \tag{23}$$

4) Estimate the mean and covariance by:

$$\begin{aligned} \mu_t &= \sum_{j=1}^{M} w_t^{(j)} x_t^j, \\ \Sigma_t &= \sum_{j=1}^{M} w_t^{(j)} (\mu_t - x_t^j)(\mu_t - x_t^j)^H. \end{aligned} \tag{24}$$

The extension algorithm of the GPF – time update algorithm.

1) Draw samples from $p_v(x_t, \mu_t, \Sigma_t)$ and denote them as $\{x_t^{(j)}\}_{j=1}^M$.
2) For $j = 1,...,M$, , sample from $p(x_{t+1} \mid x_t = x_t^j)$ to obtain $\{x_{t+1}^{(j)}\}_{j=1}^M$.
3) Compute the mean $\bar{\mu}_{t+1}$ and covariance $\bar{\Sigma}_{t+1}$ as:

$$\begin{aligned} \bar{\mu}_{t+1} &= \frac{1}{M}\sum_{j=1}^{M} x_{t+1}^j, \\ \bar{\Sigma}_{t+1} &= \sum_{j=1}^{M} (\bar{\mu}_{t+1} - x_{t+1}^j)(\bar{\mu}_{t+1} - x_{t+1}^j)^H . \end{aligned} \tag{25}$$

5. Experimental tests of an extension of the GPF algorithm

The extension of the GPF proposed in this paper is applied now to two different systems. In Example 1, a nonlinear time series model is chosen which comes from economy. Here the measurement noise is considered to be Gaussian noise. And in Example 2, another nonlinear model, which has been used extensively in the literature for benchmarking numerical filtering techniques [15], is presented in the presence of Gaussian noise. The performance of the proposed algorithm was compared with a conventional SIS filter (particle filter) and GPF.

5.1. Case 1

Here, we present the result for one model: the short-term variations and long-term dynamics model [16], which is popularly used in econometrics. We choose this model because it is highly nonlinear. The DSS equations for this model can be written as:

$$\begin{aligned} x_n &= a \cdot x_{n-1} + w, \\ y_n &= c \cdot \exp(x_n) + v, \end{aligned} \tag{26}$$

where $w \sim N(0, b^2)$, $v \sim N(0, d^2)$. We assume that the parameter $\theta = [a, b, c, d]$ are known quantities. We want to obtain estimates of the state process X given observations from the data vector Y.

To compare the performance of the estimation routines of the particle filter, the Gaussian particle filter and the extension of the Gaussian Particle Filter, we apply these procedures on a simple nonlinear model where $\theta = [0.6, 0.2, 2, 0.2]$. (26) is used to simulate $T = 100$ observations under θ to obtain X and Y from the time update process and measurement systems.

We apply the particle filter, Gaussian particle filter and the extension of the Gaussian particle filter using n = 100 particles to re-estimate X. We then test how well each estimation procedure fits the initial simulated unobserved time-series X.

We use the Root Mean Square Error (RMSE) to evaluate the estimation accuracy, which is defined as:

$$RMSE = \left[\frac{1}{T}\sum_{n=1}^{T}(x_n - \hat{x}_n)^2\right]^{1/2}, \tag{27}$$

where $\hat{x}_n$ is the mean of particle estimates, T is the number of iterations, in time steps.

From Fig. 2, we can see that each method is able to do a reasonable job in re-estimating $X_{1:T}$, but the extension of GPF gives a better estimation of the system state than the others.

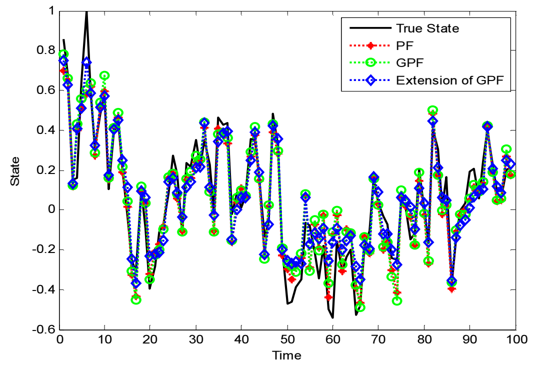

Fig. 2. Case 1: this figure compares the plots of simulated state represented by the solid black line, to that of the estimated state vector $\hat{X}$, represented by the dotted asterisk, circle and square lines. These correspond to the particle filter, Gaussian particle filter and the extension of the Gaussian particle filter with $v = 0.18$. The number of particles is 100.

From Fig. 2, we can see that each method is able to do a reasonable job in re-estimating $X_{1:T}$, but the extension of GPF gives a better estimation of the system state than the others. What is not easily seen in Fig. 2, is a comparison with respect to both performance and efficiency. We refer to Table 1 to illustrate this comparison. The extension of GPF with v having a proper estimation provides an appreciable improvement in the error when compared with both the particle filter and GPF; for example, for the case $v = 0.18$, the mean squared error (MSE) (where $MSE = RMSE^2$) of the Gaussian particle filter is 0.0089 while that of the extension of GPF is 0.0044. The extension of GPF with a proper v also outperforms a little the GPF in time cost; for example, for the extension of GPF with $v = 0.18$, the time cost is 0.0450s while that of the GPF is 0.0458s.

The MSE of each filter was subsequently computed. Fig. 3 shows the MSE of each filter with 100 particles for 50 Monte Carlo runs. The comparison of the average MSE over the 50 times samples among the filters, in terms of the number of a particle, is shown in Fig. 4. The average MSE was compared for the number of particles 50, 100 and 500, respectively. Even for a small number of particles the extension of GPF performs significantly better than GPF. An increase in the number of particles reduces the MSE further for the extension of GPF. The performance of the extension of GPF becomes very close for the number of particles 500 and 100. The performance of the extension of GPF appears to be better than the other filters taken here for comparison.

Table 1. Case 1: comparison of particle filters (3 iterations).

Filter type	1	2	3
Particle filter			
MSE	0.0117	0.0099	0.0105
Time/s	0.0375	0.0388	0.0395
Gaussian particle filter			
MSE	0.0089	0.0077	0.0073
Time/s	0.0458	0.0487	0.0472
The extension of Gaussian particle filter	$v = 0.18$	$v = 0.19$	$v = 2.2$
MSE	0.0044	0.0039	0.0049
Time/s	0.0450	0.0484	0.0470

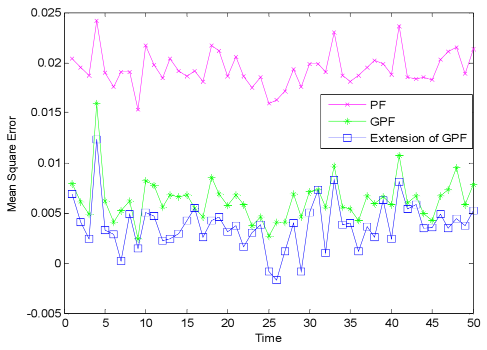

Fig. 3. Case 1: MSE for 50 Monte Carlo runs (the number of particles is 100), with $v = 0.19$ for the extension of Gaussian particle filter.

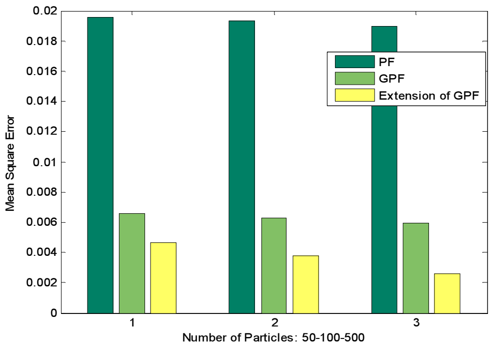

Fig. 4. Case 1: average of MSE for over time samples with the number of particles 50, 100 and 500, respectively (for 50 Monte Carlo runs). The v of the extension of Gaussian particle filter is 0.19.

5.2. Case 2

In this section we apply the extension of GPF to estimate the state of another nonlinear time series. The continuous time dynamics of the system are:

$$x_t = 0.5x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2(t-1)) + u_t, \qquad (28)$$
$$y_t = \frac{x_t^2}{20} + v_t,$$

where Gaussian distribution noise $u_t \sim N(0,1)$, $v_t \sim N(0,1)$. The simulation was carried out for 100 time samples. The performance of the extension of GPF has been compared with the PF and GPF.

From Fig. 5, it is easily seen that each method is able to do a reasonable job in estimating the state of the system, but the extension of GPF gives a better estimation of the system state than the others.

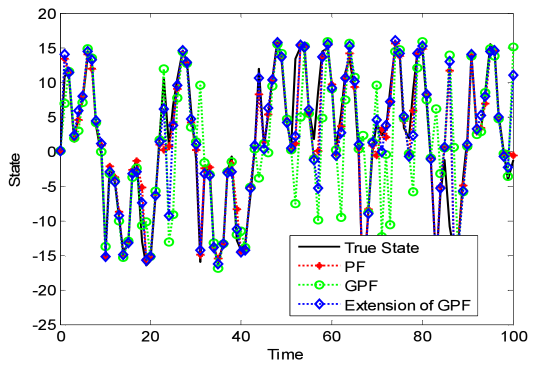

Fig. 5. Case 2: this figure compares the plots of simulated state $X_{1:T}$, represented by the solid black line, to that of the estimated state vector $\hat{X}$, represented by the dotted asterisk, circle and square lines. These correspond to the particle filter, Gaussian particle filter and the extension of the Gaussian particle filter with $v = 0.6$. The number of particles is 100.

Fig. 6 and Fig. 7 show the MSE with 100 particles and the average MSE in filtering while using a different number of particles from 50 to 500. It is clear that the estimation accuracy of the extension of GPF is the best of all even while using smaller size, especially when the particle number $N_s > 100$. This is because that it is easier to show insufficient diversity of particles when the number of particles is small, and our approach will benefit more from using the support particles which come from a more proper prior PDF $p_v(x)$.

The processing time of 1000 iterations when using the number of particle is 50 is given in Table 2. The result shows that our approach has cost nearly equal processing time compared with the GPF.

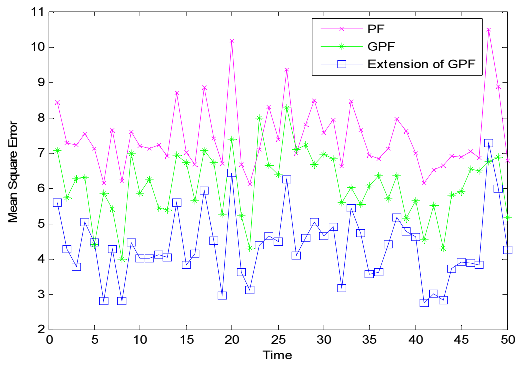

Fig. 6. Case 2: MSE for 50 Monte Carlo runs (the number of particles is 100), with *v* = 0.6 for the extension of Gaussian particle filter.

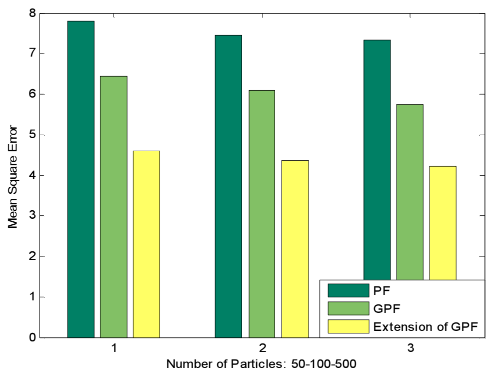

Fig. 7. Case 2: average of MSE for over time samples with the number of particles 50, 100 and 500, respectively (for 50 Monte Carlo runs). The v of the extension of Gaussian particle filter is 0.6.

Table 2. Case 2: Performance comparison of particle filters (1000 iterations).

Filter type	PF	GPF	Extension of GPF (v = 0.6)
Processing time/s	0.1565	0.5633	0.5621

6. Conclusions

This paper studies a model of state estimation based on Generalized Gaussian distributions. Effective applications in non-linear system state estimation using an extension of the

Gaussian particle filter are discussed. Simulations on estimating the system state illustrate the benefits of the extension of Gaussian particle filter against the Gaussian particle filter (GPF) method. The results compared with the Gaussian particle filter algorithm show that the extension of Gaussian particle filter algorithm has better performance of state estimation and nearly equal computational cost.

Acknowledgements

The authors would like to thank the National Natural Science Foundation of China (Grant no. 60871056) and the Specialized Research Fund for the Doctoral Program of High Education of China for their support of this research.

References

[1] Simon, J., Jeffrey, U. (1996). A general method for approximation nonlinear transformations of probability distributions, *http://www.robots.ox.ac.uk/siju/work/work.html*

[2] Kalman, E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82, 34–45.

[3] Djuric, M., Kotecha H., Zhang, J., Huang, Y., Ghirmai, T., Bugallo, F., Miguez, J. (2003). Particle filtering. *IEEE Signal Processing Magazine*, 20(5), 19–38.

[4] Kotecha, H., Djuric, M. (2003). Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10), 2592–2601.

[5] Landau, D., Lifshitz, M. (1980). *Statistical physics*. Part 1., Paris, Pergamon Press.

[6] Anderson, D., Moore, B. (1979). *Optimal filtering*. Englewood Cliffs, NJ, Prentice-Hall.

[7] Aiazzi, B., Alaparone, L., Baronti., S. (1999). Estimation based on entropy matching for generalized Gaussian PDF modelling. *IEEE Signal Proc. Lett.*, 6(6), 138–140.

[8] Birney, A., Fisher, R. (1995). On the modelling of DCT and subband image data for compression. *IEEE Trans. Image Process.*, 4(2), 186–193.

[9] Do, N., Vetterli, M. (2002). Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Trans. Image Process.*, 11(2), 146–158.

[10] Joshi, L., Fisher, R. (1995). Comparison of generalized Gaussian and Laplacian modelling in DCT image coding. *IEEE Signal Proc. Lett.*, 2(5), 81–82.

[11] Muller, F. (1993). Distribution shape of two-dimentional DCT coefficients of natural images. *Electron. Lett.*, 29(22), 1935–1936.

[12] Kokkinakis, K., Nandi, K. (2005). Exponent parameter estimation for generalized Gaussian probability density functions with application to speech modelling. *Signal Processing*, 85(2005), 1852–1858.

[13] Varanasi, K., Aazhang, B. (1989). Parametric generalized Gaussian density estimation. *J. Acoust. Soc. Amer.*, 86, 1404–1415.

[14] Mallat, G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Machine Intel.*, 7(11), 674–693.

[15] Arulampalam, S., Maskell, S., Gordon, N., Clapp, T. (2002). A tutorial on particle filters for online nonlinear/nonlinear-Gaussian Bayesian tracking. *IEEE Trans. Signal Processing*, 50(2), 174–188.

[16] Schwartz, S., Smith, J. (2000). Short-term variations and long-term dynamics in commodity prices. *Management Science*, 46(7), 893–911.