Gaussian proposal density using moment matching in SMC methods

S. Saha · P.K. Mandal · Y. Boers · H. Driessen · A. Bagchi

Abstract In this article we introduce a new Gaussian proposal distribution to be used in conjunction with the sequential Monte Carlo (SMC) method for solving non-linear filtering problems. The proposal, in line with the recent trend, incorporates the current observation. The introduced proposal is characterized by the exact moments obtained from the dynamical system. This is in contrast with recent works where the moments are approximated either numerically or by linearizing the observation model. We show further that the newly introduced proposal performs better than other similar proposal functions which also incorporate both state and observations.

S. Saha (✉) · P.K. Mandal · A. Bagchi
Department of Applied Mathematics, University of Twente,
7500 AE, Enschede, The Netherlands
e-mail: s.saha@math.utwente.nl

P.K. Mandal
e-mail: p.k.mandal@math.utwente.nl

A. Bagchi
e-mail: a.bagchi@math.utwente.nl

Y. Boers · H. Driessen
THALES Nederland BV, Haaksbergerstraat 49, 7554 PA,
Hengelo, The Netherlands

Y. Boers
e-mail: yvo.boers@nl.thalesgroup.com

H. Driessen
e-mail: hans.driessen@nl.thalesgroup.com

1 Introduction

Consider a nonlinear dynamic system given by

$$x_k = f(x_{k-1}, w_k), \tag{1}$$

$$y_k = h(x_k, v_k), \quad k = 1, 2, \ldots \tag{2}$$

where (x_k) are the unobservable system values (the state) with (known) initial prior density $p(x_0) \equiv p(x_0|x_{-1})$ and (y_k) are the observed values (the measurements). The process noises (w_k) are assumed to be independent of the measurement noises (v_k). The problem is to estimate the unobserved system value x_n in some optimal manner from all the observations $y_{1:n} \equiv (y_1, y_2, \ldots, y_n)$, up to time n, or equivalently, estimate the conditional density (also known as filtered density) $p(x_n|y_{1:n})$. An analytical solution can be found only for a few special cases such as when both the system and observation equations (1)–(2) are linear and the noise processes are Gaussian (Kalman filter). For general models, analytical approximations such as Extended Kalman filter and Gaussian sum filter (Anderson and Moore 1979; Jazwinski 1970; Bagchi 1993) and other approximate methods using numerical integration (Kitagawa 1987), the unscented Kalman filter (Julier and Uhlmann 1997; Wan and van der Merwe 2000) and the Gaussian quadrature Kalman filter (Ito and Xiong 2000) are proposed in the literature.
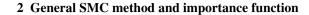
Simulation based sequential Monte Carlo (SMC) methods, also known as Particle Filters (PF's), provide the target filter density $p(x_n|y_{1:n})$ in the form of a cloud of particles (Handschin and Mayne 1969; Akashi and Kumamoto 1975; Gordon et al. 1993; West 1993; Pitt and Shephard 1999; Doucet et al. 2001; Arulampalam et al. 2002). The biggest advantage of the SMC method is that it can easily adapt to

nonlinearity in the model and/or non-Gaussian noises. The efficiency of the PF algorithm depends on the so-called importance function $\pi(\cdot)$, also often referred to as the proposal distribution, used to generate the particles. It has been shown in (Doucet et al. 2000) that the importance function of the form $\pi = p(x_k|x_{k-1}, y_k)$ is optimal in a certain sense. There are, however, two major, in practice prohibitive, drawbacks for using this importance function. Firstly, drawing samples according to $p(x_k|x_{k-1}, y_k)$ is, in general, difficult. Secondly, it is also difficult to get an analytical expression which is needed for the weight update.

In this article we propose as the importance function a Gaussian approximation of $p(x_k|x_{k-1}, y_k)$ by matching the exact moments up to second order of the distribution of (x_k, y_k) conditional on x_{k-1}. Recently, Doucet et al. (2000) and Guo et al. (2005) also proposed similar importance functions. However, Doucet et al. (2000) use the linearised approximation of the observation model to calculate the moments, while Guo et al. (2005) approximate the moments by different numerical methods such as the Gaussian-Hermite quadrature rule or the Julier-Uhlmann quadrature rule. Besides the possibility that the use of exact moments may lead to better approximation of the optimal proposal distribution, our method has one distinct advantage over the one in Guo et al. (2005), namely that, it is computationally less demanding. We also note that, the dynamical systems considered by Doucet et al. (2000) and Guo et al. (2005) are with additive Gaussian noise processes, whereas our method would work for more general models, so long as the observation model is a polynomial. For comparison we have also considered unscented particle filter (van der Merwe et al. 2000) which, too, works for general dynamical systems. Our experimental results with additive Gaussian noise processes show that the overall performance of our proposal function is better than that of the other proposals, considering the trade off between the RMSE and the computational load.

The rest of the article is organized as follows. In Sect. 2 the general SMC method is reviewed very briefly and the role of the importance function is discussed. We describe our proposed importance function using the exact moments in Sect. 3. Construction of other importance functions as proposed by Doucet et al. (2000), Guo et al. (2005) and van der Merwe et al. (2000) are briefly reviewed in Sect. 4. Implementation issue of our proposed method is discussed in Sect. 5. Section 6 contains the numerical comparison results of these methods based on two examples—one with polynomial (Sect. 6.1) and the other with non-polynomial (Sect. 6.2) observation equation. Finally, Sect. 7 concludes the article.

2 General SMC method and importance function

Suppose that the system dynamics are given by (1)–(2). The sequential Monte Carlo method is based on importance sampling and allows one to estimate recursively in time the distribution function $p(x_k|y_{1:k})$. The estimate is given in the form of a weighted particle cloud $\{(x_k^{(i)}, \tilde{w}_k^{(i)}), i = 1, \ldots, N\}$. Given the observations up to the current time k ($y_{1:k}$) and the particles up to time $k-1$ ($x_{0:k-1}^{(i)}$), $x_k^{(i)}$ is drawn according to a normalized importance function $\pi(x_k|x_{0:k-1}^{(i)}, y_{1:k})$ whose support includes that of the true posterior and subsequently the importance weights are updated. For a full account we refer the readers to Arulampalam et al. (2002).

Usually in practice, the importance function is taken to be the transition density, i.e., $\pi(x_k|x_{0:k-1}^{(i)}, y_{1:k}) = p(x_k|x_{k-1}^{(i)})$ because it is easily available from the model. It is known that this algorithm suffers from the degeneracy problem, that is to say, the variance of the importance weights can only increase over time. It has been shown by Doucet et al. (2000) that an importance function of the form $\pi(x_k|x_{0:k-1}^{(i)}, y_{1:k}) = p(x_k|x_{k-1}^{(i)}, y_k)$ addresses this issue by minimizing the variance of the (unnormalized) importance weight $w_k^{(i)}$ conditional upon $x_{0:k-1}^{(i)}$ and $y_{1:k}$.

In general, though, this choice of the importance function is not practical as it is difficult to generate samples from this distribution. Furthermore, one needs an analytical expression of the importance function to be used in the weight update equation, which is also generally difficult with this choice of the importance function.

3 Importance function based on exact moment matching (EMM)

Suppose the system dynamics are given by (1)–(2). We further assume the following.

Assumption A *All the moments of (x_k, y_k) conditional on x_{k-1} up to second order, i.e., $E(x_k|x_{k-1})$, $E(x_k x_k^T|x_{k-1})$, $E(y_k|x_{k-1})$, $E(y_k y_k^T|x_{k-1})$, and $E(x_k y_k^T|x_{k-1})$ are known.*

To determine the importance function, to be used in conjunction with the particle filtering algorithm, we proceed as follows. We approximate the joint distribution of (x_k, y_k), conditional on x_{k-1}, by a Gaussian distribution with matching moments up to the second order. Let the corresponding mean $\mu^{(k)}$ and the covariance $\Sigma^{(k)}$ be given by

$$\mu^{(k)} = \begin{pmatrix} \mu_1^{(k)} \\ \mu_2^{(k)} \end{pmatrix} \quad \text{and} \quad \Sigma^{(k)} = \begin{pmatrix} \Sigma_{11}^{(k)} & \Sigma_{12}^{(k)} \\ (\Sigma_{12}^{(k)})^T & \Sigma_{22}^{(k)} \end{pmatrix}. \quad (3)$$

Note that $\mu^{(k)}$ and $\Sigma^{(k)}$ can be calculated from the moments which are assumed to be known from Assumption A. Subsequently, we take the importance function to be the conditional distribution of x_k, given (x_{k-1}, y_k), derived from the approximated Gaussian distribution above. Thus, we assume $\pi(x_k|x_{0:k-1}^{(i)}, y_{0:k}) \sim \mathcal{N}(m_k, \Sigma_k)$, where

$$m_k = \mu_1^{(k)} + \Sigma_{12}^{(k)}\big[\Sigma_{22}^{(k)}\big]^{-1}\big(y_k - \mu_2^{(k)}\big), \tag{4}$$

$$\Sigma_k = \Sigma_{11}^{(k)} - \Sigma_{12}^{(k)}\big[\Sigma_{22}^{(k)}\big]^{-1}\big(\Sigma_{12}^{(k)}\big)^T. \tag{5}$$

We note here that a sufficient condition for Assumption A to hold is:

Condition 1 *Both $E(y_k|x_k)$ and $E(y_k y_k^T|x_k)$ are polynomials in x_k of degree at most m and all conditional moments of x_k, given x_{k-1}, are known up to order m.*

In the special case when the noise processes in (1)–(2) are additive Gaussian, suppose the system dynamics are given by

$$x_k = f(x_{k-1}) + w_k, \quad w_k \sim \mathcal{N}(0, Q), \tag{6}$$

$$y_k = h(x_k) + v_k, \quad v_k \sim \mathcal{N}(0, R), \quad k = 1, 2, \dots \tag{7}$$

then the conditional moments of all order of x_k given x_{k-1} are known. If, in addition, $h(\cdot)$ in (7) is a polynomial, then Condition 1 would be satisfied and hence Assumption A would hold. The precise formulas for the quantities can be found in Appendix A.

4 Other Gaussian importance functions

There are other Gaussian importance functions proposed in the literature. We mention three of them here. The first two are based on the Gaussian approximation of the optimal importance function $p(x_k|x_{k-1}, y_k)$ which are ideologically similar to EMM, but the moments are approximated in different ways. The third one, on the other hand, uses a bank of unscented Kalman filters to obtain the proposal density.

4.1 Importance function by linearization (LIN)

In an earlier paper Doucet et al. (2000) consider a dynamical system with additive Gaussian noise, given by (6)–(7). Observing that the optimal importance function $p(x_k|x_{k-1}, y_k)$ is Gaussian when $h(\cdot)$ in the observation model (7) is linear, the authors linearize the observation equation (7) to obtain

$$y_k \approx h(f(x_{k-1})) + C_k(x_k - f(x_{k-1})) + v_k \tag{8}$$

where $C_k = \frac{\partial h}{\partial x_k}(f(x_{k-1}))$. Subsequently, they use the corresponding Gaussian distribution as importance function. This essentially reduces to approximating the conditional distribution of (x_k, y_k), given x_{k-1}, by the Gaussian distribution with mean vector μ^* and covariance matrix Σ^* given by

$$\mu^* = \begin{pmatrix} f(x_{k-1}) \\ h(f(x_{k-1})) \end{pmatrix} \quad \text{and}$$

$$\Sigma^* = \begin{pmatrix} Q & QC_k^T \\ C_k Q & C_k Q C_k^T + R \end{pmatrix}. \tag{9}$$

4.2 Numerically approximated moment matching

In a more recent article, Guo et al. (2005) consider the same dynamical system given by (6)–(7). The importance function proposed by them is also in effect derived from a Gaussian approximation of the joint distribution of (x_k, y_k) conditional on x_{k-1}. Guo et al. (2005), however, approximate the moments in (3) by various numerical techniques, such as the Gauss-Hermite quadrature (GHQ) rule and the Julier-Uhlmann quadrature (JUQ) rule. We refer the reader to the original article for the details.

4.3 Unscented particle filter (UPF)

Unscented particle filter algorithm of van der Merwe et al. (2000) is suitable for general dynamical systems given by (1)–(2). In this method, for each particle a separate unscented Kalman filter is propagated to generate the Gaussian proposal distribution. Once again we refer to the original article for details.

5 Implementation of the EMM

Clearly, the EMM as described in Sect. 3 can be implemented if the Assumption A holds. A proper classification of models for which Assumption A holds is not very easy. However, as mentioned in Sect. 3, if the dynamical system is given by (6)–(7) with $h(\cdot)$ in (7) a polynomial function, then EMM can be implemented.

When the exact values of the quantities in (3) cannot be calculated, we propose to approximate the observation equation by one of polynomial form and implement the EMM to derive the importance function. For instance, consider a real-valued dynamical system given by (6)–(7). We assume further that the function $h(\cdot)$ is n times differentiable. We approximate $h(\cdot)$ locally by its n-th degree Taylor polynomial around $x_k^* = f(x_{k-1})$ to get the following observation equation.

$$y_k = \sum_{m=0}^{n} a_m(x_k - x_k^*)^m + v_k \quad \text{with}$$

$$a_m = \frac{1}{m!}\left(\frac{\partial^m h(x_k)}{\partial x_k^m}\right)_{x_k = x_k^*}. \tag{10}$$

Then the quantities in (3) can be approximated by the corresponding quantities for the dynamical system governed by (6) and (10). See Appendix B for details.

Note that the approach proposed above extends the methodology used by Doucet et al. (2000) where the observation equation is approximated by the first degree Taylor polynomial, whereas we consider higher degree polynomials. It is also worthwhile to note the difference between the approach followed by Guo et al. (2005) and the one proposed above. Guo et al. (2005) work with the given nonlinear model and during setting up of the Gaussian importance density, they approximate the moments. We, on the other hand, first approximate the observation equation with a n-th degree polynomial and further derive the Gaussian importance density using the exact moments (based on the approximated polynomial model).

In the following section we present two illustrative numerical examples. We notice that EMM performs better than the other methods in the sense that its computational load is considerably less, while the qualities of the filters (in terms of RMSE) are comparable.

6 Numerical simulation results

In this section we consider two examples—one with a polynomial observation model and the other with a non-polynomial model—and compare the filtered estimates obtained by different methods. In both examples we consider additive Gaussian noise processes.

6.1 Polynomial observation model

As in Doucet et al. (2000) we consider the system dynamics to be given by (6)–(7) with

$$f(x_{k-1}) = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k), \qquad (11)$$

$$h(x_k) = \frac{x_k^2}{20}. \qquad (12)$$

In our simulations, we set $Q = 10$, $R = 1$ and generate a time series data of length 100 starting with $x_0 \sim \mathcal{N}(0, 1)$. Given only the noisy observations y_k, particle filter algorithm is performed with the importance functions described in Sect. 4 (LIN, GHQ, JUQ, UPF) and the new one (EMM) proposed in Sect. 3. We estimate the state sequence x_k, $k = 1, 2, \ldots, 100$, with all the different methods mentioned above.

Note that, in this case, the differences in the moments used in EMM and LIN can be clearly seen. The moments

used in EMM, as given in (3), are

$$\mu^{(k)} = \begin{pmatrix} f(x_{k-1}) \\ \frac{f^2(x_{k-1})}{20} + \frac{Q}{20} \end{pmatrix} \quad \text{and}$$

$$\Sigma^{(k)} = \begin{pmatrix} Q & \frac{f(x_{k-1})Q}{10} \\ \frac{f(x_{k-1})Q}{10} & \frac{f^2(x_{k-1})Q}{100} + \frac{Q^2}{200} + R \end{pmatrix},$$

while the moments used in LIN, as given in (9), are

$$\mu^* = \begin{pmatrix} f(x_{k-1}) \\ \frac{f^2(x_{k-1})}{20} \end{pmatrix} \quad \text{and}$$

$$\Sigma^* = \begin{pmatrix} Q & \frac{f(x_{k-1})Q}{10} \\ \frac{f(x_{k-1})Q}{10} & \frac{f^2(x_{k-1})Q}{100} + R \end{pmatrix}.$$

For GHQ, we use the five point quadrature rule and for JUQ, the three ($n = 1$) sigma points were calculated using $\kappa = 2$. For UPF, the parameters are taken to be the same as that of van der Merwe et al. (2000) with $\alpha = 1, \beta = 0, \kappa = 2$ and $P_0 = 1$. However, while resampling we use systematic resampling scheme whereas van der Merwe et al. (2000) use residual resampling scheme.

For all methods, the initial distribution $p(x_0)$ is taken to be $\mathcal{N}(0, 5)$ and resampling was done when the effective sample size became less than one-third of the original sample size N. For each method, we first calculate the root mean squared error (RMSE) over $M = 100$ runs for each time point k and then the average (over time) RMSE, given by $\frac{1}{100}\sum_{k=1}^{100}(\frac{1}{M}\sum_{j=1}^{M}(\hat{x}_k^j - x_k^j)^2)^{\frac{1}{2}}$. Here x_k^j is the true (simulated) state for time k in the j-th run and $\hat{x}_k^j$ is the corresponding (point) estimate using a PF method.

Each of these methods is implemented with different Monte Carlo sample sizes $N = 100, 250, 500$ and 1000. In Table 1 the average RMSE's are presented. Also reported are the average (over the 100 runs) CPU time, in seconds, to complete a run and the average number of resampling steps (NRS) out of the 100 time steps.

First of all, we see from the table that, as expected, the performances (as measured by RMSE) of all the methods become similar as sample size N increases. This is in conformity with the fact that for any proposal distribution the particle filter converges to the true posterior as $N \to \infty$. The UPF seems to have a considerably higher computational load than the other methods. This can be explained by the fact that one needs to run the unscented Kalman filter for each particle (at each step) to calculate the proposal. Performances of GHQ, JUQ and EMM are more or less similar (which is better than LIN), but the time taken to arrive at the estimate is less in EMM than that by GHQ and JUQ. It appears that the numbers of resampling steps are almost the same for GHQ, JUQ and EMM, which is slightly better than LIN. So, the extra computational load for GHQ and

Table 1 Comparison of the performance of different proposal distributions with a polynomial observation model

N		LIN	GHQ	JUQ	UPF	EMM
	RMSE	4.7356	4.5724	4.5647	4.9552	4.6179
100	CPU	0.0255	0.0677	0.0617	1.9730	0.0403
	NRS	36.64	31.43	31.35	46.48	31.39
	RMSE	4.6715	4.4628	4.4989	4.5430	4.4838
250	CPU	0.0394	0.1855	0.1684	5.0514	0.0770
	NRS	36.26	32.05	32.29	46.38	32.18
	RMSE	4.4923	4.4299	4.4567	4.4766	4.4406
500	CPU	0.0650	0.5283	0.5008	10.0609	0.1423
	NRS	38.23	32.55	32.43	46.49	32.67
	RMSE	4.4310	4.4211	4.4122	4.4491	4.4162
1000	CPU	0.1352	1.6559	1.6128	20.1678	0.2892
	NRS	39.42	33.29	32.92	46.33	33.23

Table 2 Comparison of the performance of different proposal distributions with a nonpolynomial observation model

N		LIN	GHQ	JUQ	UPF	EMM
	RMSE	4.1643	4.0875	4.0977	4.1051	4.0936
100	CPU	0.0227	0.0686	0.0711	2.0225	0.0567
	NRS	18.13	11.88	12.31	28.76	17.13
	RMSE	4.1114	4.0592	4.0636	4.0630	4.0669
250	CPU	0.0359	0.1977	0.1963	5.1927	0.0752
	NRS	20.79	12.04	12.51	28.70	18.10
	RMSE	4.1027	4.0416	4.0509	4.0563	4.0524
500	CPU	0.0591	0.5930	0.5941	10.3734	0.1102
	NRS	23.40	12.29	12.58	28.61	18.78
	RMSE	4.0745	4.0415	4.0426	4.0483	4.0423
1000	CPU	0.1275	1.6741	1.6936	20.9481	0.2188
	NRS	25.45	12.35	12.64	28.82	19.69

JUQ relative to our EMM method can be construed as a result of computing the moments numerically. Thus one can conclude that the EMM method is more efficient compared to the other methods as it is computationally less demanding in arriving at the comparable level of efficiency.

6.2 Non-polynomial observation model

Let us consider the following model

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

$$w_k \sim \mathcal{N}(0, 10), \tag{13}$$

$$y_k = \tan^{-1}(x_k) + v_k, \quad v_k \sim \mathcal{N}(0, 1),$$

$$k = 1, 2, \dots. \tag{14}$$

Once again, a time series data of length 100 was simulated starting with $x_0 \sim \mathcal{N}(0, 1)$ and the different particle filters were applied on the observation y_k. Here the exact moments given in (3) are unknown. For EMM we have considered a 2nd degree Taylor polynomial, as described in Sect. 5. The other setup are as in the previous example in Sect. 6.1. The performances of different methods are presented in Table 2.

Again, comparing the RMSE's we observe that the performances of GHQ, JUQ, UPF and EMM are fairly similar, and they are all better than LIN. But when CPU times are compared, UPF is the worst performer. A very close look reveals that GHQ and JUQ may produce slightly lower RMSE compared to EMM. However, this relative gain is achieved at the expense of high computational load. Thus, considering the trade of between the RMSE and the computational cost, EMM appears to provide a practical and efficient proposal density.

7 Conclusion

In this article a new importance function has been proposed which is based on the Gaussian approximation of the conditional distribution of (x_k, y_k), given x_{k-1}, with the first two moments matched exactly to those of the true conditional distribution. To use the proposed method one needs to know the moments of the system dynamics up to the second order. A specific case in which this is satisfied is when the noise processes are additive Gaussian and the observation equation is polynomial. When the exact moments are not known but the noise processes are additive Gaussian and the observation model is smooth, we use a polynomial approximation of the observation model to derive the importance function. With the help of two examples it has been shown that the proposed EMM method provides a more practical and efficient proposal density considering the trade off between the performance (RMSE) and the computational load.

Appendix A

Consider the system dynamics as given in (6)–(7). Then, the conditional moments of x_k given x_{k-1} can be derived as follows.

$$E(x_k^m | x_{k-1}) = \sum_{r=0}^{m} \binom{m}{r} [f(x_{k-1})]^r E(w_k^{m-r})$$

$$= \sum_{r=0}^{m} \binom{m}{r} [f(x_{k-1})]^r \mu_{m-r}, \tag{15}$$

where μ_j is the j-th (raw) moment of the $\mathcal{N}(0, Q)$ distribution, given by

$$\mu_{2k+1} = 0 \quad \text{and} \quad \mu_{2k} = \frac{(2k)!}{2^k\, k!}\, Q^k,$$
$$\text{for } k = 0, 1, 2, \ldots. \tag{16}$$

When $h(x)$ is a polynomial of the form, $h(x) = \sum_{r=0}^{n} a_r\, x^r$, both $E(y_k|x_k)$ and $E(y_k\, y_k^T|x_k)$ are polynomials in x_k, thus satisfying Condition 1. Subsequently, we have

$$E(y_k|x_{k-1}) = \sum_{m=0}^{n} a_m E(x_k^m|x_{k-1})$$
$$= \sum_{m=0}^{n} \sum_{r=0}^{m} a_m \binom{m}{r} [f(x_{k-1})]^r \mu_{m-r}, \tag{17}$$

$$E(x_k\, y_k^T|x_{k-1}) = \sum_{m=0}^{n} a_m E(x_k^{m+1}|x_{k-1})$$
$$= \sum_{m=0}^{n} \sum_{r=0}^{m+1} a_m \binom{m+1}{r}$$
$$\times [f(x_{k-1})]^r \mu_{m+1-r}, \tag{18}$$

$$E(y_k\, y_k^T|x_{k-1}) = \sum_{m=0}^{n} \sum_{l=0}^{n} a_m a_l E(x_k^{m+l}|x_{k-1})$$
$$= \sum_{m=0}^{n} \sum_{l=0}^{n} \sum_{r=0}^{m+l} a_m a_l \binom{m+l}{r}$$
$$\times [f(x_{k-1})]^r \mu_{m+l-r}. \tag{19}$$

Then (15)–(19) ensure the validity of Assumption A.

Appendix B

Consider the dynamical system described by (6) and (10). Observing that the conditional distribution of $(x_k - x_k^*)$ given x_{k-1} is $N(0, Q)$, the following moments can be calculated as

$$E(y_k|x_{k-1}) = \sum_{m=0}^{n} a_m E[(x_k - x_k^*)^m|x_{k-1}]$$
$$= \sum_{m=0}^{n} a_m\, \mu_m, \tag{20}$$

$$E(x_k\, y_k^T|x_{k-1}) = \sum_{m=0}^{n} a_m E[x_k(x_k - x_k^*)^m|x_{k-1}]$$
$$= \sum_{m=0}^{n} a_m E[(x_k - x_k^*)^{m+1}|x_{k-1}]$$

$$+ x_k^* \sum_{m=0}^{n} a_m E[(x_k - x_k^*)^m|x_{k-1}]$$
$$= \sum_{m=0}^{n} a_m\, \mu_{m+1} + x_k^* \sum_{m=0}^{n} a_m \mu_m, \tag{21}$$

$$E(y_k\, y_k^T|x_{k-1}) = \sum_{m=0}^{n} \sum_{l=0}^{n} a_m a_l E[(x_k - x_k^*)^{m+l}|x_{k-1}]$$
$$= \sum_{m=0}^{n} \sum_{l=0}^{n} a_m a_l \mu_{m+l}, \tag{22}$$

where μ_j's are given by (16).

References

Akashi, H., Kumamoto, H.: Construction of discrete-time nonlinear filter by Monte Carlo methods with variance-reducing techniques. Syst. Control **19**(4), 211–221 (1975). (In Japanese)

Anderson, B., Moore, J.: Optimal Filtering. Prentice-Hall, Englewood Cliffs (1979)

Arulampalam, S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. **50**(2), 174–188 (2002)

Bagchi, A.: Optimal Control of Stochastic Systems. Series in Systems and Control Engineering. Prentice-Hall International, Englewood Cliffs (1993)

Doucet, A., de Freitas, J., Gordon, N. (eds.): Sequential Monte Carlo Methods in Practice. Springer, New York (2001)

Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering. Stat. Comput. **10**, 197–208 (2000)

Gordon, N., Salmond, D.J., Smith, A.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proc. F, Radar Signal Process. **140**(2), 107–113 (1993)

Guo, D., Wang, X., Chen, R.: New sequential Monte Carlo methods for nonlinear dynamic systems. Stat. Comput. **15**(2), 135–147 (2005)

Handschin, J.E., Mayne, D.Q.: Monte Carlo techniques to estimate the conditional expectation in multistage nonlinear filtering. Int. J. Control **9**(5), 547–559 (1969)

Ito, K., Xiong, K.: Gaussian filters for nonlinear filtering problems. IEEE Trans. Automat. Contr. **45**(5), 910–927 (2000)

Jazwinski, A.: Stochastic Processes and Filtering Theory. Academic Press, New York (1970)

Julier, S., Uhlmann, J.: A new extensions of the Kalman filter to nonlinear systems. In: Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management, vol. II, pp. 182–193. SPIE, Bellingham (1997)

Kitagawa, G.: Non-Gaussian state-space modeling of nonstationary time series. J. Am. Stat. Assoc. **82**(400), 1032–1063 (1987)

Pitt, M.K., Shephard, N.: Filtering via simulation: Auxiliary particle filters. J. Am. Stat. Assoc. **94**(446), 590–599 (1999)

van der Merwe, R., de Freitas, N., Doucet, A., Wan, E.: The unscented particle filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, Cambridge (2000)

Wan, E., van der Merwe, R.: The unscented Kalman filter for nonlinear estimation. In: Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications and Control Symposium (AS-SPCC), Lake Louise, Alberta, Canada. IEEE Press, New York (2000)

West, M.: Mixture models, Monte Carlo, Bayesian updating and dynamic models. Comput. Sci. Stat. **24**, 325–333 (1993)