Evolution Strategies Based Particle Filters for Nonlinear State Estimation

Katsuji Uosaki*, Yuuya Kimura*, and Toshiharu Hatanaka*

* Department of Information and Physical Sciences,
Graduate School of Information Science and Technology,
Osaka University
Suita, Osaka 565-0871, Japan
(Tel: +81-6-6879-7833; Fax: +81-6-6879-7836; Email: uosaki@ist.osaka-u.ac.jp)

Abstract: Recently, particle filters have attracted attentions for nonlinear state estimation. They evaluate a posterior probability distribution of the state variable based on observations in simulation using so-called importance sampling. However, degeneracy phenomena in the importance weights deteriorate the filter performance. A new filter, Evolution Strategies Based Particle Filter, is proposed to circumvent this difficulty and to improve the performance. Numerical simulation results illustrate the applicability of the proposed idea.

Keywords: Nonlinear filtering, particle filters, evolution strategies, evolutionary computation, importance sampling, resampling, selection.

1. Introduction

Estimation of the state variables of dynamic systems using a sequence of their noisy observations is one of the crucial problems in control system science. For discrete time state space formulation of dynamic systems, difference equations are used to model the evolution of the system with time and observations are made at discrete time instants. We focus here the recursive state estimation approach, where the estimate of the state is updated as new observation comes in. This problem can be discussed within the Bayesian framework. In this approach, we first compute a posteriori probability density function (pdf) of the state based on the observations by using Bayes' law, and then find the best estimate in some sense. Well-known Kalman filter is derived by this approach as the minimum mean square error estimate based on the posterior pdf computed for linear state space model with Gaussian noise [1], [13]. However, it is generally difficult to compute analytically the posteriori pdf for nonlinear/non-Gaussian models, and some approximations should be introduced. Using the linear approximations of the nonlinear functions in system and observation equations around the estimate, we can evaluate the state estimate. This approach is called extended Kalman filter (EKF) [10], [8]. Another approach is to approximate the integrals in Bayes' rule by using some form of weighted sum based on the discrete grid. This leads to so-called "particle filtering," which approximates the integrals by Monte Carlo simulations based on the importance sampling and obtain the estimate based on the importance weights in the weighted sum [5], [2]. A common problem in the particle filter is the degeneracy phenomenon, where almost all importance weights tend to zero after some iteration. It implies a large computational effort is devoted to updating the particles with negligible weights. Some modifications such as resampling particle filter have been proposed to resolve this difficulty. Recognizing the similarities and differences of the operations in particle filters and evolution strategies [12], one of the evolutionary computation approaches, we propose here a novel evolution strategies based particle filter. Numerical simulation studies have been conducted to exemplify the applicability of this approach to nonlinear filtering.

This work is partially supported by the Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science (C)(2)14550447.

2. Particle Filters

Consider the following nonlinear state space model.

$$x_{k+1} = f(x_k, u_k, v_k) \tag{1}$$

$$y_k = g(x_k, w_k) \tag{2}$$

where x_k, u_k, y_k are the state variable, input and observation, respectively, f, g are known possibly nonlinear functions, v_k, w_k are i.i.d. system noise and observation noise sequences, respectively. We assume v_k and w_k are mutually independent. Problem to be considered here is to find the best estimate of the state variable x_k in some sense based on the all available data of observations $y_{1:k} = \{y_1, y_2, \dots, y_k\}$. We can solve the problem by calculating the posteriori probability density function (pdf) of the state variable x_k of time instant k based on all the available data of observation sequence $y_{1:k}$. For examples, using the posteriori pdf, we can obtain the minimum mean squared error estimate (MMSE) and the maximum a posteriori probability (MAP) estimate as follows.

$$\hat{x}_k = E[x_k|y_{1:k}] = \int x_k p(x_k|y_{1:k}) dx_k \tag{3}$$

$$\hat{x}_k = \arg\max_{x_k} p(x_k|y_{1:k}) \tag{4}$$

The posteriori pdf $p(x_k|y_{1:k})$ of x_k based on the observation sequence $y_{1:k}$ is evaluated recursively from a priori pdf

$p(x_0|y_0) \equiv p(x_0)$ of the initial state variable x_0 as follows.
Time evolution (Chapman-Kolmogorov equation)

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \quad (5)$$

Observation update (Bayes' rule)

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \quad (6)$$

where normalizing constant

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k$$

depends on the likelihood $p(y_k|x_k)$, which is determined by the observation equation (2).
In most cases, it is difficult to evaluate the integrals in (5) and (6) except the case f and g are linear and v_k and w_k are zero-mean Gaussian with covariances Q and R, respectively, such that

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + v_k \\ y_k &= Cx_k + w_k \end{aligned} \quad (7)$$

where we can obtain a Gaussian conditional density for the state, i.e.,

$$\begin{aligned} p(x_k|y_{1:k-1}) &\sim N(\hat{x}_{k|k-1}, P_{k|k-1}) \\ p(x_k|y_{1:k}) &\sim N(\hat{x}_{k|k}, P_{k|k}) \end{aligned} \quad (8)$$

where

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_k \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \\ P_{k|k} &= (I - K_kC)P_{k|k-1} \\ K_k &= P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1} \end{aligned} \quad (9)$$

This is the well-known Kalman filter.
In such cases, some approximations should be introduced. Using the linear approximations of the nonlinear functions in system and observation equations around the estimate, we can evaluate the state estimate. This approach is called extended Kalman filter (EKF) [10], [8]. Another approach is to approximate the integrals with the following weighted sum on the discrete grids.

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{n} w_k^i \delta(x_k - x_k^i) \quad (10)$$

where $\delta(\cdot)$ is Dirac's delta function and w_k^i is the weight for the discrete grid x_k^i with $w_k^i \geq 0$, $\sum_{i=1}^{n} w_k^i = 1$. By this approximation, MMSE (3) and MAP estimate (4) are given by

$$\text{MMSE} \qquad \hat{x}_k = \sum_{i=1}^{n} w_k^i x_k^i \quad (11)$$

$$\text{MAP} \qquad \hat{x}_k = x_k^{\arg\max_i w_k^i} \quad (12)$$

2.1. Importance Sampling

First, we briefly review the idea of "importance sampling." Consider the case where an approximation for the pdf $p(x)$ is the following function with discrete grids.

$$p(x) \approx \frac{1}{n}\sum_{i=1}^{n} w^i \delta(x - x^i)$$

with Dirac's delta functions $\delta(\cdot)$ and x^i randomly sampled according to the pdf $p(x)$. We can approximate the integral

$$I = \int g(x)p(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} g(x^i) \quad (13)$$

When it is hard to sample x^i from a general pdf $p(x)$, we find a pdf $q(x)$, from which sampling is possible, then sample x^i from it and approximate the integral by

$$\begin{aligned} E[g(x)] &= \int g(x)p(x)dx \\ &= g(x)\frac{p(x)}{q(x)}q(x)dx \\ &\approx \frac{1}{n}\sum_{i=1}^{n} w^i g(x^i) \end{aligned} \quad (14)$$

where

$$w^i = \frac{p(x^i)}{q(x^i)}$$

Here, the pdf $g(x)$ is called the importance density and is chosen to be closer as possible to the pdf $p(x)$. This sampling process is called "importance sampling."

2.2. Sequential Importance Sampling Filter

Applying this idea, the grid x_k^i, $(i = 1, \ldots, n)$ in (10) are sampled from the importance density $q(x_k|y_{1:k})$. Then the weights are given by

$$w_k^i = \frac{p(x_k^i|y_{1:k})}{q(x_k^i|y_{1:k})} \quad (15)$$

If the importance density $q(x_k|y_{1:k-1})$ is chosen to factorize such that

$$q(x_k|y_{1:k}) = q(x_k|x_{k-1}, y_{1:k})q(x_{k-1}|y_{1:k-1}) \quad (16)$$

Then we can obtain samples x_k^i by augmenting each of the existing samples x_{k-1}^i sampled from the importance density $q(x_{k-1}|y_{1:k-1})$ with the new state sampled from $q(x_k|x_{k-1}, y_{1:k})$
Noting that

$$\begin{aligned} p(x_k|y_{1:k}) &= \frac{p(y_k|x_k, y_{1:k-1})p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \\ &= \frac{p(y_k|x_k, y_{1:k-1})p(x_k|x_{k-1}, y_{1:k-1})}{p(y_k|y_{1:k-1})} \\ &\times \; p(x_{k-1}|y_{1:k-1}) \\ &= \frac{p(y_k|x_k)p(x_k|x_{k-1})}{p(y_k|y_{1:k-1})}p(x_{k-1}|y_{1:k-1}) \\ &\propto p(y_k|x_k)p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1}) \end{aligned} \quad (17)$$

we have

$$w_k^i \propto \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)p(x_{k-1}^i|y_{1:k-1})}{q(x_k^i|x_{k-1}^i, y_{1:k})q(x_{k-1}^i|y_{1:k-1})}$$
$$= w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_{1:k})} \quad (18)$$

Summarizing these steps, we can obtain a particle filter shown in Fig.1. This filter is called "Sequential Importance Sampling Particle Filter" (SIS).

Procedure *SIS*

For $k = 0$
- $i = 1, \dots, n$, sample $x_0^i \sim q(x_0|y_0)$;
- $i = 1, \dots, n$, evaluate the weight $w_0^i = p(y_0|x_0^i)p(x_0^i)/q(x_0^i|y_0)$.

For $k \geq 1$
- $i = 1, \dots, n$, sample $x_k^i \sim q(x_k|x_{k-1}^i, y_{1:k})$;
- $i = 1, \dots, n$, evaluate the weight $w_k^i = w_{k-1}^i \frac{(p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_k^i, y_{1:k}))}$;
- $i = 1, \dots, n$, normalize the weight $\tilde{w}_k^i = w_k^i / \sum_{i=1}^n w_k^i$.
- Let $p(x_k|y_{1:k}) \approx \sum_{i=1}^n \tilde{w}_k^i \delta(x_k - x_k^i)$.

Fig. 1. Algorithm of SIS filter

2.3. Sampling Importance Resampling Filter

A common problem in SIS filter is the degeneracy phenomenon, where almost all particles will be almost zero after a few iterations. By this degeneracy, a large computational effort is devoted to updating particles whose contribution to the approximation of the posterior pdf $p(x_k|y_{1:k})$ is negligible. In order to prevent this phenomenon, we can introduce resampling process, where particles with smaller weights are eliminated and particles with relatively larger weights are resampled. The resampling process involves generating new grid points x_k^{*i} $(i = 1, \dots, n)$ by resampling from the grid approximation (10) randomly with probability

$$\Pr(x_k^{*i} = x_k^j) = \tilde{w}_k^j \quad (19)$$

The weights are reset to

$$w_k^{*i} = 1/n \quad (20)$$

The effective sample size defined by

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^n (\tilde{w}_k^i)^2} \quad (21)$$

is used as a measure of degeneracy, where $\tilde{w}_k^i$ is a normalized weight. Note that $1 \leq \hat{N}_{eff} \leq n$ and that $\hat{N}_{eff} = 1$ occurs when $\tilde{w}_k^j = 1$ for some j and $\tilde{w}_k^i = 0$ for all i except j, and $\hat{N}_{eff} = n$ holds when $\tilde{w}_k^1 = \tilde{w}_k^2 = \cdots = \tilde{w}_k^n$. This implies smaller N_{eff} implies severe degeneracy. Hence if $\hat{N}_{eff} < N_{thres}$ for some predetermined N_{thres}, resampling should be desirable. Particle filter with this resampling process is called "Sampling Importance Resampling Particle Filter" (SIR). (See Fig.2)

Procedure *SIR*

For $k = 0$
- $i = 1, \dots, n$, sample $x_0^i \sim q(x_0|y_0)$;
- $i = 1, \dots, n$, evaluate the weight $w_0^i = p(y_0|x_0^i)p(x_0^i)/q(x_0^i|y_0)$.

For $k \geq 1$
- $i = 1, \dots, n$, sample $\tilde{x}_k^i \sim q(x_k|x_{k-1}^i, y_{1:k})$;
- $i = 1, \dots, n$, evaluate the weight $w_k^i = w_{k-1}^i \frac{(p(y_k|\tilde{x}_k^i)p(\tilde{x}_k^i|x_{k-1}^i)}{q(\tilde{x}_k^i|\tilde{x}_k^i, y_{1:k}))}$;
- $i = 1, \dots, n$, normalize the weight $\tilde{w}_k^i = w_k^i / \sum_{i=1}^n w_k^i$.
- Evaluate $\hat{N}_{eff} = 1/\sum_{i=1}^n (\tilde{w}_k^{*i})^2$.

If $\hat{N}_{eff} > N_{thres}$
- $x_k^i = \tilde{x}_k^i$ for $i = 1, \dots, n$,

otherwise
- $i = 1, \dots, n$, sample an index $j(i)$ distributed according to discrete distribution with n elements satisfying $\Pr(j(i) = \ell) = \tilde{w}_k^\ell$ for $\ell = 1, \dots, n$;
- $i = 1, \dots, n$, $x_k^i = \tilde{x}_k^{j(i)}$, $w_k^i = 1/n$.
- Let $p(x_k|y_{1:k}) \approx \sum_{i=1}^n \tilde{w}_k^i \delta(x_k - x_k^i)$.

Fig. 2. Algorithm of SIR filter

3. Evolution Strategies Based Particle Filter

In this section, we propose a novel particle filter called "Evolution Strategies Based Particle Filter" (SIE) by some steps in "Sampling Importance Resampling Particle Filter" (SIR) and evolution strategies.

3.1. Evolutionary Computation

Evolutionary computation approach is computational models of natural evolutionary processes as key elements in the design and implementation of computer-based problem solving systems. A variety of evolutionary computation approaches such as 'Evolutionary Programming' (EP) [7], 'Evolution Strategies' (ES) [12], 'Genetic Algorithm' (GA) [9], and 'Genetic Programming' (GP) [11] have been proposed and studied. Extensive survey and comments are given in [4],[3], [6]. The common conceptual base is simulating the evolution of individual structures via processes of selection and perturbation. These processes depend on the perceived performance (fitness) of the individual structures as defined by the environments.

Evolutionary computation approach maintains a population of structures that evolve according to rules of selection and other operators, such as recombination and mutation. Each individual is evaluated, receiving a measure of its fitness in the environment. *Selection (reproduction)* focuses attention on high-fitness individuals, thus exploiting the available fitness information. *Recombination* (now commonly known as *crossover*) and *mutation* perturb those individuals, providing general heuristics for exploration. Figure 3 outlines a basic evolutionary computation approach.

Here we explain ES briefly. Rechenberg and Schwefel [12], developed evolution strategies to solve hydrodynamic problems. It is applied to continuous function optimization in real-valued n-dimensional space. Mutation affects much more to the solution rather than crossover. The simplest method can be implemented as follows: Let $\chi^{(k)} = (x_1^{(k)}, \cdots x_n^{(k)}) \in R^n$, $(k = 1, \cdots, \mu)$ be each individual in the population.

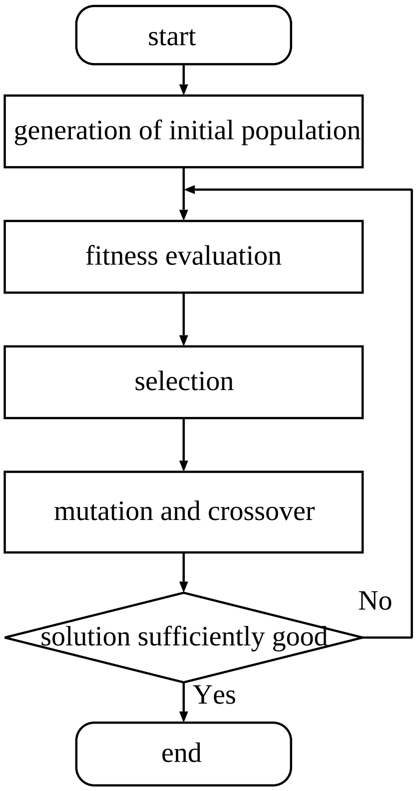


Fig. 3. Evolutionary computation approach

3.1.1 Generation of initial population

We generate an initial population of parent vectors $\{\chi^{(k)},\ (k = 1, \cdots, \mu)\}$ randomly from a feasible range in each dimension.

3.1.2 Evolution operations

1. Crossover

This process allows for mixing of parental information while passing it to their descendants. A typical crossover rule is

$$x'_j = x_{S,j} + \chi \cdot (x_{T,j} - x_{S,j}) \tag{22}$$

where S and T denote two parent individuals selected at random from the population and $\chi \in [0, 1]$ is a uniform random or deterministic variable. The index j in x'_j indicates j-th component of new individuals.

2. Mutation

This process introduces innovation into the population. It is realized by following additive process,

$$\begin{aligned} \sigma'_j &= \sigma_j \exp(\tau' N(0,1) + \tau N_j(0,1) \\ x''_j &= x'_j + \sigma'_j N_j(0,1) \end{aligned} \tag{23}$$

Here, $N(0, 1)$ denotes a realization of normal random variable with mean and unit variance, $N_j(0, 1)$ denotes random variable sampled anew for counter j of normal random variable with mean and unit variance and σ_j denote the mean step size. The factor τ and τ' are suggested to set as follows.

$$\tau \propto \left(\sqrt{2\sqrt{n}}\right)^{-1}, \qquad \tau' \propto \left(\sqrt{2n}\right)^{-1} \tag{24}$$

The factors τ and τ' are chosen dependent on the size of population μ. In this approach, small variations are much more frequent than larger variations, expressing the state of affairs on the phenotypic level in nature.

3. Selection

This is the completely deterministic process choosing the individuals of higher fitness out of the union of parents and offspring or offspring only to form the next generation in order to evolve towards better search region.

- $(\mu + \lambda)$-selection

This creates λ offspring from μ parents and selected the μ best individuals out of the union of parents and offspring.

- (μ, λ)-selection

This creates λ offspring from μ parents and selected the μ best individuals out of offspring $(\lambda \geq \mu)$.

3.2. Evolution Strategies Based Filter

Recognizing the fact the importance sampling and resampling processes in SIR filter are corresponding to mutation and selection processes in ES, we will propose a novel particle filter, Evolution Strategies Based particle filter. In SIR filter, the importance sampling process samples x_k^i according to the importance density $q(x_k^i|x_{k-1}^i, y_{1:k})$, and this corresponds to mutation process in ES from the viewpoint of generating offspring x_k^i from the parents x_{k-1}^i with extrapolation by $f(x_{k-1})$ and perturbation by v_k. On the other hands, resampling process in SIR filter selects offspring with probability

$$w_k^i \propto \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)p(x_{k-1}^i|y_{1:k-1})}{q(x_k^i|x_{k-1}^i, y_{1:k})q(x_{k-1}^i|y_{1:k-1})}$$

and this corresponds to selection process in ES with fitness function w_k^i. The main difference is resampling in SIR is carried out randomly and the weights are reset as $1/n$, while the selection in ES is deterministic and the fitness function is never reset. Hence, by replacing the resampling process in SIR by the deterministic selection process in ES, we can derive a new particle filter as follows. Based on the particles x_{k-1}^i, $(i = 1, \ldots, n)$ sampled from the importance density $q(x_{k-1}|y_{1:k-1})$, we generates p $x_k^{i(j)}$, $(j = 1, \ldots, p)$ sampled from the importance density function $q(x_k|x_{k-1}^i, y_{1:k})$. Corresponding weights $w_k^{i(j)}$ are evaluated by

$$w_k^{i(j)} = w_{k-1}^i \frac{p(y_k|x_k^{i(j)} p(x_{k-1}^{i(j)}|x_{k-1}^i)}{q(x_k^{i(j)}|x_{k-1}^i, y_{1:k})}$$
$$i = 1, \ldots, n, \; j = 1, \ldots, p \qquad (25)$$

From the set of np particles and weights $\{x_k^{i(j)}, w_k^{i(j)}, \; (i = 1, \ldots, n, \; j = 1, \ldots, p)\}$, we choose n sets with the larger weights, and set as $x_k^i, w_k^i (i = 1, \ldots, n)$. This process corresponds to (n, np)-selection in ES. We call this particle filter using (n, np)-selection in ES as Evolution Strategies Based Particle Filter (SIE). The algorithm is summarized in Fig.2.

Procedure *SIE*

For $k = 0$
- $i = 1, \ldots, n$, sample $x_0^i \sim q(x_0|y_0)$;
- $i = 1, \ldots, n$, evaluate the weight
  $w_0^i = p(y_0|x_0^i)p(x_0^i)/q(x_0^i|y_0)$.

For $k \geq 1$
- $i = 1, \ldots, n$ and $j = 1, \ldots, p$
  sample $\tilde{x}_k^{i(j)} \sim q(x_k|x_{k-1}^i, y_{1:k})$;
- $i = 1, \ldots, n$ and $j = 1, \ldots, p$,
  evaluate the weight
  $w_k^{i(j)} = w_{k-1}^i \frac{(p(y_k|\tilde{x}_k^{i(j)})p(\tilde{x}_k^{i(j)}|x_{k-1}^i)}{q(\tilde{x}_k^i|\tilde{x}_k^i, y_{1:k}))}$.
- Sort the set of pairs $\{\tilde{x}_k^{i(j)}, w_k^{i(j)} \; (i = 1, \ldots, n, j = 1, \ldots, p)\}$ by the size of $w_k^{i(j)}$ in descending order.
- Take the first n x_k^i from the ordered set $\{\tilde{x}_k^{i(p)}, \tilde{w}_k^{i(p)}\}$.
- $i = 1, \ldots, n$, normalize the weight
  $w_k^i = w_k^i / \sum_{i=1}^n w_k^i$.
- Let $p(x_k|y_{1:k}) \approx \sum_{i=1}^n \tilde{w}_k^i \delta(x_k - x_k^i)$

Fig. 4. Algorithm for SIE filter

4. Numerical Example

Consider the following nonlinear state space model.

$$x_k = f_k(x_{k-1}, k) + v_{k-1} \qquad (26)$$

$$y_k = \frac{x_k^2}{20} + w_k \qquad (27)$$

where

$$f_k(x_{k-1}, k) = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1 + x_{k-1}^2} + 8\cos(1.2k)$$

and v_k and w_k are i.i.d. zero-mean normal random variates with variance $Q_k = 10$ and $R_k = 1$, respectively. or equivalently,

$$p(x_k|x_{k-1}) = \mathcal{N}(x_k; f_k(x_{k-1}, k), Q_{k-1}) \qquad (28)$$

$$p(y_k|x_k) = \mathcal{N}\left(y_k; \frac{x_k^2}{20}, R_k\right) \qquad (29)$$

The sample behaviors of the estimates by SIS ($n = 400$), SIR ($n = 200, N_{eff} = 200$) and proposed SIE ($n = 20, p = 20$) filters are given in Fig.5. Mean squared errors at time instant $k = 100$ for 10 simulations is shown in Table4.

Table 1. Mean square errors

SIS	133.44
SIR	48.87
SIE	61.21

Though the results indicate that SIE filter shows the intermediate performance between SIS and SIR filters, it can be carried out routinely since evaluation of the effective number and comparison with the threshold value as in SIR are not necessary. The performance of course depends on the choice of design parameters n, N_{eff} and p and better choice will be pursued.

5. Conclusions

Recognizing the similarity and the difference between the importance sampling and resampling process in SIR filter and mutation and selection processes in ES, we propose a novel particle filters, SIE filter, by substituting (μ, λ)-selection in ES into resampling process in SIR. Introducing of other evolution operations such as crossover and modification of mutation will have the potential to create high performance particle filters.

References

[1] B. D. O. Anderson and J. B. Moore: *Optimal Filtering*, Prentice-Hall, 1979.

[2] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp: "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. SP-50, No.2, pp.174–188, 2002.

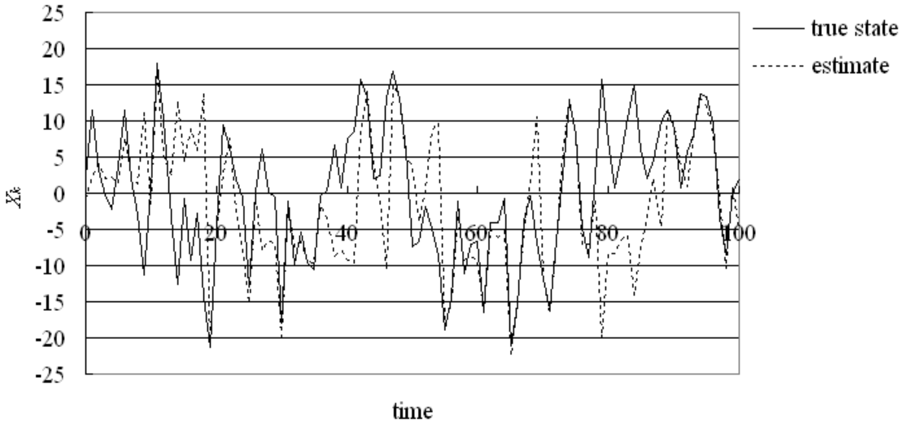
[3] T. Bäck: *Evolutionary Computation*, Oxford Press, 1996.

[4] T. Bäck and H.-P. Schwefel: "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, vol. 1, No.1, pp.1–23, 1993.

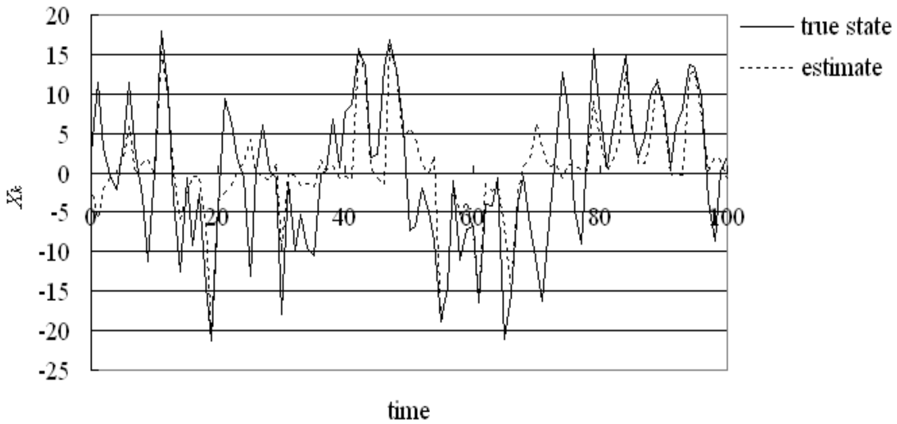
[5] A. Doucet: "On sequential simulation-based methods for Bayesian filtering," Technical Report CUED/F-INFENG/TR 310, Department of Engineering, Cambridge University, 1998.

[6] D. B. Fogel: *Evolutionary Computation*, IEEE Press, 1995.

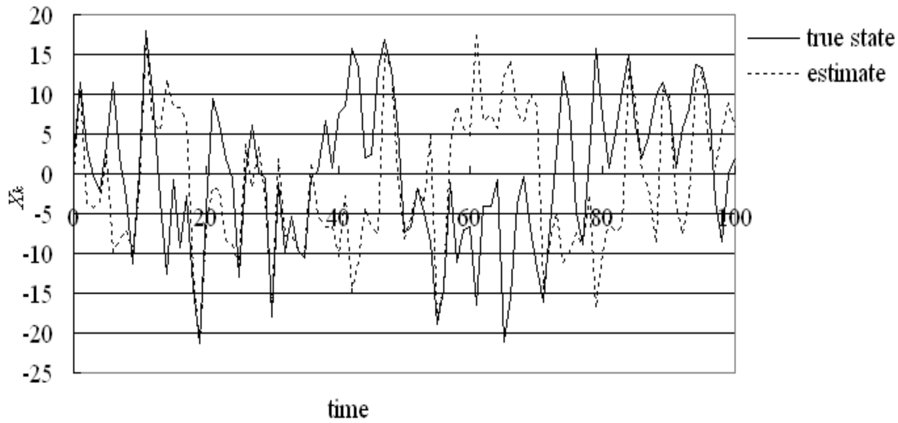
[7] L. J. Fogel, A. J. Owens and M. J. Walsh: "Artificial intelligence through a simulation of evolution," *Proc. of the 2nd Cybernetics Science Symp.*, pp.131–155, 1965, reprinted in D. B. Fogel (eds.), *Evolutionary Computation, The Fossil Record*, pp.230–254, IEEE Press, 1995.



(a) SIS



(b) SIR



(c) SIE

Fig. 5. Sample paths of state estimation

[8] G. C. Goodwin and J. C. Agüero: "State and parameter estimation for linear and nonlinear systems," *Proc. of the 7th International Conf. on Control, Automation, Robotics and Vision*, 2002.

[9] J. H. Holland: *Adaptation in Natural and Artificial Systems*, MIT Press, 1992.

[10] A. H. Jazwinski: *Stochastic Process and Filtering Theory*, Academic Press, 1970.

[11] J. R. Koza: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.

[12] H.-P. Schwefel: *Evolution and Optimum Seeking*, J. Wiley, 1995.

[13] H. Sorenson: *Kalman Filtering: Theory and Application*, IEEE Press, 1985.