Nonlinear State Estimation by Evolution Strategies Based Particle Filters

Katsuji Uosaki, Yuuya Kimura, Toshiharu Hatanaka
Department of Information and Physical Sciences,
Graduate School of Information Science and TechnologyC
Osaka University
Suita, Osaka 565-0871, Japan
uosaki@ist.osaka-u.ac.jp

Abstract- There has been significant recent interest of particle filters for nonlinear state estimation. Particle filters evaluate a posterior probability distribution of the state variable based on observations in Monte Carlo simulation using so-called importance sampling. However, degeneracy phenomena in the importance weights deteriorate the filter performance. By recognizing the similarities and the difference of the processes between the particle filters and Evolution Strategies, a new filter, Evolution Strategies Based Particle Filter, is proposed to circumvent this difficulty and to improve the performance. The applicability of the proposed idea is illustrated by numerical studies.

1 Introduction

One of the crucial problems in control system science is estimation of the state variables of dynamic systems using a sequence of their noisy observations. 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 assumed to available at discrete time instants. The focus will be on 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 ([Sorenson 1985]). In this approach, we first compute a posterior probability density function (pdf) of the state based on all available information, including the set of observations using Bayes' law, and then we find an optimal (with respect to any criterion such as the mean squared errors) estimate of the state from the posterior pdf. The well-known Kalman filter ([Kalman and Bucy 1961]) 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 ([Anderson and Moore 1979], [Sorenson 1985], [Katayama 2000]). However, it is generally difficult to compute analytically the posterior pdf for nonlinear/non-Gaussian models, and some approximations should be introduced. A number of approximation approaches have been devised. Historically, the first of these approaches was extended Kalman filter (EKF) ([Jazwinski 1970], [Goodwin and Agüero 2002]). It uses the linear approximations of the nonlinear functions in system and observation equations around the estimate, and applies the Kalman filter to obtain estimates for the state. Then, the class of filters which approximate the posterior pdf by mixture distributions was developed. Gaussian sum filter ([Sorenson 1985]) and interacting multiple model (IMM) filter ([Tugnait 1982]) are belonging to this class. Further general approach is grid-based filter, which evaluates the posterior pdf at a series of prescribed points in the sampled space based on deterministic numerical integration methods ([Sorenson 1988]). Recent massive increase of the computational power allowed to the rebirth of Monte Carlo integration and its application of Bayesian filtering, or Monte Carlo filters ([Kitagawa 1987], [West 1993], [Tanizaki and Mariano 1994]). We focus here on a class of Monte Carlo filters, known as "particle filters" ([Liu and Chen 1998], [Doucet 1998], [Doucet, de Freitas and Gordon 2001], [Arulampalam et al. 2002]). It approximates the posterior pdf by swarms of points, called 'particles,' which evolves and adapts to incoming data. Each particle has an assigned weight and the posterior pdf can be approximated by a discrete distribution which has support on each of the particles. We can see some similarities between particle filters and grid-based filters; both produce a set of weighted points and use these as a basis for an approximation of the posterior pdf. However, particles are generated randomly from the system equation and naturally follow the movement of the state, while the points of the grid-based filters are chosen arbitrarily by the user and a new choice may have to be made at each time instants to follow the moving state. Further, since the particle filters use Monte Carlo integration while the grid-based filters evaluate the integrals in deterministic numerical methods, the computation cost in calculating the weights of particles for particle filters is considerably smaller than one in calculating the weights assigned to each grids for the grid-based filters especially for higher dimensional case.

In application of the particle filters, a common problem is the degeneracy phenomenon, where almost all importance weights tend to zero after some iteration. It implies a large computational effort is devoted to update the particles with negligible weights. Some modifications such as resampling particle filter have been proposed to resolve this difficulty. In this paper, we propose a novel particle filter, based on evolution strategies ([Schwefel 1995]), one of the evolutionary computation approaches, and show its applicability.

2 Particle Filter

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, and v_k, w_k are independently identically distributed (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 $y_{1:k} = \{y_1, y_2, \ldots, y_k\}$. We can solve the problem by calculating the posterior 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 posterior pdf, we can obtain the minimum mean squared error estimate (MMSE) and the maximum a posterior probability (MAP) estimate as follows ([Goodwin and Agüero 2002]).

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

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

The posterior 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)

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

Observation update (Bayes' rule)

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

where normalizing constant

$$p(y_k|y_{1:k-1}) = \int p(y_k|x_k) p(x_k|y_{1:k-1}) dx_k \tag{7}$$

depends on the likelihood $p(y_k|x_k)$, which is determined by the observation equation (2) and the known statistics of w_k. While, in (5), $p(x_k|x_{k-1})$ is defined by the system equation (1) and the known statistics of v_{k-1}.
In most cases, it is difficult to evaluate the integrals in (5) and (6) except the case where f and g are linear and v_k and w_k are zero-mean Gaussian with covariances Q and R, respectively, such that

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + v_k \\ y_k &= Cx_k + w_k \end{aligned} \tag{8}$$

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

$$\begin{aligned} p(x_k|y_{1:k-1}) &\sim N(\hat{x}_{k|k-1}, P_{k|k-1}) \\ p(x_k|y_{1:k}) &\sim N(\hat{x}_{k|k}, P_{k|k}) \end{aligned} \tag{9}$$

where

$$\begin{aligned} \hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_k \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \\ P_{k|k} &= (I - K_k C)P_{k|k-1} \\ K_k &= P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1} \end{aligned} \tag{10}$$

This is the well-known Kalman filter.
In such cases, some approximations should be introduced as explained in Section 1. Another approach is to approximate the integrals with the following weighted sum on the discrete grids.

$$p(x_k|y_{1:k}) \approx \sum_{i=1}^{n} w_k^{(i)} \delta(x_k - x_k^{(i)}) \tag{11}$$

where $\delta(x)$ is Dirac's delta function ($\delta(x) = 1$ for $x = 0$ and $\delta(x) = 0$ otherwise), 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

$$\text{MMSE} \qquad \hat{x}_k = \sum_{i=1}^{n} w_k^{(i)} x_k^{(i)} \tag{12}$$

$$\text{MAP} \qquad \hat{x}_k = x_k^{\arg\max_i w_k^{(i)}} \tag{13}$$

In particle filters, $x^{(i)}$, $(i = 1, \ldots, n)$ are generated randomly.

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

$$I = \int g(x)p(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} g(x^{(i)}) \tag{14}$$

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

$$\begin{aligned} E[g(x)] &= \int g(x)p(x)dx \\ &= \int g(x)\frac{p(x)}{q(x)}q(x)dx \\ &\approx \sum_{i=1}^{n} w^{(i)}g(x^{(i)}) \end{aligned} \tag{15}$$

where

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

and $x^{(i)}$, $(i = 1, \ldots, n)$ are sampled from the pdf $q(x)$, which 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 the idea of the importance sampling, if the particles $x_k^{(i)}$, $(i = 1, \ldots, n)$ in (11) are sampled from the importance density $q(x_k|y_{1:k})$, then the weights are given by

$$w_k^{(i)} \propto \frac{p(x_k^{(i)}|y_{1:k})}{q(x_k^{(i)}|y_{1:k})} \tag{16}$$

In sequential case, at each iteration, we have samples constituting an approximation to $p(x_{k-1}|y_{1:k-1})$ and want to approximate $p(x_k|y_{1:k})$ with a new set of samples y_k. When the importance density $q(x_k|y_{1:k-1})$ is chosen to factorize such that

$$q(x_k|y_{1:k}) = q(x_k|x_{k-1}, y_{1:k})q(x_{k-1}|y_{1:k-1}) \tag{17}$$

Then we can obtain samples $x_k^{(i)}$ from the importance density $q(x_k|y_{1:k})$ 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 $x_k^{(i)}$ sampled from $q(x_k|x_{k-1}, y_{1:k})$ Noting that

$$\begin{aligned} p(x_k|y_{1:k}) &= \frac{p(y_k|x_k, y_{1:k-1})p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \\ &= \frac{p(y_k|x_k, y_{1:k-1})p(x_k|x_{k-1}, y_{1:k-1})}{p(y_k|y_{1:k-1})} \\ &\times \; p(x_{k-1}|y_{1:k-1}) \\ &= \frac{p(y_k|x_k)p(x_k|x_{k-1})}{p(y_k|y_{1:k-1})}p(x_{k-1}|y_{1:k-1}) \\ &\propto p(y_k|x_k)p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1}) \end{aligned} \tag{18}$$

we have

$$\begin{aligned} w_k^{(i)} &\propto \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})p(x_{k-1}^{(i)}|y_{1:k-1})}{q(x_k^{(i)}|x_{k-1}^{(i)}, y_{1:k})q(x_{k-1}^{(i)}|y_{1:k-1})} \\ &= w_{k-1}^{(i)}\frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)}, y_{1:k})} \end{aligned} \tag{19}$$

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 $k = 0$
- $i = 1, \ldots, n$, sample $x_0^{(i)} \sim q(x_0|y_0)$
- $i = 1, \ldots, n$, evaluate the weight
  $$w_0^{(i)} = \frac{p(y_0|x_0^{(i)})p(x_0^{(i)})}{q(x_0^{(i)}|y_0)}$$

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

Figure 1: Algorithm of SIS filter

2.3 Sampling Importance Resampling Filter

A common problem in the 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 particles $x_k^{*(i)}$ $(i = 1, \ldots, n)$ by resampling from the grid approximation (11) randomly with probability

$$\Pr(x_k^{*(i)} = x_k^{(j)}) = \tilde{w}_k^{(j)} \tag{20}$$

The weights are reset to

$$w_k^{*(i)} = \frac{1}{n} \tag{21}$$

The effective sample size defined by

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{n}(\tilde{w}_k^i)^2} \tag{22}$$

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 threshold value 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 $k = 0$

$i = 1, \ldots, n$, sample $x_0^{(i)} \sim q(x_0|y_0)$

$i = 1, \ldots, n$, evaluate the weight

$$w_0^{(i)} = \frac{p(y_0|x_0^{(i)})p(x_0^{(i)})}{q(x_0^{(i)}|y_0)}$$

For $k \geq 1$

$i = 1, \ldots, n$, sample $\tilde{x}_k^{(i)} \sim q(x_k|x_{k-1}^{(i)}, y_{1:k})$

$i = 1, \ldots, n$, evaluate the weight

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{(p(y_k|\tilde{x}_k^{(i)})p(\tilde{x}_k^{(i)}|x_{k-1}^{(i)})}{q(\tilde{x}_k^{(i)}|\tilde{x}_k^{(i)}, y_{1:k})}$$

$i = 1, \ldots, n$, normalize the weight

$$\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{n} w_k^{(i)}}$$

Evaluate $\hat{N}_{eff} = \dfrac{1}{\sum_{i=1}^{n}(\tilde{w}_k^{*i})^2}$

If $\hat{N}_{eff} > N_{thres}$

$x_k^{(i)} = \tilde{x}_k^{(i)}$ for $i = 1, \ldots, n$

otherwise

For $i = 1, \ldots, n$, sample an index $j(i)$ distributed according to discrete distribution with n elements satisfying

$\Pr(j(i) = \ell) = \tilde{w}_k^{\ell}$ for $\ell = 1, \ldots, n$

For $i = 1, \ldots, n$, $x_k^{(i)} = \tilde{x}_k^{j(i)}$, $w_k^{(i)} = \dfrac{1}{n}$

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

Figure 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 recognizing the similarities in some steps of the "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) ([Fogel, Owens and Walsh 1965]), 'Evolution Strategies' (ES) ([Schwefel 1995]), 'Genetic Algorithm' (GA) ([Holland 1992]), and 'Genetic Programming' (GP) ([Koza 1992]) have been proposed and studied. Extensive survey and comments are given in ([Bäck and Schwefel 1993],[Bäck 1996], [Fogel 1995]). The common conceptual base is simulating the evolution of individuals (candidate solutions) via processes of selection and perturbation. These processes depend on the perceived performance (fitness) of the individuals 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* (also refer to 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. Evolution Strategies is developed by Rechenberg and Schwefel ([Schwefel 1995]) to solve hydrodynamic problems. It is applied to continuous function optimization in real-valued n-dimensional space. Mutation is applied more often to the solution rather than crossover. The simplest method can be implemented as follows: Let $\boldsymbol{x}^{(k)} = (x_1^{(k)}, \cdots x_n^{(k)}) \in \boldsymbol{R}^n$, $(k = 1, \cdots, \mu)$ be each individual in the population.

3.1.1 Generation of initial population

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

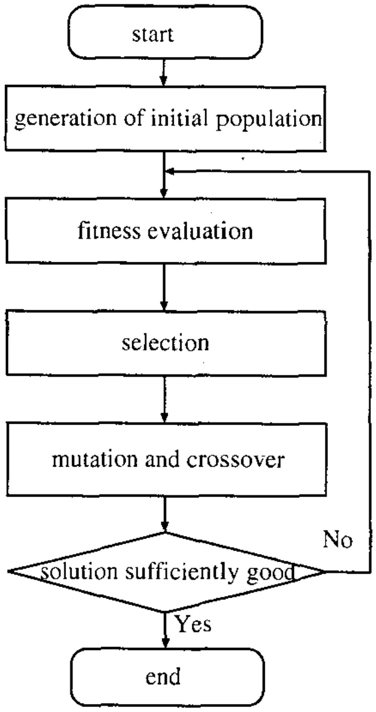


Figure 3: Evolutionary computation approach

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

$$x'_j = x_{S,j} + \chi \cdot (x_{T,j} - x_{S,j}) \tag{23}$$

   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. This is a similar operator used in differential evolution ([Storn and Price 1995])

2. Mutation

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

$$\begin{aligned} \sigma'_j &= \sigma_j \exp(\tau' N(0,1) + \tau N_j(0,1) \\ x''_j &= x'_j + \sigma'_j N_j(0,1) \end{aligned} \tag{24}$$

   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 ([Bäck and Schwefel 1993]).

$$\tau \propto \left(\sqrt{2\sqrt{n}}\right)^{-1}, \qquad \tau' \propto \left(\sqrt{2n}\right)^{-1} \tag{25}$$

   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

We will propose here a novel particle filter based on Evolution Strategies by recognizing the fact the importance sampling and resampling processes in SIR filter are corresponding to mutation and selection processes in ES. Resampling process in SIR filter selects offsprings 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 $w_k^{(i)}$. On the other hand, the importance sampling process in SIR filter 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 offsprings $x_k^{(i)}$ from the parents $x_{k-1}^{(i)}$ with extrapolation by $f(x_{k-1})$ and perturbation by v_k. 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 $\ell\, x_k^{i(j)}$, $(j = 1, \ldots, \ell)$ 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

$$w_k^{i(j)} = w_{k-1}^{(i)} \frac{p(y_k|x_k^{i(j)}p(x_{k-1}^{i(j)}|x_{k-1}^{(i)})}{q(x_k^{i(j)}|x_{k-1}^{(i)}, y_{1:k})} \quad i = 1, \ldots, n,\ j = 1, \ldots, \ell \tag{26}$$

From the set of np particles and weights $\{x_k^{i(j)}, w_k^{i(j)},\ (i = 1, \ldots, n,\ j = 1, \ldots, \ell)\}$, 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 $(n, n\ell)$-selection in ES. We call this particle filter using $(n, n\ell)$-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

$$w_0^{(i)} = \frac{p(y_0|x_0^{(i)})p(x_0^{(i)})}{q(x_0^{(i)}|y_0)}$$

For $k \geq 1$

$i = 1, \ldots, n$ and $j = 1, \ldots, \ell$,

sample $\tilde{x}_k^{i(j)} \sim q(x_k|x_{k-1}^{(i)}, y_{1:k})$

$i = 1, \ldots, n$ and $j = 1, \ldots, \ell$,

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 $\{\tilde{x}_k^{i(j)}, w_k^{i(j)}\ (i = 1, \ldots, n,\ j = 1, \ldots, \ell)\}$ by the size of $w_k^{i(j)}$ in descending order

Take the first $n\ x_k^{(i)}$ from the ordered set $\{\tilde{x}_k^{i(\ell)}, \tilde{w}_k^{i(\ell)}\}$.

$i = 1, \ldots, n$, normalize the weight

$$w_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^n w_k^{(i)}}$$

Let $p(x_k|y_{1:k}) \approx \sum_{i=1}^n \tilde{w}_k^{(i)} \delta(x_k - x_k^{(i)})$

Figure 4: Algorithm for SIE filter

4 Numerical Example

Consider the following nonlinear state space model first proposed in ([Andre Netto, Gimeno and Mendes 1978]) and discussed by several authors including ([Kitagawa 1987]).

$$x_k = f_k(x_{k-1}, k) + v_{k-1} \tag{27}$$

$$y_k = \frac{x_k^2}{20} + w_k \tag{28}$$

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,

$$p(x_k|x_{k-1}) = \mathcal{N}(x_k; f_k(x_{k-1}, k), Q_{k-1}) \tag{29}$$

$$p(y_k|x_k) = \mathcal{N}\left(y_k; \frac{x_k^2}{20}, R_k\right) \tag{30}$$

The sample behaviors of the estimates by SIS ($n = 400$), SIR ($n = 200, N_{eff} = 200$) and proposed SIE ($n = 20, \ell = 20$) filters are given in Fig.5. Mean squared errors with one standard deviation confidential interval at time instant $k = 100$ for 10 simulations is shown in Table4. 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 choices of design parameters n, N_{eff} and p and better choices of these parameters will be pursued.

Table 1: Mean square errors

SIS	133.44 ± 10.84
SIR	48.87 ± 6.96
SIE	61.21 ± 11.56

5 Conclusions

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

Acknowledgement

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.

Bibliography

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

[Andre Netto, Gimeno and Mendes 1978] M. L. Andre Netto, M. L. Gimeno and M. J. Mendes: "On the optimal and suboptimal nonlinear filtering problems for discrete time systems," *IEEE Trans. on Automatic Control*, Vol. AC-23, 1062/1067 (1978)

[Arulampalam et al. 2002] 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, 174/188 (2002)

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

[Bäck and Schwefel 1993] T. Bäck and H.-P. Schwefel: "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, Vol. 1, No.1, 1/23 (1993)

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

[Doucet, de Freitas and Gordon 2001] A. Doucet, N. de Freitas and N. Gordon (Eds.): *Sequential Monte Carlo Methods in Practice*, Springer-Verlag (2001) January 2001

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

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

[Goodwin and Agüero 2002] 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)

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

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

[Kalman and Bucy 1961] R. Kalman and R. Bucy: "New results in linear filtering and prediction theory," *J. of Basic Engineering, Trans. ASME Series D*, Vol 83, 95/108 (1961).

[Katayama 2000] T. Katayama: *Applied Kalaman Filter*, Asakaura-Shoten (2000) (*in Japanese*)

[Kitagawa 1987] G. Kitagawa: "Non-Gaussian state-space modeling of nonstationary time series," *J. of American Statistical Association*, Vol.82, 1032/1063 (1987)

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

[Liu and Chen 1998] J. Liu and R. Chen: "Sequential Monte Carlo Methods for Dynamic Systems," *J. of American Statistical Association*, Vol 93, 1032/1044 (1998).

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

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

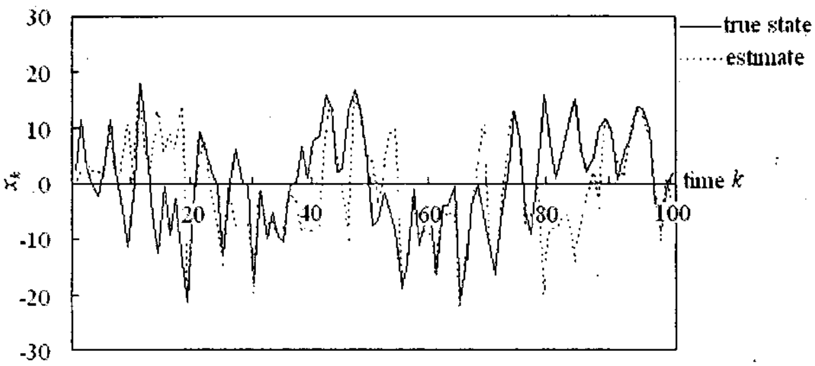
[Sorenson 1988] H. W. Sorenson: "Recursive estimation for nonlinear dynamic systems," in J. C. Spall (Ed.) *Bayesian Analysis of Time Series and Dynamic Models*, 127/165 (1988)

[Storn and Price 1995] R. Storn and K. Price: Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces, Technical Report TR-95-012, ICSI (1995)

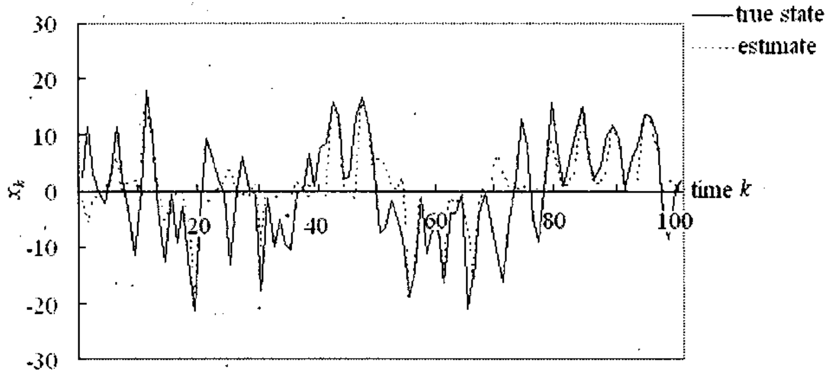
[Tanizaki and Mariano 1994] H. Tanizaki and R. S. Mariano: "Prediction, filtering and smoothing in non-linear and non-normal cases using Monte Carlo integration," *J. of Applied Econometrics*, Vol. 9, 163/179 (1994)

[Tugnait 1982] J. K. Tugnait: "Detection and estimation for abruptly changing systems," *Automatica*, Vol. 18, No.5, 607/615 (1982)

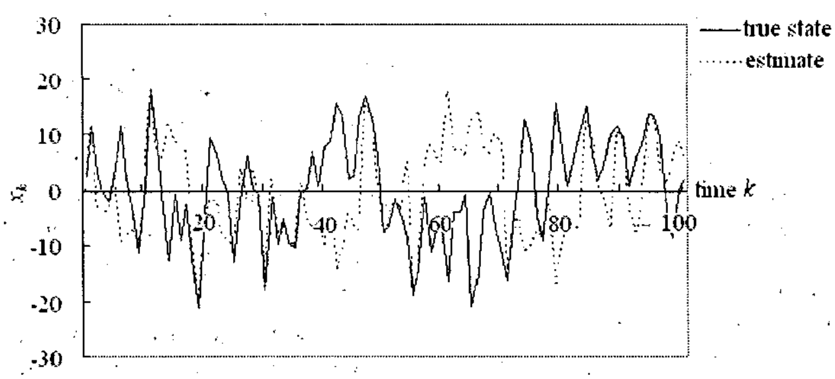
[West 1993] M. West: "Mixture models, Monte Carlo, Bayesian updating and dynamic models," *Computer Science and Statistics*, Vol. 24, 325/333 (1993)



(a) SIS



(b) SIR



(c) SIE

Figure 5: Sample paths of state estimation