Improved particle filter based on differential evolution

H.-W. Li, J. Wang and H.-T. Su

Resampling schemes for a particle filter based on the differential evolution (DE) algorithm are presented. By using these schemes, several types of differential evolution particle filters (DEPFs) are proposed. In the proposed filters, the unscented Kalman filter is utilised to generate the importance proposal distribution and the different DE algorithms are used as the resampling scheme. Simulation results demonstrate that the proposed DEPFs outperform the sequential importance resampling algorithm, the regularised particle filter, and the unscented particle filter.

Introduction: In recent years, the particle filter (PF), as an effective estimator for the nonlinear filtering problem, has been widely used in many fields, including signal processing, biostatistics, economics, and engineering. The PF is a Monte Carlo (MC) method for implementing recursive Bayesian estimation. It exploits some random particles with associated weights to approximate the true posterior density function. The basic PF goes back to the 1950s, but the PF was not used in practical applications until 1993 when Gordon *et al.* [1] proposed the sequential importance resampling (SIR) algorithm. The SIR resamples particles from a discrete approximation of the posterior density to eliminate the so-called particle degeneracy problem. But the resampling process of SIR leads to loss of diversity among the particles, i.e. the particle impoverishment problem [2]. To improve the performance of the PF, choosing a good proposal distribution, and/or modifying the resampling scheme are often adopted. For example, in [3], the extended Kalman filter (EKF) Gaussian approximation is used as the proposal distribution for a PF, and in [4], the EKF proposal is replaced by an unscented Kalman filter (UKF) proposal, and the unscented particle filter (UPF) is proposed. Musso *et al.* [5] presented the regularised particle filter (RPF). Unlike the SIR, the RPF resamples from a continuous approximation of the posterior density to alleviate the particle impoverishment problem.

In this Letter, we regard the resampling process as an optimisation problem, and present differential evolution (DE) based resampling schemes for the PF. With the use of the presented resampling schemes, four types of particle filters are proposed. We refer to them as differential evolution particle filters (DEPFs). In the DEPFs, once the particles are sampled from the UKF proposal distribution, the presented resampling schemes are adopted to generate the resampled particles.

Problem formulation: Many nonlinear filtering problems can be written in the form of the dynamic state space (DSS) model as follows:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) \text{ (system equation)} \tag{1}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{v}_k) \text{ (observation equation)} \tag{2}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ are the state variable and observation at time k, respectively. $f(\bullet)$ and $h(\bullet)$ are some known functions, system noise $\boldsymbol{u}_{k-1}$ and observation noise $\boldsymbol{v}_k$ are random vectors of given distributions. The DSS model represents the time-varying dynamics of an unobserved $\boldsymbol{x}_k$, as the distribution $p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})$. The distribution $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ represents the observation equation conditioned on the unknown $\boldsymbol{x}_k$, which is to be estimated. The system equation represents a system evolving with time k, and the initial state is given by distribution $p(\boldsymbol{x}_0)$. The objective is to estimate recursively in time the filtering distribution $p(\boldsymbol{x}_k \mid \boldsymbol{z}_{0:k})$ and the predictive distribution $p(\boldsymbol{x}_{k+1} \mid \boldsymbol{z}_{0:k})$ at time k given all the observations $\boldsymbol{z}_{0:k} \equiv \{\boldsymbol{z}_0, \ldots, \boldsymbol{z}_k\}$.

Proposed resampling schemes and particle filters: PFs rely on importance sampling, so the design of proposal distributions that can approximate the posterior distribution well is very important [4]. To utilise fully the current observation information, as has been done in [4], the UKF is utilised to generate the proposal distribution. Let $\{\boldsymbol{x}_k^{(i)} : i = 1, \ldots, N\}$ denote the state particles at time k, where N is the number of particles. Define $\boldsymbol{x}_{0:k} \equiv \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_k\}$. Let $\{\hat{\boldsymbol{x}}_k^{(i)} : i = 1, \ldots, N\}$ be the particles drawn from the UKF proposal distribution $q(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{z}_{1:k})$, i.e. $\hat{\boldsymbol{x}}^{(i)} \sim q(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{z}_{1:k}) = \mathcal{N}(\bar{\boldsymbol{x}}_k^{(i)}, \hat{\boldsymbol{P}}_k^{(i)})$, where $\mathcal{N}(\bar{\boldsymbol{x}}_k^{(i)}, \hat{\boldsymbol{P}}_k^{(i)})$ is the Gaussian distribution (with mean $\bar{\boldsymbol{x}}_k^{(i)}$ and covariance $\hat{\boldsymbol{P}}_k^{(i)}$) obtained by using UKF. In this Letter, because of space limitations, we do not show the computation process of the mean and covariance matrix of the UKF proposal distribution, the detailed computation process can be found in [4]. The weight of $\hat{\boldsymbol{x}}_k^{(i)}$ is calculated by

$$\boldsymbol{w}_k^{(i)} \propto \boldsymbol{w}_{k-1}^{(i)} \frac{p(\boldsymbol{z}_k \mid \hat{\boldsymbol{x}}_k^{(i)}) p(\hat{\boldsymbol{x}}_k^{(i)} \mid \boldsymbol{x}_{k-1}^{(i)})}{q(\hat{\boldsymbol{x}}_k^{(i)} \mid \boldsymbol{x}_{0:k-1}^{(i)}, \boldsymbol{z}_{1:k})} \tag{3}$$

normalising the weight as $\tilde{\boldsymbol{w}}_k^{(i)} = \boldsymbol{w}_k^{(i)} / \sum_{i=1}^{N} \boldsymbol{w}_k^{(i)}$. Then, a resampling process will be executed to obtain a new set of particles denoted by $\{\dot{\boldsymbol{x}}_k^{(i)} : i = 1, \ldots, N\}$ from $\{(\hat{\boldsymbol{x}}_k^{(i)}, \tilde{\boldsymbol{w}}_k^{(i)} : i = 1, \ldots, N\}$.

As stated above, in this Letter, the resampling process is treated as an optimisation problem, and effective resampling schemes based on the DE algorithm are presented. DE is a population-based stochastic algorithm for global optimisation, which has earned a reputation as a very effective global optimiser [6]. In the presented resampling schemes, the particles $\{\hat{\boldsymbol{x}}_k^{(i)} : i = 1, \ldots, N\}$ are regarded as the target vectors of the current population, and the corresponding weights $\{\tilde{\boldsymbol{w}}_k^{(i)} : i = 1, \ldots, N\}$ as the objective functions of the target vectors, respectively. Unlike the traditional resampling schemes, there is no elimination and replication of particles, the DE resampling schemes recombine the particles by using an iterative process of mutation, crossover and selection. As a result, a new set of diverse particles will be propagated.

Let $\{\boldsymbol{E}_{x,g} = (\boldsymbol{X}_{l,g}) : l = 1, \ldots, L, g = 1, \ldots, G\}$ denote the current population of DE, which consists of L D-dimensional target vectors $\{\boldsymbol{X}_{l,g} = (X_{j,l,g}) : j = 1, \ldots, D\}$, where g denotes the number of generations, and j the dimensionality. We regard the particles $\{\hat{\boldsymbol{x}}_k^{(i)} : i = 1, \ldots, N\}$ as the target vectors $\{\boldsymbol{X}_{l,g} : l = 1, \ldots, L\}$ of the current population, and the corresponding weights as the objective functions $\{f(\boldsymbol{X}_{l,g}) : l = 1, \ldots, L\}$, in this case $N = L$.

Then a mutation step will be executed. In this step, the perturbation of a base vector $\boldsymbol{X}_{r1,g}$ by using a difference vector based mutation $\boldsymbol{V}_{l,g} = \boldsymbol{X}_{r1,g} + F \times (\boldsymbol{X}_{r2,g} - \boldsymbol{X}_{r3,g})$ creates a mutation vector $\{\boldsymbol{V}_{l,g} = (V_{j,l,g}), j = 1, \ldots, D\}$, where $F \in (0,1+)$ is the scale factor that controls the rate at which the population evolves. Once the $\boldsymbol{V}_{l,g}$ is obtained, its objective function is calculated as $f(\boldsymbol{V}_{l,g}) = \hat{f}(\boldsymbol{V}_{l,g}) / \sum_l^L \hat{f}(\boldsymbol{V}_{l,g})$, where $\hat{f}(\boldsymbol{V}_{l,g}) = \dfrac{p(\boldsymbol{z}_k \mid \boldsymbol{V}_{l,g}) p(\boldsymbol{V}_{l,g} \mid \boldsymbol{x}_{k-1}^{(i)})}{q(\boldsymbol{V}_{l,g} \mid \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k)}$, here $i = l$. In fact, $\boldsymbol{V}_{l,g}$ can also be in the form of $\boldsymbol{V}_{l,g} = \boldsymbol{X}_{r1,g} + F \times (\boldsymbol{X}_{r2,g} - \boldsymbol{X}_{r3,g} + \boldsymbol{X}_{r4,g} - \boldsymbol{X}_{r5,g})$ or $\boldsymbol{V}_{l,g} = \boldsymbol{X}_{best,g} + F \times (\boldsymbol{X}_{r2,g} - \boldsymbol{X}_{r3,g})$ or $\boldsymbol{V}_{l,g} = \boldsymbol{X}_{best,g} + F \times (\boldsymbol{X}_{r2,g} - \boldsymbol{X}_{r3,g} + \boldsymbol{X}_{r4,g} - \boldsymbol{X}_{r5,g})$, and different forms of $\boldsymbol{V}_{l,g}$ have led to various variants of DE [6]. $\boldsymbol{X}_{best,g}$ is the target vector with the largest value of the objective function in the current population. The base vector index $r1$ and the difference vector indexes $r2$–$r5$ are randomly selected once per mutant. All the random indexes and the mutation vector's index l should be mutually exclusive. In what follows, the DEPFs using the above-mentioned different mutation forms of the DE algorithms are called DPF1, DPF2, DPF3 and DPF4 for short, respectively.

The next step is crossover, which mixes parameters of the $\boldsymbol{V}_{l,g}$ and the $\boldsymbol{X}_{l,g}$ to generate the trial vector $\{\boldsymbol{U}_{l,g} = (U_{j,l,g}), j = 1, \ldots, D\}$. The crossover is defined as:

$$\boldsymbol{U}_{l,g} = U_{j,l,g} = \begin{cases} V_{j,l,g} & \text{if}(\text{rand}_j[0,1) \le Cr \text{ or } j = j_{\text{rand}}) \\ X_{j,l,g} & \text{otherwise} \end{cases} \tag{4}$$

where $Cr \in [0, 1]$ is the crossover probability, $\text{rand}_j[0,1)$ is a uniform random number generator, and the random index j_{rand} is taken from the mutant [7]. Following the crossover is the selection step. In this step, $\boldsymbol{U}_{l,g}$ and $\boldsymbol{X}_{l,g}$ which has a higher value of objective function will survive into the next generation, that is

$$\boldsymbol{X}_{l,g+1} = \begin{cases} \boldsymbol{U}_{l,g} & \text{if } f(\boldsymbol{U}_{l,g}) \ge f(\boldsymbol{X}_{l,g}) \\ \boldsymbol{X}_{l,g} & \text{otherwise} \end{cases} \tag{5}$$

When the new population is propagated, the process of mutation, crossover and selection is repeated until the optimum is found or a prespecified maximum G of the number of generations is reached. The above steps describe the presented resampling schemes. A single iteration of the proposed particle filters based on DE resampling schemes is given as follows:

- Calculate the UKF proposal distribution $\mathcal{N}(\bar{\boldsymbol{x}}_k^{(i)}, \hat{\boldsymbol{P}}_k^{(i)}\}$ and sample $\{\hat{\boldsymbol{x}}_k^{(i)} : i = 1, \ldots, N\}$ from it, evaluate the weights $\{\boldsymbol{w}_k^{(i)} : i = 1, \ldots, N\}$ by (3) and normalise the weights.

• Exploit DE algorithms which are based on different mutations to obtain $\{\dot{x}_k^{(i)}, \dot{\tilde{w}}_k^{(i)} : i = 1, \ldots, N\}$ according to (4)–(5), where $\dot{\tilde{w}}_k^{(i)}$ is the weight of $\dot{x}_k^{(i)}$.
• Estimate the system state x_k by $x_k = \sum_{i=1}^{N} \dot{x}_k^{(i)} \dot{\tilde{w}}_k^{(i)}$.

Simulations: To compare the performance of the proposed filters to those of the SIR, the RPF and the UPF, we consider the following non-stationary problem:

$$x_k = 1 + sin(0.04\pi k) + 0.5x_{k-1} + u_{k-1} \tag{6}$$

$$z_k = \begin{cases} 0.2x_k^2 + v_k & 1 \le k \le 30 \\ 0.5x_k - 2 + v_k & 30 < k \le T \end{cases} \tag{7}$$

where u_{k-1} is a Gamma(3,2) random variable, $v_k \sim \mathcal{N}(0, 0.00001)$, and the total observation time is $T = 50$. The parameters of the UKF are the same as those in [4]. In the proposed filters, $F = 0.9$, $Cr = 0.6$, and the maximum number of generations is $G = 20$. First, 500 MC simulations and 10 particles are used. Fig. 1 shows the variation of estimation root mean squared error (RMSE) with observation time for all particle filters, where DSIR denotes the SIR based on the DE resampling scheme. It can be seen that the DSIR outperforms the SIR, and the RMSEs of the DEPFs are smaller than those of the SIR, the RPF and the UPF. In addition, the DPF3 has the best performance among the proposed DEPFs. Secondly, 200 MC tests and 200, 100, and 10 particles are used, respectively. The results of mean and variance of the 200 RMSE for 200 MC tests are shown in Table 1. It is shown that the DPF3 needs a smaller number of particles than that of the SIR, RPF and UPF to obtain accurate estimations of the state. With 10 particles, the DPF3 outperforms the SIR and the RPF using 200 particles. With 100 particles, the DPF3 outperforms the UPF using 200 particles. By using the DE schemes, the diversity of the particles is increased. As a result, the performance of estimation is improved.

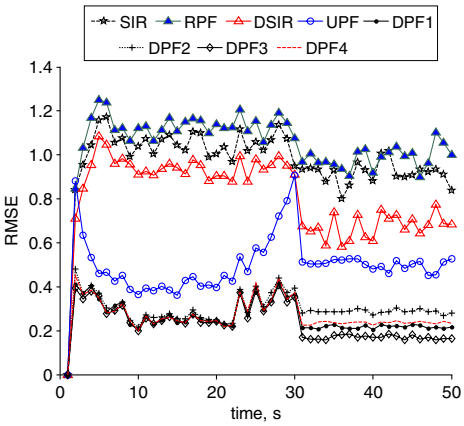

Fig. 1 *RMSE of SIR, RPF, UPF, DSIR and DEPFs for 500 MC simulations, where N = 10*

Table 1: Mean and variance of 200 RMSE of SIR, RPF, UPF, and DPF3 for 200 MC simulations

Algorithm	Mean	Variance	*N*
RPF	0.4222	0.0377	200
PF	0.4576	0.0631	
UPF	0.0830	0.0117	
DPF3	0.0573	0.0068	
RPF	0.5604	0.0375	100
PF	0.5680	0.0613	
UPF	0.0922	0.0086	
DPF3	0.0778	0.0051	
RPF	1.0467	0.0302	10
PF	0.9687	0.0334	
UPF	0.5024	0.0097	
DPF3	0.2332	0.0082	

Conclusion: Using the presented resampling schemes, the differential evolution particle filters are proposed. Simulation results demonstrate that the proposed filters have improved performance compared with that of the SIR, the RPF and the UPF.

Acknowledgment: This work was supported by the National Natural Science Foundation of China (no. 60901065).

© The Institution of Engineering and Technology 2011
13 June 2011
doi: 10.1049/el.2011.1825
One or more of the Figures in this Letter are available in colour online.

H.-W. Li, J. Wang and H.-T. Su (*National Lab of Radar Signal Processing, Xidian University, Xi'an 710071, People's Republic of China*)

E-mail: hwli@mail.xidian.edu.cn

References

1 Gordon, N., Salmond, D., and Smith, A.: 'Novel approach to nonlinear/non-Gaussian Bayesian state estimation', *IEE Proc. F*, 1993, **140**, (2), pp. 107–113
2 Arulampalam, M.S., Maskell, S., Gordon, N., and Clapp, T.: 'A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking', *IEEE Trans. Signal Process.*, 2002, **50**, (2), pp. 174–188
3 De Freitas, J.F.G., Niranjan, M., Gee, A.H., and Doucet, A.: 'Sequential Monte Carlo methods to train neural network models', *Neural Comput.*, 2000, **12**, (4), pp. 955–993
4 Merwe, R., Doucet, A., Freitas, N., and Wan, E.: 'The unscented particle filter', Technical report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000, pp. 1–40
5 Musso, C., Oudjane, N., and Legland, F.: 'Improving regularised particle filter', *in* Doucet, A., de Freitas, J.F.G., Gordon, N.J. (Eds): 'Sequential Monte Carlo methods' (Springer-Verlag, 2001), pp. 247–272
6 Chakraborty, U.K.: 'Advances in differential evolution' (Springer-Verlag, Berlin Heidelberg, 2008), pp. 1–27
7 Montgomery, J.: 'Crossover and the different faces of differential evolution searches'. IEEE Cong. Evolutionary Computation, Barcelona, Spain, 2010, pp. 1–8