Ensemble unscented Kalman filter for state inference in continuous–discrete systems

Bin Liu

School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, People's Republic of China
E-mail: bins@ieee.org

Published in *The Journal of Engineering*; Received on 13th March 2014; Accepted on 29th April 2014

Abstract: The authors consider non-linear state filtering problem in continuous–discrete systems, where the system dynamics is modelled by a stochastic differential equation, and noisy measurements of the system are obtained at discrete time instances. A novel particle method is proposed based on sequential importance sampling. This approach uses a bank of the continuous–discrete unscented Kalman filters (CDUKFs) to obtain the importance proposal distribution, retaining the advantage of the CDUKF in continuous–discrete systems as well as the accuracy of particle filter in highly non-linear systems. Simulation results show that the algorithm outperforms some other benchmarks substantially in estimation accuracy.

1 Introduction

Continuous–discrete dynamical models are used to model continuous-time systems, where a continuous-time signal is observed at discrete time instances. Such models appear commonly in engineering and physics applications, especially in the fields of control, communication and navigation [1–5]. Several methods have been proposed for dealing with continuous–discrete optimal filtering problems. For example, the Kalman–Bucy filter is discussed in [6–8] and the continuous-time unscented Kalman filter (CTUKF) is presented in [9]. Such Kalman-type filtering methods work optimally for linear Gaussian models, but lead to seriously biased estimate if the models are non-Gaussian and/or non-linear.

In this paper, an ensemble continuous–discrete unscented Kalman filtering (EnCDUKF) method is developed in the framework of sequential importance sampling (SIS), which in theory works optimally for any non-Gaussian and/or non-linear models, provided that the likelihood function is computable. Simulation results show a significant performance improvement over other benchmarks, such as the continuous–discrete variant of the bootstrap particle filter [10] and the CDUKF method [9].

2 Continuous–discrete filtering problem

A general form of continuous–discrete systems can be modelled as follows

$$\begin{aligned} \mathrm{d}\boldsymbol{x}(t) &= f(\boldsymbol{x}(t),\ t)\mathrm{d}t + \boldsymbol{L}(t)\,\mathrm{d}\boldsymbol{\beta}(t) \\ \boldsymbol{y}_k &= h_d(\boldsymbol{x}(t_k),\ t_k) + \boldsymbol{r}_k \end{aligned} \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ denotes the state at time t, $f(\cdot,\ \cdot)$ is the drift function, $\boldsymbol{L}(t)$ represents a dispersion matrix, $\boldsymbol{\beta}(t)$ refers to a Brownian motion with diffusion matrix $\boldsymbol{Q}_c(t)$, k represents discrete time instance of measurements, $\boldsymbol{y}_k \in \mathbb{R}^m$ is the kth discrete-time measurement, t_k is the arrival time of $\boldsymbol{y}_k$, h_d is the measurement function, $\boldsymbol{r}_k$ is a zero-mean Gaussian measurement noise with covariance matrix $\boldsymbol{R}_k$. The purpose of (Bayesian) continuous–discrete filtering is to recursively compute the posterior distribution $p[\boldsymbol{x}(t_k)/\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k]$. Theoretically, the solution to the continuous–discrete filtering problem can be computed by the following two steps, namely prediction and update [6]:

- *Prediction step:* calculates the predicted probability density at time step t_k from the 'Kolmogorov forward partial differential equation' using the old posterior probability density at time step t_{k-1} as the boundary condition.
- *Update step:* uses the Bayes' rule for computing the posterior probability density of state at time step t_k from the predicted probability density of the prediction step and the likelihood of the measurement $\boldsymbol{y}_k$.

3 Proposed method

The proposed EnCDUKF method uses a bank of CDUKFs for generating proposal distribution within an SIS framework. As shown in [9], the CDUKF is able to accurately propagate the mean and covariance of the Gaussian approximation to the state distribution in continuous–discrete systems, thus distributions generated by the CDUKF generally have a good support overlap with the true posterior distribution. It is desired for SIS that the proposal distribution mimic the true posterior, so it predicts that EnCDUKF method should have a good performance in accuracy of filtering. A brief algorithmic flow of the EnCDUKF is as shown in Fig. 1.

3.1 CDUKF update

Now we present the details necessary for implementation of the CDUKF in the SIS framework described above; the full derivation of CDUKF refers to [9].

The key idea of unscented Kalman filter (UKF) is to use a set of elaborately selected sigma points for unscented transform (UT) to obtain the state inference for non-linear systems [11–14]. Supposing that the dimension of the unknown state $\boldsymbol{x}$ is n, the UT usually needs $2n+1$ sigma points and each sigma point is associated with a specific weight. The weights are calculated as below

$$W_0 = \lambda/(n+\lambda); \quad W_0^{(c)} = \lambda/(n+\lambda) + (1-\alpha^2+\beta) \tag{3}$$

$$W_i = 1/\{2(n+\lambda)\}, \quad W_i^{(c)} = 1/\{2(n+\lambda)\}, \quad i = 1,\ \ldots,\ 2n \tag{4}$$

where $\lambda = \alpha^2(n+\kappa) - n$ and the positive constants α, β and κ are used as parameters of UT [11, 15].

For each particle $\hat{\boldsymbol{m}}_{k-1}^i$, the CDUKF process includes the followings two steps: prediction and update.

1. *Initialisation*: $k = 0$

   - for $i = 1, \ldots, N$, draw a set of state samples $\hat{\boldsymbol{m}}_0^i$ from the prior distribution $p(\boldsymbol{x}(t_0))$, then calculate the covariance matrix as below

$$\hat{P}_0^i = \mathrm{E}\left[\left(\hat{\boldsymbol{m}}_0^i - \bar{\boldsymbol{m}}_0\right)\left(\hat{\boldsymbol{m}}_0^i - \bar{\boldsymbol{m}}_0\right)^T\right] \tag{2}$$

   where $\bar{\boldsymbol{m}}_0 = \mathrm{E}[\boldsymbol{x}(t_0)]$ and $[\cdot]^T$ denotes the operation of matrix transposition.

2. For $k = 1, 2, \ldots$

   a. Importance sampling step

      – For $i = 1, \ldots, N$

         * Update the samples with the CDUKF
         * Sample $\hat{\boldsymbol{m}}_k^i$ from the Gaussian distribution generated by the CDUKF
         * Set $\hat{\boldsymbol{m}}_{0:k}^i = \begin{bmatrix}\hat{\boldsymbol{m}}_{0:k-1}^i & \hat{\boldsymbol{m}}_k^i\end{bmatrix}$ and $\hat{\boldsymbol{P}}_{0:k}^i = \begin{bmatrix}\hat{\boldsymbol{P}}_{0:k-1}^i & \hat{\boldsymbol{P}}_k^i\end{bmatrix}$, respectively.

      – For $i = 1, \ldots, N$, evaluate the importance weight up to a normalising constant.

      – For $i = 1, \ldots, N$, normalise the importance weights.

   b. Resampling step.

   c. Output: The output is generated in the same manner as for the generic particle filter.

Fig. 1 *Brief algorithmic flow of the EnCDUKF*

Prediction: Integrate the following differential equations

$$\frac{\mathrm{d}\boldsymbol{m}(t)}{\mathrm{d}t} = f(\boldsymbol{X}(t),\ t)\omega \tag{5}$$

$$\frac{\mathrm{d}\boldsymbol{P}(t)}{\mathrm{d}t} = \boldsymbol{X}(t)\boldsymbol{W}f^{\mathrm{T}}(\boldsymbol{X}(t),\ t) + f(\boldsymbol{X}(t),\ t)\boldsymbol{W}\boldsymbol{X}^{\mathrm{T}}(t) \tag{6}$$

$$+ \boldsymbol{L}(t)\boldsymbol{Q}_c(t)\boldsymbol{L}^{\mathrm{T}}(t) \tag{7}$$

from the initial conditions $\boldsymbol{m}(t_{k-1}) = \hat{\boldsymbol{m}}_{k-1}^i$, $\boldsymbol{P}(t_{k-1}) = \hat{\boldsymbol{P}}_{k-1}^i$ to time instance t_k, resulting in the predicted mean and covariance, denoted by $\hat{\boldsymbol{m}}_k^{i-}$ and $\hat{\boldsymbol{P}}_k^{i-}$, respectively. Here $\boldsymbol{X}(t) \triangleq [\,\boldsymbol{m}(t) \quad \cdots \quad \boldsymbol{m}(t)] + \sqrt{c}\begin{bmatrix}0 & \sqrt{\boldsymbol{P}(t)} & -\sqrt{\boldsymbol{P}(t)}\end{bmatrix}$ and

$$\boldsymbol{W} = \left(\boldsymbol{I} - [\,\omega \quad \cdots \quad \omega]\right) \times \mathrm{diag}\begin{pmatrix} W_0^{(c)} & \cdots & W_{2n}^{(c)}\end{pmatrix} \tag{8}$$

$$\times \left(\boldsymbol{I} - [\,\omega \quad \cdots \quad \omega]\right)^{\mathrm{T}} \tag{9}$$

where $\omega = [\,W_0 \quad \cdots \quad W_{2n}\,]^{\mathrm{T}}$ is the matrix of the sigma points' weights, $c = \alpha^2(n + \kappa)$, $\sqrt{\boldsymbol{A}}$ is the square root of matrix $\boldsymbol{A}$ satisfying $\sqrt{\boldsymbol{A}}\sqrt{\boldsymbol{A}}^{\mathrm{T}} = \boldsymbol{A}$, $\boldsymbol{I}$ is an identity matrix with appropriate dimension and diag means diagonalisation.

Update: First compute the predicted mean and the covariance of the measurement, denoted by μ_k^i and $\boldsymbol{S}_k^i$ respectively, and the cross-covariance of the state and measurement $\boldsymbol{C}_k^i$, as follows

$$\begin{aligned}
\boldsymbol{X}_k^{i-} &= \begin{bmatrix}\boldsymbol{m}_k^{i-} & \cdots & \boldsymbol{m}_k^{i-}\end{bmatrix} + \sqrt{c}\begin{bmatrix}0 & \sqrt{\boldsymbol{P}(t_k)} & -\sqrt{\boldsymbol{P}(t_k)}\end{bmatrix} \\
\boldsymbol{Y}_k^{i-} &= h_d(\boldsymbol{X}_k^{i-},\ k) \\
\mu_k^i &= \boldsymbol{Y}_k^{i-}\omega \\
\boldsymbol{S}_k^i &= \boldsymbol{Y}_k^{i-}\boldsymbol{W}[\boldsymbol{Y}_k^{i-}]^{\mathrm{T}} + \boldsymbol{R}_k \\
\mathbf{C}_k^i &= \boldsymbol{X}_k^{i-}\mathbf{W}[\mathbf{Y}_k^{i-}]^{\mathrm{T}}
\end{aligned} \tag{10}$$

where the expression $\boldsymbol{Y} = h_d(\boldsymbol{X}, k)$ means that the jth column of the matrix $\boldsymbol{Y}$, denoted as $\boldsymbol{Y}_j$, is calculated by $\boldsymbol{Y}_j = h_d(\boldsymbol{X}_j)$. Then calculate the filter gain $\boldsymbol{K}_k^i$, the state mean $\bar{\boldsymbol{m}}_k^i$ and the covariance $\hat{\boldsymbol{P}}_k^i$ as follows

$$\begin{aligned}
\boldsymbol{K}_k^i &= \boldsymbol{C}_k^i\{\boldsymbol{S}_k^i\}^{-1} \\
\bar{\boldsymbol{m}}_k^i &= \hat{\boldsymbol{m}}_k^{i-} + \boldsymbol{K}_k^i[\boldsymbol{y}_k - \mu_k^i] \\
\hat{\boldsymbol{P}}_k^i &= \hat{\boldsymbol{P}}_k^{i-} - \boldsymbol{K}_k^i\boldsymbol{S}_k^i\,\boldsymbol{K}_k^{i\,\mathrm{T}}
\end{aligned} \tag{11}$$

The final output of the CDUKF is just a Gaussian distribution $\mathcal{N}(\bar{\boldsymbol{m}}_k^i,\ \hat{\boldsymbol{P}}_k^i)$, which characterises the uncertainty on the estimate of the state at the kth time step.

3.2 Importance sampling

The EnCDUKF runs a bank of CDUKFs, yielding a batch of Gaussian distributions: $\mathcal{N}(\bar{\boldsymbol{m}}_k^i,\ \hat{\boldsymbol{P}}_k^i)$, $i = 1, \ldots, N$. Such distributions are then used as proposal to generate new importance samples. Specifically, it draws one sample from each Gaussian proposal: $\hat{\boldsymbol{m}}_k^i \sim \mathcal{N}(\bar{\boldsymbol{m}}_k^i,\ \hat{\boldsymbol{P}}_k^i)$, $i = 1, \ldots, N$. Then, for $i = 1, \ldots, N$, evaluate the importance weights up to a normalising constant

$$\tilde{w}_k^i \propto \frac{p(\boldsymbol{y}_k \mid \hat{\boldsymbol{m}}_k^i)p(\hat{\boldsymbol{m}}_k^i \mid \hat{\boldsymbol{m}}_{k-1}^i)}{N(\hat{\boldsymbol{m}}_k^i \mid \bar{\boldsymbol{m}}_k^i,\ \hat{\boldsymbol{P}}_k^i)} \tag{12}$$

The prior density $p(\hat{\boldsymbol{m}}_k^i \mid \hat{\boldsymbol{m}}_{k-1}^i)$ and the likelihood $p(\boldsymbol{y}_k \mid \hat{\boldsymbol{m}}_k^i)$ are determined by the state dynamic function and the measurement function in (1), respectively. Then normalise the importance weights as below

$$w_k^i = \tilde{w}_k^i \Big/ \sum_{i=1}^{N} \tilde{w}_k^i \tag{13}$$

Finally, the resampling step is applied to multiply/suppress particles with high/low importance weights w_k^i to obtain N random particles $\tilde{\boldsymbol{m}}_{0:k}^i,\ \tilde{\boldsymbol{P}}_{0:k}^i$. More detail about the resampling strategy can refer to [16].

4 Performance evaluation

We evaluate the performance of the proposed method via simulations. First, a continuous–discrete model used in [7, 9] is adopted here

$$\begin{aligned}
\mathrm{d}\boldsymbol{x}(t)/\mathrm{d}t &= -\sin\boldsymbol{x}(t) + v(t) \\
\boldsymbol{y}_k &= 0.5\sin(\boldsymbol{x}_k) + n_k
\end{aligned} \tag{14}$$

for which the state noise $v(t)$ and the measurement noise n_k are associated with spectral density $q_c = 0.01$ and $r_c = 0.1$, respectively. The proposed EnCDUKF method is compared with the continuous–discrete bootstrap particle filter [10] and the CDUKF [9] in estimation accuracy. The number of particles used for both the particle filter and the EnCDUKF is 50. The state dynamics is simulated to last over a time period of $T = 3$ s using the Euler–Maruyama

Table 1 RTAMSE comparison

Algorithms	RTAMSE
the proposed algorithm	0.1135
particle filter [10]	0.1726
CDUKF [9]	0.5047

scheme [17] with identical time steps $\delta = 0.001$ s. The measurement sampling interval is fixed with $T_s = 0.1$ s. A set of 100 times Monte Carlo (MC) simulations are applied to each method for performance comparison. The performance is evaluated via comparing the root time averaged mean square errors (RTAMSE)

$$\text{RTAMSE} = \sqrt{\frac{1}{SM}\sum_{j=1}^{S}\sum_{k=1}^{M}\left\|\boldsymbol{x}_{j,k} - \hat{\boldsymbol{x}}_{j,k}\right\|^2} \tag{15}$$

where $\boldsymbol{x}_{j,k}$ denotes the true state at the arrival time of the kth measurement in the jth simulation, $\hat{\boldsymbol{x}}_{j,k}$ is the estimate of $\boldsymbol{x}_{j,k}$, M is the total number of discrete measurements in a simulation and S is the number of independent simulations. Each algorithm is initialised with identical settings. The result is listed in Table 1, which demonstrates that the proposed method is much preferable to the benchmark algorithms in dealing with state filtering problems for continuous–discrete systems.

We randomly select an example run to check the estimate biases of these methods. The result is shown in Fig. 2, which indicates that the proposed method converges much faster than the others.

To further demonstrate the superiority of the proposed algorithm, we test a more challenging simulation case, in which a stochastic jump component is added to the system dynamic function as follows

$$\mathrm{d}\boldsymbol{x}(t)/\mathrm{d}t = -\sin\boldsymbol{x}(t) + \varrho(t)J(t) + w(t) \tag{16}$$

where $\varrho(t) \sim N(\mu_y,\ \sigma_y^2)$, $J(t) \sim Bi(1, \lambda)$ is an independent Bernoulli random variable with fixed intensity λ. Set $\lambda = 0.001$, $\mu_y = -0.2$ and $\sigma_y^2 = 0.1$ in this case. The other settings are the same as in the former simulation case.

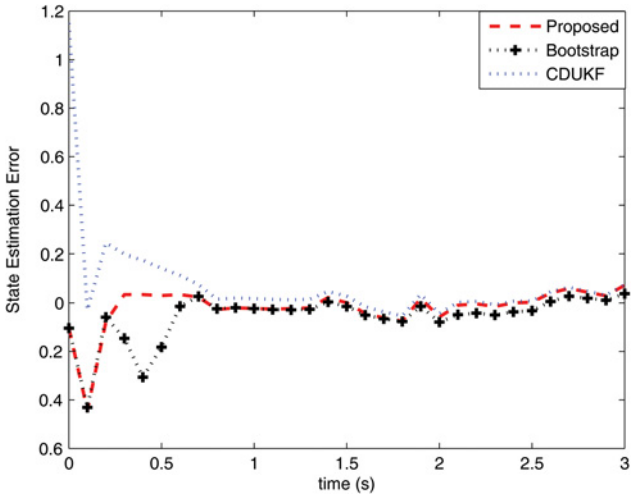

Fig. 2 *State estimation error of each algorithm in an example run*

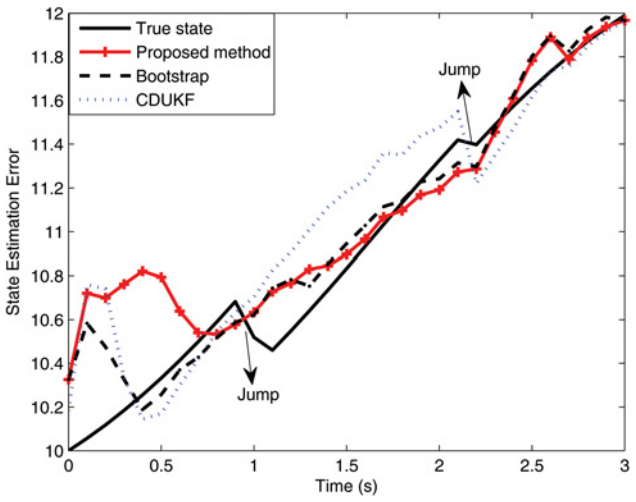

Fig. 3 *True and the estimated state trajectory of each algorithm*

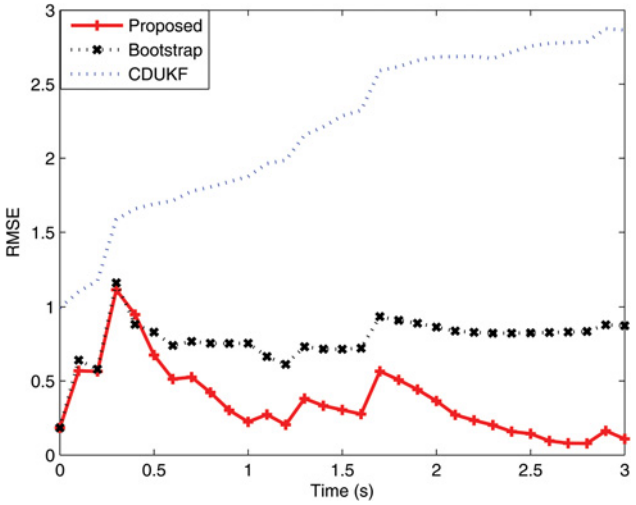

Fig. 4 *RMSE comparison*

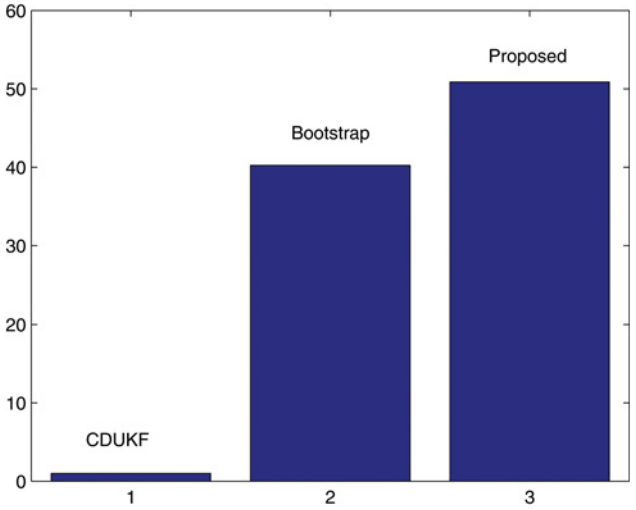

Fig. 5 *Average computing time of each algorithm*

Here the root mean square error (RMSE)

$$\mathrm{RMSE}_k = \sqrt{\frac{1}{S}\sum_{j=1}^{S}\left\|\boldsymbol{x}_{j,k} - \hat{\boldsymbol{x}}_{j,k}\right\|^2} \tag{17}$$

is used as the performance criterion for comparison. Hundred times MC simulations are applied for each algorithm. An illustration of the tracking trajectories for an example run is plotted in Fig. 3 and the RMSE comparison result is shown in Fig. 4.

The results again demonstrate that the proposed method is superior to benchmark filtering algorithms, that is, the bootstrap particle filter [10] and the CDUKF [9], for continuous–discrete systems, whereas the cost is the computing resource, as indicated by the average relative computing time shown in Fig. 5.

5 Conclusion

For continuous–discrete dynamic systems, the non-linear state filtering problem is challenging, since the existing algorithms all have their own limitations. This paper proposes a powerful candidate method capable of tackling any non-linear and/or non-Gaussian state filtering problems in continuous–discrete dynamic systems and the key idea is to use a bank of the CDUKFs to design proposal importance distributions of SIS, thus achieve the effect of combining both advantages of the CDUKF and SIS. Simulation results demonstrate the high efficiency and accuracy of our method.

6 Acknowledgments

This work was supported by the National Natural Science Foundation (NSF) of China (grant nos. 61302158, 61172166), the Provincial Science and Technology Plan (NSF) of Jiangsu province (grant no. BK20130869), the Natural Science research project for colleges and universities in Jiangsu province (grant no. 13KJB520019) and the Scientific Research Foundation of Nanjing University of Posts and Telecommunications (grant no. NY213030).

7 References

[1] Bar-Shalom Y., Li X.R., Kirubarajan T.: 'Estimation with applications to tracking and navigation' (Wiley Inter science, New York, 2001)
[2] Grewal M.S., Weill L.R., Andrews A.P.: 'Global positioning systems, inertial navigation and integration' (Wiley Inter science, New York, 2001)
[3] Stengel R.F.: 'Optimal control and estimation' (Dover Publications, Inc., New York, 1994)
[4] VanTrees H.L.: 'Detection, estimation, and modulation theory part I' (John Wiley Sons, New York, 1968)
[5] VanTrees H.L.: 'Detection, estimation, and modulation theory part II' (John Wiley Sons, New York, 1971)
[6] Jazwinski A.H.: 'Stochastic processes and filtering theory' (Academic Press, New York, 1970)
[7] Gelb A. (Ed.): 'Applied optimal estimation' (The MIT Press, Cambridge, 1974)
[8] Grewal M.S., Andrews A.P.: 'Kalman filtering, theory and practice using MATLAB' (Wiley Inter science, New York, 2001)
[9] Simo S.: 'On unscented Kalman filtering for state estimation of continuous-time nonlinear systems', *IEEE Trans. Autom. Control*, 2007, **52**, (9), pp. 1631–1641
[10] Tommi S., Simo S.: 'Application of Girsanov theorem to particle filtering of discretely observed continuous-time non-linear systems', *Bayesian Anal.*, 2008, **3**, (3), pp. 555–584
[11] Julier S.J., Uhlmann J.K.: 'Unscented filtering and nonlinear estimation', *Proc. IEEE*, 2004, **92**, (3), pp. 401–422
[12] Julier S.J., Uhlmann J.K.: 'New extension of the Kalman filter to nonlinear systems'. AeroSense'97, 1997, pp. 182–193
[13] Wan E.A., Van Der Merwe R.: 'The unscented Kalman filter for nonlinear estimation'. IEEE Adaptive Systems for Signal Processing, Communications, and Control Symp. , 2000, pp. 153–158
[14] Wan E.A., Van Der Merwe R.: 'The unscented Kalman filter', in Haykin S. (Ed.): 'Kalman filtering and neural networks' (Wiley, New York, 2001), pp. 221–280
[15] Van der Merwe R., Doucet A., de Freitas N., Wan E.: 'The unscented particle filter'. Technical Report, CUED/F-INFENG/TR 380, 2000
[16] Arnaud D., Simon G., Chistophe A.: 'On sequential Monte Carlo sampling methods for Bayesian filtering', *Stat. Comput.*, 2000, **10**, pp. 197–208
[17] Kloeden P.E., Platen E.: 'Numerical solution to stochastic differential equations' (Springer, 1999)