

COMPUTATIONALLY EFFICIENT PARTICLE FILTERING USING ADAPTIVE TECHNIQUES

*Bhavana B Manjunath**, *Antonia Papandreou-Suppappola**, *Chaitali Chakrabarti**, *Darryl Morrell***

*Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287, USA

**Department of Engineering, Arizona State University, Mesa, AZ 85212, USA

ABSTRACT

We propose a computationally efficient particle filtering algorithm that adaptively chooses between the sequential importance resampling (SIR) particle filter and the unscented particle filter (UPF). The technique is based on the use of the Kullback-Leibler distance (KLD) sampling and the choice of either of the algorithms is governed by the error in estimation. The SIR particle filter is most often chosen among the variations of the particle filter because of the choice of the transitional prior as the importance density and easy evaluation of weights. However, it can be inefficient for highly non-linear dynamic systems. In contrast, the UPF which uses the scaled unscented transform performs better than the SIR but is computationally more expensive. The proposed algorithm couples the easy evaluation of the weights and the faster sampling capabilities of the SIR filter with the improved accuracy of the UPF. We apply the technique to a scalar estimation problem and demonstrate through simulations that the new algorithm is more accurate than the SIR particle filter and is faster than the UPF for systems characterized by highly non-linear measurement models.

1. INTRODUCTION

Most of the real-world tracking and estimation problems in applications such as sensing and signal processing, medical prognosis, communications, model diagnostics, navigation and neural network training involve elements of non-Gaussianity, non-linearity and non-stationarity. They can be solved with high accuracy with the use of particle filtering algorithm and its variations [1, 2].

These algorithms are based on Sequential Importance Sampling (SIS) and the key idea is to represent the posterior density function by a set of random samples ("particles") sampled from an importance sampling density and associated weights. Based on the number of particles used and the choice of the importance sampling density, these algorithms can be computationally intensive; they can differ in performance, efficiency, and computational cost. There has been an upsurge of interest to improve the accuracy as well as time efficiency

of particle filtering algorithms so that they can be applied to real-time systems. Different choices of importance density [2, 3, 4] and varying modification in the resampling stage [5, 6, 7] have yielded various enhancements to the generic particle filter.

Our work is based on the SIR particle filter and the UPF. The UPF performs better than the SIR for highly non-linear scenarios; however, this increase in estimation accuracy is traded-off with a higher computational complexity. Our objective is to propose a new particle filtering approach for highly non-linear dynamic scenarios that maintains the accuracy of the UPF but is closer in computational complexity to the SIR. Towards this goal, we first analyze the performance and computational complexity of the SIR and the UPF in Section 2. We then develop a new system for sequential Bayesian filtering in which the SIR particle filtering and the UPF algorithms are adaptively configured based on the Kullback-Leibler distance (KLD) sampling technique. The implementation is aimed at achieving a computationally efficient system with reduced estimation error. The detailed description of the technique is provided in Section 3. In order to demonstrate the improved performance of the new algorithm, we test it using a one-dimensional (1-D) scalar estimation problem and provide the simulation results in section 4.

2. BACKGROUND

2.1. SIR and UPF Algorithms

Consider a filtering problem that estimates the unknown state vector \mathbf{x}_k at discrete time instant k , based on a discrete time stochastic model given by,

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}) \quad (1)$$

where \mathbf{f}_{k-1} is a known, non-linear function and \mathbf{v}_{k-1} is the process vector that represents possible state modeling errors. The estimation is based on available information that includes a set of noisy measurements \mathbf{z}_k , at time k . The measurement is related to the state by the measurement equation given by,

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k) \quad (2)$$

Thanks to XYZ agency for funding.

where \mathbf{h}_k is a known, non-linear function and \mathbf{w}_k is the measurement noise vector. The filtering problem involves the estimation of the probability density function $p(\mathbf{x}_k|\mathbf{Z}_k)$ (where, $\mathbf{Z}_k = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$) which is achieved by the prediction and update stages [2, 1]. The method represents the posterior probability density function by a set of random samples (particles) with associated weights that can be used to estimate the state. Let $\mathbf{X}_k = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ represent all the states upto time k . Let the posterior density $p(\mathbf{X}_k|\mathbf{Z}_k)$ be represented by a set of N particles $\{\mathbf{X}_k^i, i = 1, \dots, N\}$ and associated weights $\{w_k^i, i = 1, \dots, N\}$. From [1], the posterior probability density function can be approximated as,

$$p(\mathbf{X}_k|\mathbf{Z}_k) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{X}_k - \mathbf{X}_k^i). \quad (3)$$

The posterior is expressed by

$$p(\mathbf{X}_k|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{X}_k, \mathbf{Z}_{k-1})p(\mathbf{X}_k, \mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \quad (4)$$

which can be approximated as [1]

$$p(\mathbf{X}_k|\mathbf{Z}_k) \propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}). \quad (5)$$

The weight update equation is then shown to be

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)}. \quad (6)$$

The transitional prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ is the proposal distribution in the SIR filter. The algorithm also incorporates the resampling step at every time index to eliminate the problem of degeneracy. With the transitional prior as the importance density, the importance weights are approximated as,

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)} \quad (7)$$

which reduces to

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i). \quad (8)$$

The weights are normalized such that $\frac{1}{N} \sum_i w_k^i = 1$.

The efficiency of the SIR filter suffers due to the choice of importance density as the prediction step is accomplished without incorporating the information of the latest available measurement \mathbf{z}_k .

The unscented particle filter uses the unscented Kalman filter (UKF) as the importance density which incorporates the most current measurement thereby propagating the particles to the likelihood function [8]. The unscented transform, a method for calculating the statistics of a random variable that undergoes a nonlinear transform is the basis for the unscented particle filter. Each particle is propagated by a separate unscented Kalman filter to generate updated mean and covariance, $\tilde{\mathbf{x}}_k^i$

and $\tilde{\mathbf{P}}_k^i$, respectively at every time instant k incorporating the measurement z_k . The importance density from which particles are sampled are then,

$$q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k) = \mathcal{N}(\mathbf{x}_k^i; \tilde{\mathbf{x}}_k^i, \tilde{\mathbf{P}}_k^i). \quad (9)$$

The unscented particle filter has been shown in [8] to perform more accurately than the SIR. But there is an increased computational cost due to the propagation of each particle using the UKF.

3. PROPOSED ALGORITHM

The proposed algorithm couples the easy evaluation of the weights and the faster sampling capabilities of the SIR filter with the improved accuracy of the UPF. This is achieved by using a computationally efficient strategy to adaptively choose between the two algorithms. This strategy is based on checking, at each time-step, whether the number of particles used by the SIR is sufficient to yield a desired accuracy. If the number of particles is not sufficient, then the UPF is used at that time-step instead of the SIR. In order to decide on the number of samples required for a desired accuracy, at every iteration, the KLD-sampling algorithm is applied [9]; this is a method used to adaptively estimate the number of particles needed to bound the error of the particle filter.

3.1. Use of KLD Sampling Algorithm

The KL-distance between the posterior and the empirical probability distributions has been incorporated as a decision factor to choose between the SIR filter and the UPF in the proposed system. Let N be the number of particles drawn from the proposal distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ at time k to approximate the posterior probability density function $p(\mathbf{x}_k|\mathbf{Z}_k)$. We start by sorting these particles in increasing order of magnitude. We assume a bin size value Δ , to obtain $M = \frac{\Delta \mathbf{x}_k}{\Delta}$ bins. Here, $\Delta \mathbf{x}_k$ corresponds to the difference in the magnitude between the largest and smallest particles (where the dimensions of the particles is taken into consideration). Thus, the bin size is an interval based on the particle magnitude values. As a different number of particles can fall within a bin, let that number be denoted by $M_l, l = 1, \dots, M$. The proposal distribution thus has $M - 1$ degrees of freedom [9]. Let the vector,

$$\mathbf{d}_k^l = [x_k^{lM_l+1}, \dots, x_k^{(l+1)M_l}] \quad (10)$$

denote the M_l th particle in the l th bin. We assume that $p_k^{m,l}$ be the probability that the m th particle in $\mathbf{d}_k^l, m = 1, \dots, M_l$ came from the l th bin. We also denote the overall probability vector in the l th bin as $\mathbf{P}_k^l = \left\{ p_k^{m,l} \right\}_{m=1}^{M_l}$. We use the maximum likelihood estimation technique to obtain an estimate of this overall probability as,

$$\hat{\mathbf{P}}_k^l = \frac{1}{N} \mathbf{d}_k^l. \quad (11)$$

Now, let $P(K(\widehat{\mathbf{P}}, \mathbf{P}) \leq \epsilon)$ denote the probability that the KL-distance (given by Equation (12)) between the approximated posterior probability density function and the new proposal distribution based on the particles \mathbf{d}_k^l is less than or equal to a pre-defined ϵ .

$$K(\widehat{\mathbf{P}}_k, \mathbf{P}_k) = \sum_{l=1}^M \hat{p}_k^l \log_{10} \left(\frac{\hat{p}_k^l}{p_k^l} \right) \quad (12)$$

The relationship between the desirable number of particles D_k and the probability $P(K(\widehat{\mathbf{P}}, \mathbf{P}) \leq \epsilon)$ is derived as [9]

$$P(K(\widehat{\mathbf{P}}, \mathbf{P}) \leq \epsilon) = P(\chi_{M-1,1}^2 \leq 2D_k\epsilon) \quad (13)$$

where $\chi_{h,g}^2$ represents the chi-square statistic with h degrees of freedom with probability g . Since we want $P(K(\widehat{\mathbf{P}}, \mathbf{P}) \leq \epsilon)$ with a probability $1 - \delta$, where δ is a pre-defined accuracy level, we obtain the following result from the chi-square distribution.

$$P(\chi_{M-1,1}^2 \leq \chi_{M-1,1-\delta}^2) = 1 - \delta. \quad (14)$$

From Equations (13) and (14), we obtain the desired value for D_k to be

$$D_k = \frac{1}{2\epsilon} \chi_{M-1,1-\delta}^2. \quad (15)$$

For ease of computation, this can be approximated by the Wilson-Hilferty transformation as,

$$D_k = \frac{M-1}{2\epsilon} \left\{ 1 - \frac{2}{9(M-1)} + \sqrt{\frac{2}{9(M-1)} z_{1-\delta}} \right\}^3 \quad (16)$$

where $z_{1-\delta}$ is the upper $(1 - \delta)$ quantile of the standard normal distribution.

3.2. Computationally Efficient Adaptive Particle Filtering

In the proposed adaptive particle filtering method, we initialize the state estimation using the SIR filter. Calculation of the desirable number of particles D_k is performed at every time step k as in Equation (16). For all iterations we fix the acceptable accuracy level ϵ and δ . We also fix Δ such that a small number of samples is chosen if the density is focused on a small subspace of the state, and a large number of samples is chosen if the samples have to spread over a major part of the state space. It should be noted that Δ determines the number of bins M which will vary every time step based on the performance. If the estimation performance is poor, M will be high due to the increased spread of the particles. If the estimation performance meets the requirement of the accuracy defined by δ , M will be low as the particles will not have high variance. We set, at the beginning of the state estimation, an upper bound D_u , a threshold on D_k . This is a design parameter which is set based on the available computational

resources. At time step k , the determining step of the new algorithm for the choice of SIR filter or the UPF at the next time step $k + 1$, is the relationship between D_k and D_u . If $D_k > D_u$, then we use the UPF until $D_k < D_u$ otherwise we continue using the SIR filter. From the expansion of Equation (16), it is seen that D_k is proportional to M . This shows that the estimation performance is reflected in the calculated D_k and validates the use of D_k as the determining step of the new algorithm for the choice of SIR filter or the UPF. Note that D_k is only used for the decision of the choice of the filter. We do not adapt the number of particles in the filter. Both the SIR and the UPF filters still use N particles we started with.

The proposed algorithm is described next in more detail:

1. At time $k = 0$ of the estimation problem, we have N particles, \mathbf{x}_0^i and their associated weights \mathbf{w}_0^i , $i = 1, \dots, N$. All the particles are initially weighted by $\frac{1}{N}$ where N is decided based on the application at hand.
2. We choose the design parameters: bin size Δ and D_u based on the resources at hand, and δ and ϵ based on the required accuracy.
3. For time $k = 1$, we initialize the estimation with the SIR filter to obtain \mathbf{x}_1^i .
4. We sample from $\mathbf{x}_k^i \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)$ and evaluate the importance weights as $w_k^i \propto w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i)$.
5. We sort particles \mathbf{x}_k^i in the increasing order of magnitude. Using the defined bin size value Δ , we obtain $M = \frac{\Delta \mathbf{x}_k}{\Delta}$ bins. Here, $\Delta \mathbf{x}_k$ corresponds to the difference in the magnitude between the largest and smallest particles (where the dimensions of the particles is taken into consideration).
6. Using M , we calculate D_k as,

$$D_k = \frac{M-1}{2\epsilon} \left\{ 1 - \frac{2}{9(M-1)} + \sqrt{\frac{2}{9(M-1)} z_{1-\delta}} \right\}^3 \quad (17)$$

where $z_{1-\delta}$ is the upper $(1 - \delta)$ quantile of the standard normal distribution.

7. If $D_k > D_u$, then we use the UPF and then continue as in Step 5 and 6 until $D_k < D_u$. When $D_k < D_u$ we use the SIR filter.

4. SIMULATION RESULTS

The simulation example considered here was the scalar estimation problem discussed in [8]. The process and observation models are given by,

$$x_{k+1} = 1 + \sin(\omega\pi k) + \phi_1 x_k + v_k \quad (18)$$

$$y_k = \begin{cases} \phi_2 x_k^2 + n_k, & k \leq 30 \\ \phi_3 x_k - 2 + n_k, & k > 30 \end{cases} \quad (19)$$

Algorithm	MSE mean	MSE variance	Computation time in s for 100 runs
SIR	0.0890	0.0901	380.6
UPF	0.0299	0.0183	2,359
CEAPF	0.0537	0.0290	1,389

Table 1. Comparison of MSE mean and variance and time for estimation.

where v_k is the process noise modeled by a Gamma random variable with shape 3 and scale 2 and n_k is zero mean additive white Gaussian noise with variance 0.00001. The scalar parameters chosen are $\omega = 4 \times 10^{-2}$, $\phi_1 = 0.5$, $\phi_2 = 0.2$, and $\phi_3 = 0.5$. The problem involves the estimation of the state sequence x given only the noisy measurements y . The estimation for 120 time steps was conducted using the SIR, UPF and the proposed CEAPF with 50 particles. For the CEAPF, the threshold on the number of particles was $N_u = 50$. Each algorithm iteration was averaged over 100 Monte-Carlo simulations on an Intel dual-core pentium-D 3GHz system with 2GB RAM. There was random re-initialization for each run. Figure 1 shows the overlaid performance plot for the estimation using the SIR, UPF and the CEAPF. Table 1 compares the results in terms of the mean and the variance on the overall error (averaged over all the time steps and over all Monte Carlo simulations) and the overall computational time. Our simulations show that the CEAPF resulted in the reduction of 40% in the MSE mean and 67% in the MSE variance when compared to the SIR particle filters. Furthermore, the CEAPF resulted in 41% reduction in computational time when compared to the UPF. Note that the CEAPF is still much slower than the SIR.

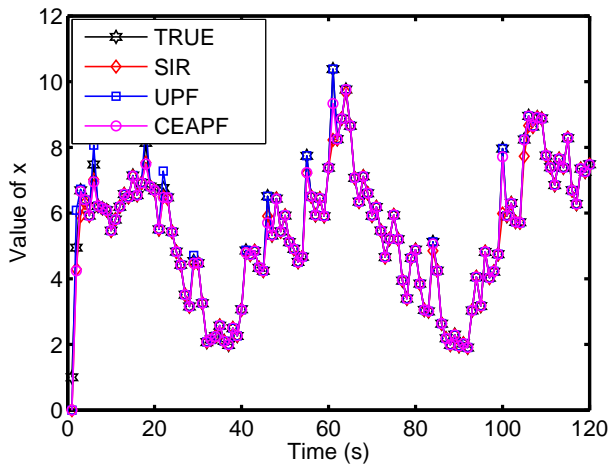


Fig. 1. Comparison of scalar estimation performance.

5. CONCLUSION

A computationally efficient particle filter with reduced estimation error for highly non-linear dynamic systems has been developed. The configuration was aimed at choosing between the SIR filter and the UPF in order to achieve a low implementation and computational overhead while still preserving adequate performance accuracy. We demonstrated through simulations that there was a 41% reduction in the computational time as compared to the UPF and a 40% reduction in the MSE as compared to SIR filter for highly non-linear scalar estimation problem.

6. REFERENCES

- [1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, Feb 2002.
- [2] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House Publishers, 2004.
- [3] M. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–602, 1999.
- [4] T. Schon, F. Gustafsson, and P.J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Trans. Signal Processing*, vol. 53, pp. 2279–2289, 2005.
- [5] M. Bolic, P.M. Djuric, and S. Hong, "New resampling algorithms for particle filters," Hong Kong, China, 2003, ICASSP, vol. 2, pp. 589–592.
- [6] M. Bolic, P.M. Djuric, and S. Hong, "Resampling algorithms for particle filters a computational complexity perspective," *EURASIP Journal on Applied Signal Processing*, vol. 15, pp. 2267–2277, 2004.
- [7] S. Hong, S.S. Chin, and P.M. Djuric, "Design and implementation of flexible resampling mechanism for high-speed parallel particle filters," *VLSI Signal Processing*, vol. 44, pp. 47–62, 2006.
- [8] R. van der Merwe, J.F.G. de Freitas, A. Doucet, and E. A. Wan, "The unscented particle filter," Tech. Rep. CUED/F-INFENG/TR380, Cambridge University Engineering Department, Aug. 2000.
- [9] D. Fox, "KLD-sampling: Adaptive particle filters," *Advances in Neural Information Processing Systems*, vol. 14, pp. 713–720, 2001.