International Journal of Innovative
Computing, Information and Control
Volume 11, Number 1, February 2015
 ISSN 1349-4198
pp. 231–245

NONLINEAR BAYESIAN MODE FILTERING

Bo Liu and Ibrahim Hoteit

Division of Computer, Electrical and Mathematic Science and Engineering
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{ bo.liu; ibrahim.hoteit }@kaust.edu.sa

Received January 2014; revised May 2014

Abstract. *This work proposes a non-parametric nonlinear Bayesian mode filtering technique for estimating the state of discrete time dynamical systems. The mathematical model of the system can be evaluated for any input of interest, and the corresponding output can be obtained without any knowledge of the model internal functioning. In the proposed method, a set of weighted samples are updated by evaluating the system state transition function, and then the kernel function based non-parametric approximation of the weighted samples is used to estimate the prior probability. The natural evolution gradient of the posterior conditional probability is derived, and hence a Monte Carlo method is applied to recursively locate the mode of the posterior conditional probability. The two dimensional Van der Pol oscillator system is considered as a numerical example. The simulation results show superior performance compared to the standard Particle filter, especially in the cases with small number of particles.*
Keywords: Bayesian filtering, Mode filtering, Natural gradient, Fisher information matrix

1. **Introduction.** Bayesian filtering techniques are widely applied in many real-world applications to estimate unknown quantities given a set of noisy observations and knowledge of the system dynamics. The most popular applications include, but not limited to, tracking, extraction of signals of interest from contaminating environments, and estimating the dynamic states for regulating the systems, for example, see [1-6]. In most of the cases, the unknown quantities can be featured by a dynamic equation and are observed by a measurement equation, which together form the so-called dynamic state space model. For such a problem, one can adopt a Bayesian filtering approach, and thereby recast the problem to one of tracking a hidden process given a set of noisy observations and the knowledge of both the dynamic process and observation equations [7].

In many applications, the numerical solution of mathematical models could result in high-dimensional outputs. For instance, the dimension of the space aircraft usually is greater than ten, and the models in geophysics have much higher dimensions [8]. It is often hard or sometimes even impossible to directly manipulate those system functions, and the only thing one can do is to evaluate the system functions at some particular inputs to obtain the corresponding outputs. From the point of view of filter designer, these are forward models, and hence the estimation error covariance matrix cannot be analytically updated. Moreover, the system is not guaranteed to be linear, making the celebrated Kalman filter not suitable for such applications. This is also true for the Extended Kalman filter, which requires the derivatives of the system functions [9].

Monte Carlo filtering, like Particle filters, is a compromise choice for this kind of problems [10]. In these filters, a set of weighted samples updated by the system state transition function is used to approximate the prior probability density, and when an observation

is available, the relative likelihood of the forecast estimates are calculated to update the importance weights. The updated weighted samples are then used to approximate the posterior probability density. However, the computation effort largely depends on the number of evaluation of those functions [11]. Given a particular slot of computation time, the number of evaluations can be limited. Moreover, although the Particle filter is a fully nonlinear Bayesian filter and the weighted samples are believed to represent the posterior probability, the optimal estimate from the Particle filter, i.e., the convex combination of those samples, could be far from the true state when the number of samples involved in the estimation is small and/or the mathematical model is not accurate enough [12]. Given that the estimate of the Particle filter is the weighted sum of the forecast samples, if the true state cannot be described as a convex combination of the forecast samples, the Particle filter may not generate an adequate estimate. To tackle this problem, we develop a non-parametric probability density estimation method based on the forecast weighted samples to estimate the probability density at any point in the state space.

Given the knowledge of the prior probability at every point of the state space, the mode of the posterior probability is regarded as the optimum of an objective function, and its corresponding natural gradient is derived to design a Monte Carlo maxima searching algorithm [13]. This algorithm could be considered as a variants of the Covariance Matrix Adaptation Evolution method, which would locate the posterior estimate at any point outside of the convex space if it has higher estimated posterior probability density.

This paper is organized as follows. Section 2 describes the system to be filtered, and introduces the Bayesian formulation of the state estimation problem. Section 3 discusses the limitation of the Particle filters and introduces a non-parametric probability estimation method for weighted samples. A natural gradient based Monte Carlo searching technique is proposed in Section 4 to estimate the mode of the posterior probability. Finally, a numerical example is presented in Section 5, and the performance of the proposed algorithm is tested wherein. Section 6 concludes the paper.

2. **Problem Formulation.**

2.1. **System descriptions.**

The considered discrete time nonlinear system with additive white noises can be described as follows

$$x_{n+1} = f(x_n) + w_n \tag{1}$$

$$y_n = h(x_n) + v_n, \tag{2}$$

where $x_n \in R^{n_x}$ and $y_n \in R^{n_y}$ refer to the system state and its corresponding observation at time instant, n, n_x and n_y are the dimensions of the state x and the observation y, respectively. $w_n \sim N(0, Q)$ and $v_n \sim N(0, R)$ are the process white noise and observation white noise, respectively. Operator $f(\cdot)$ is the states transition function and $h(\cdot)$ is the observation function. They are both deterministic, and in this work they are assumed to be forward model functions, i.e., they could only be used to evaluate the output at any particular input.

In a Bayesian context, the estimation problem is to quantify the posterior density $p(x_n|Y_n)$, where Y_n is made of all observations up to the estimation time n, i.e., $Y_n = \{y_1, y_2, \cdots, y_n\}$. The Bayesian inference of state estimation involves computing

$$p(x_n|Y_n) = \frac{p(y_n|x_n)p(x_n|Y_{n-1})}{\int p(y_n|x_n)p(x_n|Y_{n-1})dx_n}, \tag{3}$$

where the prior probability $p(x_n|Y_{n-1})$ is given by

$$p(x_n|Y_{n-1}) = \int p(x_n|x_{n-1}, Y_{n-1})p(x_{n-1}|Y_{n-1})dx_{n-1}. \tag{4}$$

Here, the previous posterior density is identified as $p(x_{n-1}|Y_{n-1})$. Since the denominator of (3) is actually a constant, (3) is rewritten as

$$p(x_n|Y_n) = cp(y_n|x_n)p(x_n|Y_{n-1}), \tag{5}$$

where the normalization constant

$$c = 1 \Big/ \int p(y_n|x_n)p(x_n|Y_{n-1})dx_n.$$

The filtering problem consists of recursively estimating the state x_n given observations Y_n. There are three widely used criterion functions for estimation [14]. The first is to minimize $\int ||x_n - \hat{x}_n||p(x_n|Y_n)dx_n$, and its corresponding solution is $\hat{x}_n = E\{x_n|Y_n\}$, which is usually called the minimum variance estimate. The second is to maximize the probability of the event $(x_n = \hat{x}_n)$, and its solution is $\hat{x}_n =$ Mode of $p(x_n|Y_n)$. The last criterion is to minimize the maximum of $|x_n - \hat{x}_n|$, and its solution is $\hat{x}_n =$ Medium of $p(x_n|Y_n)$. If the posterior probability $p(x_n|Y_n)$ is Gaussian or any other symmetrical shaped distribution, then these three criterion functions coincide with each other. In this work, we consider the mode of the posterior distribution as the estimation target.

3. **Non-parametric Monte Carlo Approximation of the Prior Probability Distribution.** The mode estimate is defined as follows,

$$\begin{aligned} \check{x}_n &= \arg_{x_n} \max p(x_n|Y_n) &(6)\\ &= \arg_{x_n} \max p(y_n|x_n)p(x_n|Y_{n-1}). &(7)\end{aligned}$$

The first term in the right hand side of this equation can be easily obtained from (2) as

$$p(y_n|x_n) = N(y_n|h(x_n), R), \tag{8}$$

where the notation $N(x|\mu, \phi)$ denotes a Gaussian probability of x with mean μ and covariance ϕ,

$$N(x|\mu, \phi) = \frac{1}{(2\pi)^{n_x/2}|\phi|^{1/2}} \exp\left\{-\frac{1}{2}||x-\mu||^2_{\phi^{-1}}\right\}, \tag{9}$$

with $||x-\mu||^2_{\phi^{-1}} = (x-\mu)^T\phi^{-1}(x-\mu)$ and $|\cdot|$ denotes determinant. The second term of the right hand side of (7) can be computed from the analysis distribution of the previous step, i.e.,

$$p(x_n|Y_{n-1}) = \int N(x_n|f(x_{n-1}), Q)p(x_{n-1}|Y_{n-1})dx_{n-1}. \tag{10}$$

This can be accomplished either numerically, or analytically, from the knowledge of the studied system. In the current work, we make two important assumptions: 1) the system is a forward model system; it is not possible to manipulate the transition function $f(\cdot)$; and 2) there is no parametric assumption on $p(x_{n-1}|Y_{n-1})$. Based on these two points, we apply a Monte Carlo method to estimate this distribution. Assume a set of N weighted samples $\{x^a_{n-1,i},\ \omega_{n-1,i}\}$, $i = 1, 2, \cdots, N$ drawn from $p(x_{n-1}|Y_{n-1})$ is available, where the importance weights $\omega_{n-1,i}$ are approximations of the relative posterior probabilities (or densities) of the samples such that

$$\sum_{i=1}^{N} \omega_{n-1,i} = 1. \tag{11}$$

The forecast samples are generated by evaluating the system state transition function at each weighted sample as $x^f_{n,i} = f(x^a_{n-1,i})$. The weighted samples $\{x^f_{n,i},\ \omega_{n-1,i}\}$, $i = 1, 2, \cdots, N$ follow the distribution $p^f_n = p(x_n|Y_{n-1})$, and its corresponding mathematical expectation can be approximated as

$$E_{p^f_n}\{x_n\} = \int x_n p(x_n|Y_{n-1}) dx_n \tag{12}$$

$$\approx \sum_{i=1}^{N} \omega_{n-1,i} x^f_{n,i}, \tag{13}$$

where the subscript p^f_n of $E_{p^f_n}\{x_n\}$ refers to the prior probability $p(x_n|Y_{n-1})$. This approximation is widely used for forecast estimation in the Unscented Kalman filters, Particle filters, and if a set of uniform importance weights is considered, i.e., $\omega_{n-1,1} = \omega_{n-1,2} = \cdots = \omega_{n-1,N} = 1/N$, this approximation simplifies to the Ensemble Kalman filters [15].

The prior probability density function $p(x_n|Y_{n-1})$ is estimated from the weighted samples $\{x^f_{n,i},\ \omega_{n-1,i}\}$ as

$$p^f_n(x) \approx \frac{1}{h^{n_x}} \sum_{i=1}^{N} \omega_{n-1,i} k\left(h^{-1}\left(x - x^f_{n,i}\right)\right), \tag{14}$$

where $k(\cdot)$ denotes a kernel function and the scalar h is the bandwidth parameter, which is tuned according to the confidence in the weighted samples $\{x^f_{n,i},\ \omega_{n-1,i}\}$. This equation could be applied to estimate the prior probability at any point of interest in R^{n_x}, especially at those points located outside of the convex region spanned by the particles. We will use it to locate the mode of the posterior probability in the next section. The derivation of (14) is detailed in Appendix, where it is shown that the approximation (14) matches at least the first two moments of the prior probability.

The tuning bandwidth parameter h in (14) controls confidence in the samples are trusted. It is analogous to the variance of the samples. To make the estimation more robust to the forecast model uncertainties, it is usually better to use a large value of h, which results in a posterior estimate that is more affected by the observations.

4. **Monte Carlo Mode Searching.** Following the standard Particle filters algorithm, once the observation is available, the relative likelihood of each sample is calculated as

$$r_{n,i} = p(y_n|x^f_{n,i}), \tag{15}$$

and the importance weights are updated as

$$\hat{\omega}_{n,i} = \omega_{n-1,i} r_{n,i}, \tag{16}$$

with the normalized new set of weights

$$\omega_{n,i} = \frac{\hat{\omega}_{n,i}}{\sum_{p=1}^{N} \hat{\omega}_{n,p}}. \tag{17}$$

The forecast samples with the updated importance weights represent the posterior probability

$$\{x^f_{n,i},\ \omega_{n,i}\} \sim p(x_n|Y_n). \tag{18}$$

The first guess estimate of the mode of the posterior probability could be selected as the weighted mean:

$$\check{x}^0_n = \sum_{i=1}^{N} \omega_{n,i} x^f_{n,i}. \tag{19}$$

If the estimate of the effective number of the samples, which is computed as follows

$$\hat{N}_{eff} = \frac{1}{\sum_{L=1}^{N} \omega_{n,L}^2}, \tag{20}$$

is less than a predefined threshold N_{th}, a resampling step is applied to generate a new set of samples with uniform weights replacing the current samples.

In the Particle filters, $\check{x}_n^0$ in (19) is taken as the analysis estimate of the state at instant time n, and unbiasedness and consistency are guaranteed when the sample number tends to infinity. Particle filters are usually expected to be more accurate than other Gaussian-based filtering schemes, such as the Unscented Kalman filters and the Ensemble Kalman filters. However, when the sample number is small and/or when the forecast model is not accurate enough, the performance of the Particle filters might significantly degrade because of the collapse of the weights [16].

Suppose that a convex region ϕ_n is defined in R^{n_x} as the minimum volume including all forecast samples $x_{n,i}^f$, $i = 1, 2, \cdots, N$. Thus any convex combination of the forecast samples

$$\sum_{i=1}^{N} \omega_{n,i} x_{n,i}^f \tag{21}$$

belongs to ϕ_n, if the weights satisfy the two conditions

$$\sum_{i=1}^{N} \omega_{n,i} = 1, \quad \omega_{n,i} \geq 0. \tag{22}$$

This means that the possible estimate region is limited to ϕ_n, which is determined by the forecast samples $\{x_{n,i}^f\}$, $i = 1, 2, \cdots, N$. If the predictive model is not perfect, and/or the number of samples is small, the estimation error will always be greater than some value no matter how strongly the observation is fitted. In other words, if the true state does not belong to the predictive convex region ϕ_n, the infimum of the estimation error could be large depending on how representative the forecast samples are.

Take the two dimensional Van der Pol oscillator [17] as a visual geometrical example, suppose we have an input state x_{in}, and draw 10 random samples from the normal distribution with the input state as the mean, denoted by $\{x_k\}$, $k = 1, 2, \cdots, 10$. The true output state x_{out} is obtained by evaluating the Van der Pol equation through Liénard transformation at the true input state, which is depicted by the black crossing in Figure 1. To estimate the output state x_{out}, we consider the corresponding perturbed observation

$$y = h(x_{out}) + v,$$

where v is the observation noise and to visualize this example on one plane, we take $h(x) = x$. All the samples are updated by a one-order approximative model as follows,

$$x_k^f = f_{VDP}(x_k), \quad k = 1, 2, \cdots, 10$$

and the locations of those forecast samples are shown as blue spots in Figure 1. Obviously, in this case, since the number of the samples is small and the forecast model is imperfect, the convex combination of the forecast samples, which is the area enclosed by the green dash lines, does not include the true state. The analysis estimate of the Particle filters is always located inside the convex region. Below we propose a Monte Carlo method that extends the search for the posterior estimate beyond this area. The method is based on the evolution approach that searches directly for the mode of the posterior distribution.

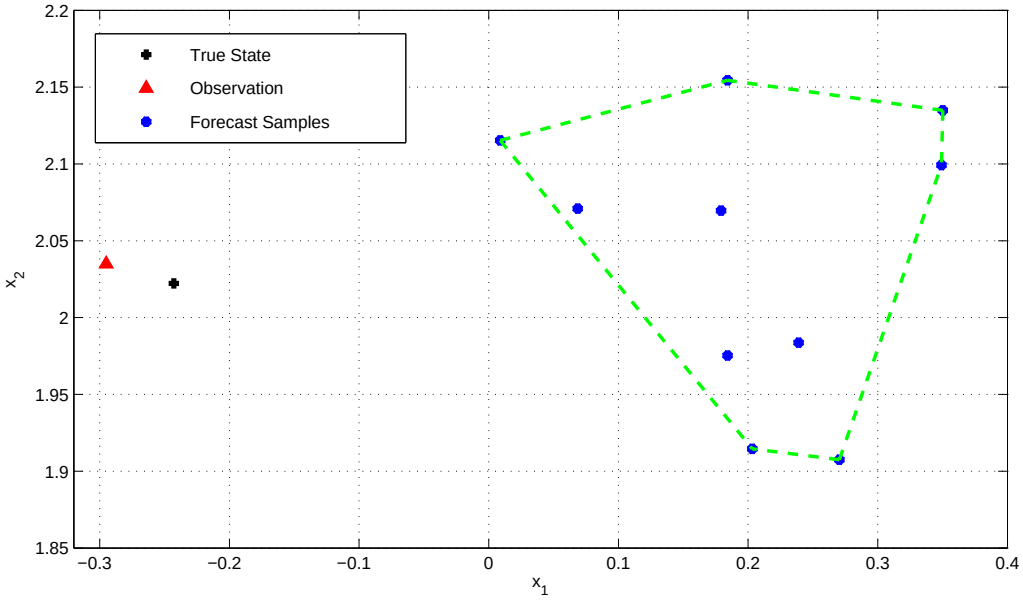

FIGURE 1. An visualized two dimensional example of the limitation of the Particle filters

Consider the objective function

$$F(x) = N(y_n|h(x), R)p(x_n = x|Y_{n-1}). \tag{23}$$

In many applications, it is not possible to compute an analytical gradient of F. To overcome this, an evolution strategy is adopted here to update the estimate in a stochastic way, while seeking for the point x that maximizes the objective function $F(x)$. We propose to search for the maxima through a Gausssian mutation approach [18]. The algorithm repeats two steps at each iteration: mutation step and update step. At the mutation step, some new points are generated from a Gaussian distribution with mean m and covariance matrix C as candidates of the maxima. At the update step, the parameters $\theta = \{m,\ C\}$ are updated to promote promising mutation by using the sample points which are generated at the mutation step.

The algorithm adjusts θ to optimize the expected fitness $J(\theta) = E\{F(x)|\theta\}$ of the next generation under the mutation distribution $N(x|\theta)$ by using the natural evolution gradient of $J(\theta)$. The expected fitness under the mutation distribution $N(x|\theta)$ is

$$J(\theta) = \int F(x)N(x|\theta)dx. \tag{24}$$

The core idea of this approach is to find, at each iteration, a small adjustment $\delta\theta$, such that the expected fitness $J(\theta + \delta\theta)$ is increased. The gradient $\nabla_\theta J(\theta)$ with respect to θ is then expressed using the log-likelihood as

$$\nabla_\theta J(\theta) = \int N(x|\theta)F(x)\nabla_\theta \ln N(x|\theta)dx. \tag{25}$$

If the gradient $\nabla_\theta J(\theta^t)$ at the current location θ^t is available, one could update θ^{t+1} by shifting θ^t in the direction of the gradient as $\theta^{t+1} = \theta^t + \alpha\nabla_\theta J(\theta^t)$, where the parameter $\alpha > 0$ is the step size. However, when the manifold of the parameter is without orthonormal linear coordinates, like curved manifolds, the natural evolution gradient could be used

instead, namely

$$\tilde{\nabla}_\theta J(\theta) = F_{IM}^{-1}(\theta)\nabla_\theta J(\theta), \tag{26}$$

where $F_{IM}^{-1}(\theta)$ is the inverse of the Fisher information matrix $F_{IM}(\theta)$ [13]. Each element of the Fisher information matrix is defined as the mathematical expectation of a second order partial derivative of the mutation probability distribution function $N(x|\theta)$ with respect to two particular parameters. If the manifold of the parameter is with orthonormal coordinates, the Fisher information matrix degenerates to the identity matrix.

Since the objective function is considered as a black-box function, the natural evolution gradient is approximated by a Monte Carlo method as [13]

$$\tilde{\nabla}_\theta J(\theta) \approx -\sum_{i=1}^{\lambda} W_{R_i} F_{IM}^{-1}(\theta) \ln N(x|\theta), \tag{27}$$

where $W_{R_i} = F(x_i)/\lambda$, and the index R_i denotes the rank of x_i among $x_1, x_2, \cdots, x_\lambda$ with respect to the evaluation of $F(x)$. That is $F(x_i)$ is the R_i-th largest value among $F(x_1), F(x_2), \cdots, F(x_\lambda)$.

Let the parameterization follows the same idea suggested by [18]:

$$\theta = [m^T vec(C)^T]^T,$$

where $vec(\cdot)$ denotes a rearranging operator from a matrix to a column vector such that $vec([v_1 v_2 \cdots v_k]) = [v_1^T v_2^T \cdots v_k^T]^T$, the gradient of the log-likelihood of $N(x|\theta)$ is

$$\nabla_\theta \ln N(x|\theta) = \begin{bmatrix} C^{-1}(x-m) \\ \frac{1}{2} vec\left(C^{-1}(x-m)(x-m)^T C^{-1} - C^{-1}\right) \end{bmatrix}.$$

Since the Fisher information matrix for a Gaussian distribution has an explicit form, the inverse of the Fisher information matrix of $N(x|\theta)$ is [19]

$$F_{IM}^{-1}(\theta) = \begin{bmatrix} C & 0 \\ 0 & 2C \bigotimes C \end{bmatrix},$$

where $\bigotimes$ denotes the Kronecker product. Therefore, the natural gradient of the log-likelihood of $N(x|\theta)$ is

$$F_{IM}^{-1}(\theta)\nabla_\theta \ln N(x|\theta) = \begin{bmatrix} (x-m) \\ vec\left((x-m)(x-m)^T - C\right) \end{bmatrix}.$$

By applying a Monte Carlo approximation to (27), the natural evolution gradient can be estimated as

$$F_{IM}^{-1}(\theta)\nabla_\theta \ln N(x|\theta) \approx \begin{bmatrix} -\sum_{i=1}^{\lambda} W_{R_i}(x-m) \\ -vec\left(\sum_{i=1}^{\lambda} W_{R_i}(x-m)(x-m)^T - \sum_{i=1}^{\lambda} W_{R_i} C\right) \end{bmatrix},$$

and the update of the next parameter is given by

$$m^{t+1} = (1-\eta)m^t + \eta \sum_{i=1}^{\lambda} W_{R_i} x_i^t$$

$$C^{t+1} = (1-\eta)C^t + \eta \sum_{i=1}^{\lambda} W_{R_i}(x_i^t - m^t)(x_i^t - m^t)^T,$$

where η is the learning rate.

The algorithm can be summarized as follows.

1) Let $m^0 = \check{x}_n^0$ and $C^0 = P_n^f$ at the initial iteration, where $\check{x}_n^0$ is the analysis estimate of the Particle filter, i.e., the weighted mean of the analysis samples and P_n^f is the weighted covariance of the analysis samples.

2) At the kth step, draw λ candidate solutions $\{x_i\}$ from a normal distribution

$$x_i \sim N(m_k, C_k), \quad i = 1, 2, \cdots, \lambda.$$

3) The objective function $F(x)$ is evaluated at the candidate solutions $\{x_i\}$, and the candidates are sorted in decreasing order

$$F(x_{1:\lambda}) \geq F(x_{2:\lambda}) \geq \cdots \geq F(x_{\lambda:\lambda}),$$

where the index $i : \lambda$ denotes the ith candidate in an order of evaluations of the objective function from the most to the least. The definition of $F(x)$ is given in (23); the first term of $F(x)$ is evaluated analytically, and the second term of $F(x)$ is evaluated by using Monte Carlo approximation (14).

4) The new mean value (also the new estimate of the maxima) is computed as

$$m^{k+1} = (1 - \eta)m^k + \eta \sum_{i=1}^{\lambda} W_{R_i} x_i^t,$$

and the mutation covariance matrix C_{k+1} is updated as

$$C^{k+1} = (1 - \eta)C^k + \eta \sum_{i=1}^{\lambda} W_{R_i} (x_i^k - m^k)(x_i^k - m^k)^T.$$

6) Check the increment of the estimate. If

$$||m^{k+1} - m^k|| < \delta,$$

where $\delta > 0$ is a predefined threshold of the stopping criterion, then take m^{k+1} as the final estimate. If not, then go to 2). The pseudo-code is depicted as follows:

Algorithm 1 The pseudo-code of the algorithm proposed

Initialization: $m^0 \leftarrow \check{x}_n^0$
Initialization: $C^0 \leftarrow P_n^f$
while $||m^{k+1} - m^k|| > \delta$ **do**
  Draw λ samples randomly from a normal distribution $x_i \sim N(m_k, C_k)$, $i = 1, 2, \cdots, \lambda$
  Evaluate $F(x_i)$, $i = 1, 2, \cdots, \lambda$
  Sort samples as $F(x_{1:\lambda}) \geq F(x_{2:\lambda}) \geq \cdots \geq F(x_{\lambda:\lambda})$
  $m^{k+1} = (1 - \eta)m^k + \eta \sum_{i=1}^{\lambda} W_{R_i} x_i^t$
  $C^{k+1} = (1 - \eta)C^k + \eta \sum_{i=1}^{\lambda} W_{R_i} (x_i^k - m^k)(x_i^k - m^k)^T$
end while

Remark 4.1. *The stochastic optimization method proposed in this section does not guarantee to achieve global convergence, meaning the convergence to stationary points independently of the starting point. In case that the objective function has two or more local modes, [20] suggested to use two thresholds for both the mutation variables and the objective function fitness to guarantee global convergence.*

5. **Simulation.** In this section, a nonlinear example is presented to demonstrate the performance of the proposed mode filtering scheme. The Van der Pol oscillator equation [17] describes the second order differential equation of position coordinate x respect to time t as follows

$$\frac{d^2x_1}{dt^2} - \rho(1 - x_1^2)\frac{dx}{dt} + x_1 = 0.$$

Applying the Liénard transformation $x_2 = x_1 - x_1^3/3 - \dot{x}_1/\rho$, where the dot indicates the time derivative, the Van der Pol oscillator can be represented in its two dimensional form [21]

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = f\left(\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}\right) \tag{28}$$

$$= \begin{bmatrix} \rho\left(x_1 - \frac{1}{3}x_1^3 - x_2\right) \\ \frac{1}{\rho}x_1 \end{bmatrix}, \tag{29}$$

where the parameter $\rho = 4$. Let $x = [x_1\ x_2]^T$. Let T denote the computation step size, the one-order approximation of the Van der Pol oscillator system is as follows

$$x(t+T) = f_{VdP}(x(t)) \tag{30}$$

$$= x(t) + Tf(x(t)), \tag{31}$$

where an incorrect parameter $\tilde{\rho} = 3.5$ is applied in this approximate model to update the forecast samples.

Figure 2 is a snapshot of one simulation. We draw 10 samples from a Gaussian distribution

$$x_j \sim N(x_{in}|P_{in}), \quad j = 1, 2, \cdots, 10, \tag{32}$$

where $x_{in} = [1\ 2]^T$ and $P_{in} = diag(0.01\ 0.01)$. Each sample is updated by the first order approximating model (30) by setting $T = 0.2$ as follows

$$x_j^f = f_{VdP}(x_j), \quad j = 1, 2, \cdots, 10. \tag{33}$$

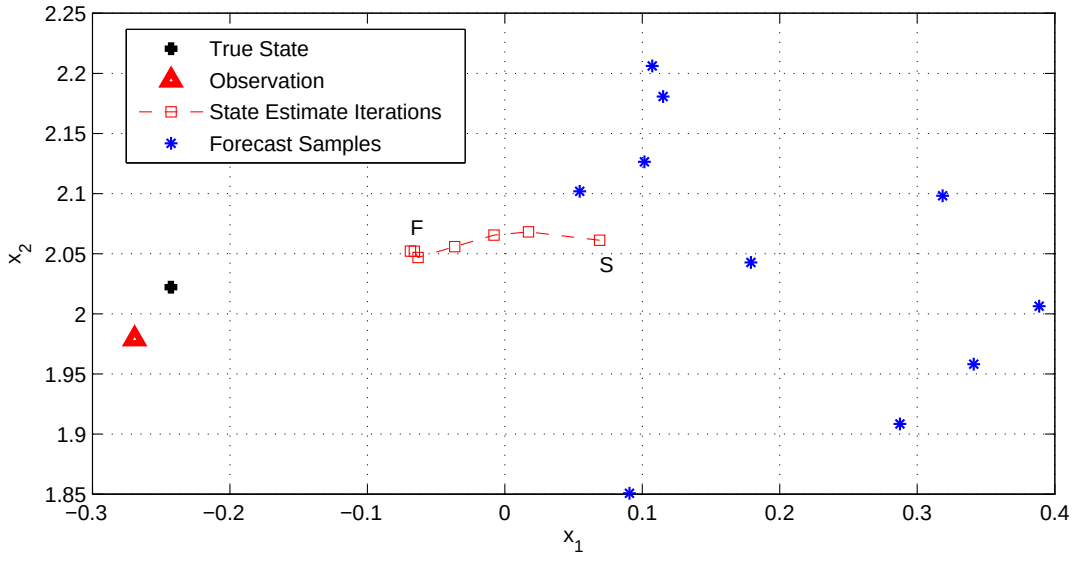

FIGURE 2. Recursive estimation from posterior estimate 'S' to a better estimate 'F'

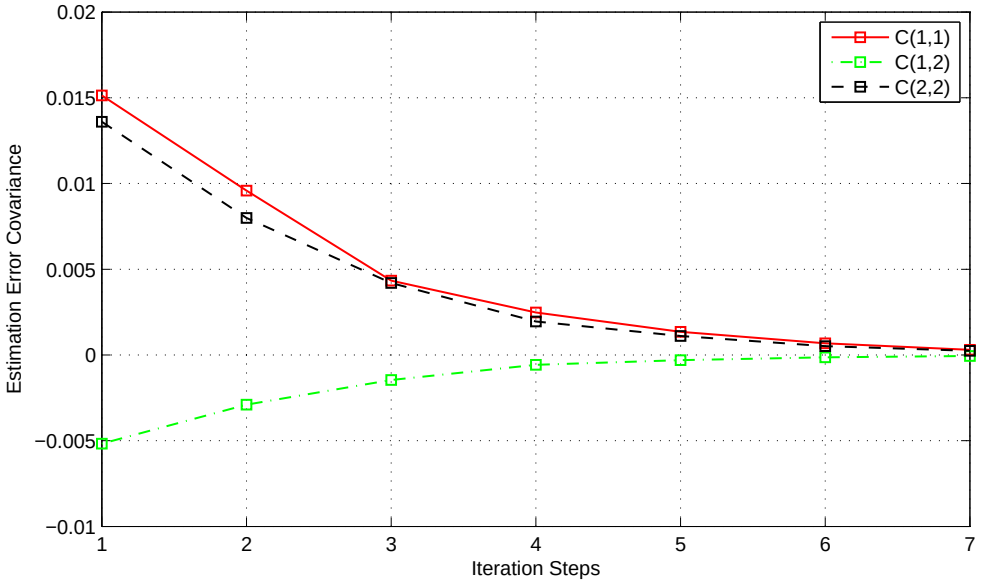

FIGURE 3. The updating of mutation covariance

The forecast samples $\{x_j^f\}$, $j = 1, 2, \cdots, 10$ are plotted in Figure 2 as blue asters. The true state x_{out} is calculated by applying the accurate model and higher order algorithm, which is denoted by the black crossing in the same figure. The noisy observation of the true state, denoted by the red triangle, is assumed to be $y = x_{out} + v$, where $v \sim N(0|R)$, and $R = diag(0.01\ 0.01)$. By applying a Particle filter, the posterior estimate of x_{out} is shown as one of the dash squares marked by 'S', which is apparently far away from the true state since both the model is not perfect and the sample size is too small. We apply the proposed method using the Gaussian kernel function

$$k(x) = (2\pi)^{-n_x/2} \exp\left\{-\frac{1}{2}||x||^2\right\} \tag{34}$$

with parameter $h = 0.3$. The initial value is set as the posterior estimate 'S'. The dash squares show the iteration results from this simulation, where 'F' denotes the stopping position. The mutation covariance C is updated at the same time and it is shown in Figure 3.

In this numerical experiment, the proposed scheme is applied for one analysis cycle but with different sample sizes. Each simulation of a particular sample size is repeated 200 times with different randomly generated samples to calculate the estimation error. The root-mean-square error (RMSE) of both the Particle filter and the proposed filter is recorded as shown in Figure 4, with

$$RMSE = \sqrt{(x_{1,out} - \hat{x}_{1,out})^2 + (x_{2,out} - \hat{x}_{2,out})^2}, \tag{35}$$

where $x_{1,out}$ and $x_{2,out}$ denote the first element and the second element of x_{out}, respectively. $\hat{x}_{1,out}$ and $\hat{x}_{2,out}$ denote the first element and the second element of $\hat{x}_{out}$, respectively. The RMSE ratio is defined as the RMSE of the Particle filter over the RMSE of the Mode filter. The results show that the proposed method could provide much better results than the Particle filter especially when the sample size is small.

A multi-steps simulation is also presented and its result is shown in Figure 5. In this simulation, the time step is $T = 0.2$, the incorrect model parameter is set to $\tilde{\rho} = 2$

and the sample size is 5. The state propagation and the observation are perturbed with additive independent Gaussian noises $w(t) \sim N(0, Q)$ and $v(t) \sim N(0, R)$, respectively. The system is then formulated as follows,

$$x(t+T) = f_{VdP}(x(t)) + w(t) \tag{36}$$

$$y(t) = x(t) + v(t), \tag{37}$$

where $Q = diag(0.1\ 0.1)$ and $R = diag(0.3\ 0.3)$. The Gaussian kernel function (34) with the bandwidth $h = 2$ is applied. This simulation runs 20 steps and both the initial state and the initial estimate follow a standard normal distribution. In the upper panel and the middle panel of Figure 5, the true states are depicted by the blue solid lines, the estimates of the states by the proposed Mode filter (MF) are depicted by the black square dash lines, and the Particle filter (PF) estimates are depicted by the red asterisk dash lines. In the bottom panel of Figure 5, the RMSEs of both the Mode filter and the Particle filter are depicted by red asterisk dash line and black square dash line, respectively. Obviously, more accurate estimates are obtained with the proposed Mode filter.

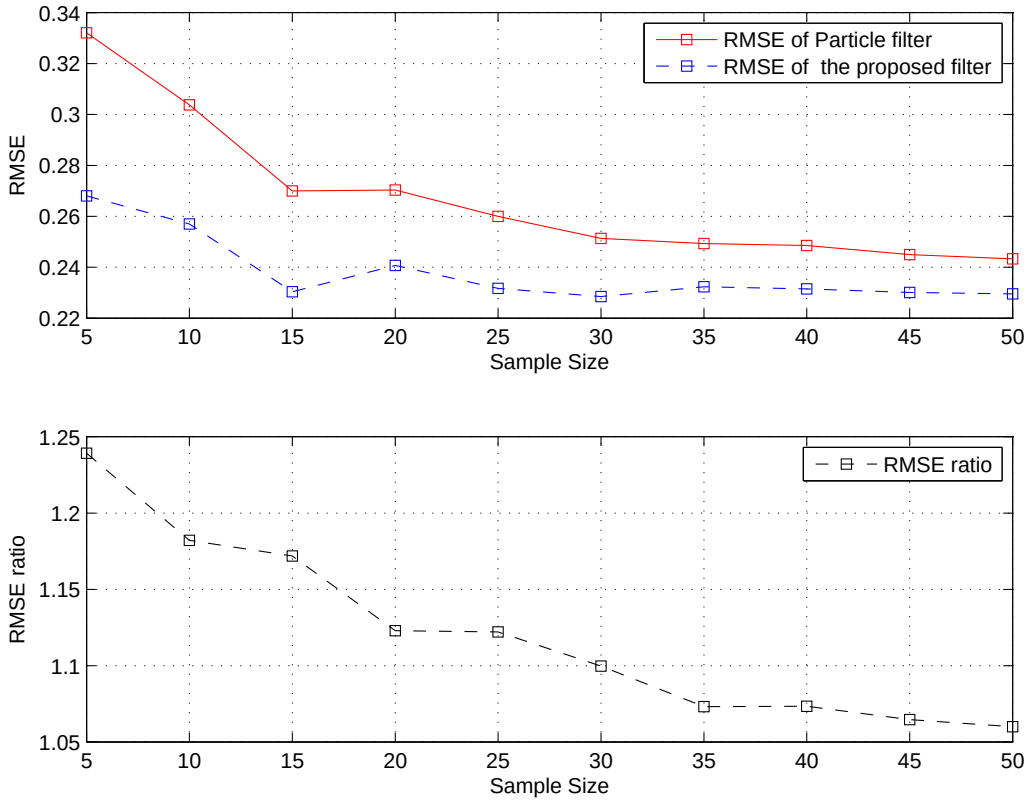

FIGURE 4. RMSE of both Particle filtering and the proposed filtering from 200 Monte Carlo Runs

6. **Conclusions.** This paper considers the filtering problem for discrete time forward model systems with additive white noises. The mode of the posterior probability is chosen as the estimation target. The proposed method follows the same technique as the Particle filters at the beginning to update a set of weighted samples by evaluating the system state transition function, and then uses a kernel function based non-parametric

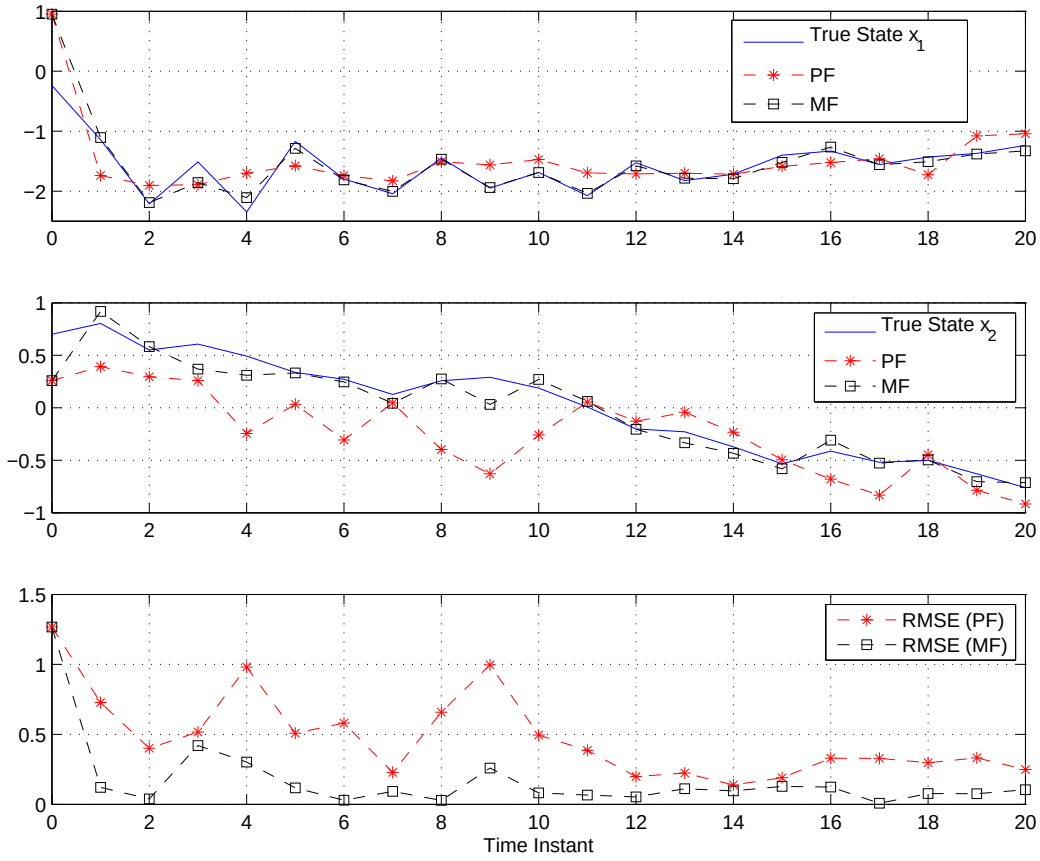

FIGURE 5. The real time estimation of Van der Pol systems. PF is short for Particle filter and MF denotes the proposed Mode filter.

approximation to estimate the prior probability from the forecast weighted samples. The natural evolution gradient of the posterior conditional probability is then derived, and an iterative Monte Carlo method is applied to locate the mode of the posterior conditional probability recursively. Particle filters generate the posterior estimate from the convex space spanned by the forecast samples, and ignore those points located outside of this convex space. The proposed method is designed to calculate the posterior estimate beyond the limited search space of the Particle filters, and this is shown to improve performances, especially for the cases with small number of particles.

Acknowledgment. Research reported in this publication was supported by the King Abdullah University of Science and Technology (KAUST).

REFERENCES

[1] P. Djuric, M. Vemula and M. Bugallo, Target tracking by particle filtering in binary sensor networks, *IEEE Transactions on Signal Processing*, vol.56, no.6, pp.2229-2238, 2008.

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

[3] P. Shi, X. Luan and C.-L. Liu, H_∞ filtering for discrete-time systems with stochastic incomplete measurement and mixed delays, *IEEE Transactions on Industrial Electronics*, vol.59, no.6, pp.2732-2739, 2012.
[4] X. Su, P. Shi, L. Wu and Y.-D. Song, A novel approach to filter design for t-s fuzzy discrete-time systems with time-varying delay, *IEEE Transactions on Fuzzy Systems*, vol.20, no.6, pp.1114-1129, 2012.
[5] D. Wang, P. Shi, J. Wang and W. Wang, Delay-dependent exponential H_∞ filtering for discrete-time switched delay systems, *International Journal of Robust and Nonlinear Control*, vol.22, no.13, pp.1522-1536, 2012.
[6] X. Su, P. Shi, L. Wu and S. K. Nguang, Induced l_2 filtering of fuzzy stochastic systems with time-varying delays, *IEEE Transactions on Cybernetics*, vol.43, no.4, pp.1251-1264, 2013.
[7] D. Yee, J. Reilly, T. Kirubarajan and K. Punithakumar, Approximate conditional mean particle filtering for linear/nonlinear dynamic state space models, *IEEE Transactions on Signal Processing*, vol.56, no.12, pp.5790-5803, 2008.
[8] X. Luo, I. Hoteit and I. Moroz, On a nonlinear Kalman filter with simplified divided difference approximation, *Physica D: Nonlinear Phenomena*, vol.241, no.6, pp.671-680, 2012.
[9] F. Gustafsson and G. Hendeby, Some relations between extended and unscented Kalman filters, *IEEE Transactions on Signal Processing*, vol.60, no.2, pp.545-555, 2012.
[10] A. J. Haug, A tutorial on Bayesian estimation and tracking techniques applicable to nonlinear and non-gaussian processes, *Technical Report*, MITRE, McLean, Virginia, US, 2005.
[11] J. Kotecha and P. Djuric, Gaussian particle filtering, *IEEE Transactions on Signal Processing*, vol.51, no.10, pp.2592-2601, 2003.
[12] B. Yang, S. Guo, N. Liu and J. Hao, Estimation with particle filter under model uncertainty, *IEEE International Conference on Signal Processing, Communications and Computing*, pp.1-5, 2011.
[13] S. I. Amari, Natural gradient works efficiently in learning, *Neural Computation*, vol.10, no.2, pp.251-276, 1998.
[14] Y. Ho and R. Lee, A Bayesian approach to problems in stochastic estimation and control, *IEEE Transactions on Automatic Control*, vol.9, no.4, pp.333-339, 1964.
[15] I. Hoteit, X. Luo and D.-T. Pham, Particle kalman filtering: A nonlinear Bayesian framework for ensemble kalman filters, *Monthly Weather Review*, vol.140, no.2, pp.528-542, 2012.
[16] I. Hoteit, D.-T. Pham, G. Triantafyllou and G. Korres, A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography, *Monthly Weather Review*, vol.136, no.1, pp.317-334, 2008.
[17] B. V. D. Pol, On relaxation-oscillations, *The London, Edinburgh and Dublin Phil. Mag. and J. of Sci.*, vol.2, no.7, pp.978-992, 1927.
[18] Y. Akimoto, Y. Nagata, I. Ono and S. Kobayashi, Bidirectional relation between CMA evolution strategies and natural evolution strategies, *Lecture Notes in Computer Science*, vol.6238, pp.154-163, 2010.
[19] S. I. Amari and H. Nagaoka, *Methods for Information Geometry*, American Mathematical Society, 2007.
[20] Y. Diouane, S. Gratton and L. N. Vicente, Globally convergent evolution strategies and CMA-ES, CERFACS, *Technical Report TR/PA/12/16*, 2012.
[21] D. Kaplan and L. Glass, *Understanding Nonlinear Dynamics*, Springer, 1995.
[22] K. Fukunaga and L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Transactions on Information Theory*, vol.21, no.1, pp.32-40, 1975.
[23] T. Hesterberg, Weighted average importance sampling and defensive mixture distributions, *Technical Report 148*, Stanford University, 1991.

Appendix. The derivation of (14) is presented in this section. It is assumed that the samples $\{x_{n,i}^f\}$ follow a probability distribution $q(x)$ with an approximating analytical form given by the sum of kernel functions [22] as

$$\hat{q}(x) = \frac{1}{N h^{n_x}} \sum_{i=1}^{N} k(h^{-1}(x - x_{n,i}^f)), \tag{38}$$

where $k(\cdot)$ is a kernel function, and h is the scalar bandwidth parameter. According to the importance sampling [23], we have

$$p(x_n = x|Y_{n-1}) = \omega_{n-1}(x)q(x). \tag{39}$$

Unfortunately, only a set of normalized evaluations of function $\omega_{n-1}(x)$ is available, thus the prior probability could only be estimated at the samples $\{x^f_{n,i}\}$, $i = 1, 2, \cdots, N$, i.e.,

$$p(x_n = x^f_{n,i}|Y_{n-1}) \approx \omega_{n-1,i}q(x^f_{n,i}). \tag{40}$$

To evaluate the prior probability $p^f_n(x) = p(x_n = x|Y_{n-1})$ at any x, we construct a set of samples from $p^f_n(x)$. First, we generate a set of integer numbers $\{D_i\}$, $i = 1, 2, \cdots, N$, which satisfy the following constraint:

$$\frac{D_i}{D} = \omega_{n-1,i}, \tag{41}$$

where $D = \sum_{i=1}^{N} D_i$, then extend the sample set $\{x^f_{n,i}\}$ to a new set

$$\{x^{df}_{n,i,j}\}, \quad i = 1, 2, \cdots, N; \quad j = 1, 2, \cdots, D_i \ \forall i, \tag{42}$$

where $x^{df}_{n,i,1} = x^{df}_{n,i,2} = \cdots = x^{df}_{n,i,D_i} = x^f_{n,i}$. In other words, there are D_i duplicates of the sample $x^f_{n,i}$ in the new sample set, and the total number of the new samples is D. This new set of samples is regarded as samples following the distribution $p(x_n|Y_{n-1})$, but with uniform weights. In the following, we will show how closely the new samples follow the prior distribution. First, the mathematic expectation of $p^f_n(x)$ is given by applying the importance sampling method in (12), and its variance is similarly obtained as follows

$$\begin{aligned} P^f_n &= \int (x_n - E_{p^f_n}\{x_n\})^2 p(x_n|Y_{n-1})dx_n && (43)\\ &= \int x^2_n \omega(x_n)q(x_n)dx_n - E^2_{p^f_n}\{x\} && (44)\\ &\approx \frac{1}{N}\sum_{i=1}^{N} \frac{(x^f_{n,i})^2 p(x_i)}{q(xi)} - E^2_{p^f_n}\{x\} && (45)\\ &= \sum_{i=1}^{N} \omega_{n-1,i}(x^f_{n,i})^2 - \left(\sum_{i=1}^{N} \omega_{n-1,i}x^f_{n,i}\right)^2. && (46) \end{aligned}$$

The average of samples $\{x^{df}_{n,i,j}\}$, $i = 1, 2, \cdots, N$, $j = 1, 2, \cdots, D_i$ is as follows

$$\begin{aligned} E\{x^{df}_{n,i,j}\} &= \frac{1}{D}\sum_{i=1}^{N}\sum_{j=1}^{D_i} x^{df}_{n,i,j} && (47)\\ &= \sum_{i=1}^{N} \omega_{n-1,i}x^f_{n,i}, && (48) \end{aligned}$$

and the variance of samples $\{x^{df}_{n,i,j}\}$ is given by

$$\begin{aligned} Var\{x^{df}_{n,i,j}\} &= E\{(x^{df}_{n,i,j})^2\} - E^2\{x^{df}_{n,i,j}\} && (49)\\ &= \sum_{i=1}^{N} \omega_{n-1,i}(x^f_{n,i})^2 - \left(\sum_{i=1}^{N} \omega_{n-1,i}x^f_{n,i}\right)^2. && (50) \end{aligned}$$

The average of the new samples (47) is identical to the expectation of the prior probability (12), and the approximated variance of the new samples (50) is identical to the

approximated variance of the prior probability (43). This means that the designed set of samples (42) matches at least the first two moments of the prior probability $p_n^f(x)$. Based on the samples (42), the prior probability at any point in R^{n_x} is estimated as follows

$$p_n^f(x) = p(x_n = x|Y_{n-1}) \tag{51}$$

$$\approx \frac{1}{Dh^{n_x}} \sum_{i=1}^{N} \sum_{j=1}^{D_i} k(h^{-1}(x - x_{n,i,j}^{df})). \tag{52}$$

Since the kernel function $k(\cdot)$ is deterministic, $\sum_{j=1}^{D_i} k(h^{-1}(x - x_{n,i,j}^{df})) = D_i k(h^{-1}(x - x_{n,i}^f))$, we have

$$p_n^f(x) \approx \frac{1}{h^{n_x}} \sum_{i=1}^{N} \omega_{n-1,i} k(h^{-1}(x - x_{n,i}^f)).$$

This equation could be applied to estimate the prior probability at any point of interest in R^{n_x}, and in particular at those points located outside of the convex region spanned by the particles.

Although samples (42) match at least the first two moments of the prior probability, the computation of the non-parametric approximation of prior probability density function (14) is directly made from the forecast samples $\{x_{n,i}^f\}$ and their corresponding weights $\{\omega_{n-1,i}\}$. Samples (42) could be considered as an intermediate set, which is only useful for analysis but not needed for the filter algorithm.