Journal of Systems Engineering and Electronics
Vol. 28, No. 6, December 2017, pp.1064 – 1071

Particle filter for nonlinear systems with multi-sensor asynchronous random delays

Junyi Zuo* and Xiaoping Zhong

School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China

Abstract: This paper is concerned with the recursive filtering problem for a class of discrete-time nonlinear stochastic systems in the presence of multi-sensor measurement delay. The delay occurs in a multi-step and asynchronous manner, and the delay probability of each sensor is assumed to be known or unknown. Firstly, a new model is constructed to describe the measurement process, based on which a new particle filter is developed with the ability to fuse multi-sensor information in the case of known delay probability. In addition, an online delay probability estimation module is introduced in the particle filtering framework, which leads to another new filter that can be implemented without the prior knowledge of delay probability. More importantly, since there is no complex iterative operation, the resulting filter can be implemented recursively and is suitable for many real-time applications. Simulation results show the effectiveness of the proposed filters.

Keywords: particle filter, nonlinear dynamic system, state estimation, measurement delay, multiple sensors.

DOI: 10.21629/JSEE.2017.06.04

1. Introduction

In the past few decades, the filtering problem has received considerable attention due to its wide applications [1 – 4]. In standard filtering problems, it is usually assumed that the measurements received by the data processing center are available in real-time. However, in many engineering applications, the arrival of the measurements can be affected by random delay, which poses a great challenge to the classical filtering theory.

Generally speaking, the filtering problems for systems with measurement delay can be divided into two categories according to whether the measurement packet is time-stamped or not. For time-stamped cases, it is exactly known when the current received measurement is collected by the sensor, and many filters have been developed [1,5,6].

For no time-stamp cases, the measurement delay is considered to randomly arise with a certain probability, and usually described by a set of stochastic parameters obeying Bernoulli distribution [7 – 9]. Following this idea, in [7] and [8], the improved extended and unscented Kalman filters were proposed for nonlinear dynamic systems with one-step or two-step random measurement delay. By using the Gaussian approximations of the one-step predictive probability densities of state and delayed measurement, a general processing framework and a novel Gaussian approximation filter were proposed for nonlinear systems with one-step delay [10]. Due to the inherent Gaussian assumption, these filters developed in Kalman filtering framework tend to perform poorly for highly nonlinear systems [11]. Recently, particle filters (PFs) [12,13] have become attractive alternatives to estimate nonlinear non-Gaussian systems since they can, in theory, provide optimal results asymptotically for arbitrary nonlinear non-Gaussian systems in the number of particles [14]. In [9], a filter, taking into account the multi-step random delays, has also been developed in the particle filtering framework.

The aforementioned filters have a common assumption that the delay probability must be known. However, this assumption may not hold in practice since the delay probability is often difficult to obtain, which makes these filters invalid [15,16 – 18]. In [16] and [18], the delay probability is estimated by using the expectation maximization (EM) algorithm and the maximum likelihood criterion, respectively. It has been reported both algorithms can produce satisfactory results if the delay probability is an unknown constant. To accommodate the unknown and time-varying delay probability, Wang et al. [17] proposed an improved identification algorithm in the EM framework. Based on the assumption that the measurement delay obeys

Manuscript received October 12, 2016.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61473227; 11472222), the Fundamental Research Funds for the Central Universities (3102015ZY001), the Aerospace Technology Support Fund of China (2014-HT-XGD), the Natural Science Foundation of Shaanxi Province (2015JM6304), and the Aeronautical Science Foundation of China (20151353018).

a Markov chain, the Gaussian-consensus filter for nonlinear systems with random measurement delay was proposed in [15]. The filter circumvents the problem of identifying the delay probability, but has to face another intractable problem, namely, how to obtain the transition probability matrix.

To the best of our knowledge, the state estimation problems for systems with measurement delay are far from being well solved. First, for the case of known delay probability, the filters mentioned above are only suitable for single-sensor-system or multi-sensor system but with synchronous delay. In fact, multi-sensor asynchronous delay is more common in practice. Second, for the case of unknown delay probability, the existing identification algorithms [16 – 18] are designed only for systems with one-step delay and, in addition, not suitable for real-time application. With further development of the networked control technique, it is necessary to design a fast suboptimal online estimator for nonlinear systems with multi-sensor asynchronous multi-step delay and unknown delay probability.

Motivated by the above analysis, we investigate these problems in this paper. The main contributions are two folds: i) by extending the algorithm in [9], the PF for systems with multi-sensor asynchronous delay and known delay probability is developed; ii) for the case of unknown delay probability, the developed PF in i) is extended by introducing a simple module for online estimating the posterior delay probability, and the resulting filter can produce decent results with a low computational cost.

2. Problem formulation

Consider the following nonlinear discrete-time stochastic system with K sensors

$$\boldsymbol{x}_t = \boldsymbol{f}_t(\boldsymbol{x}_{t-1}) + \boldsymbol{w}_{t-1} \tag{1}$$

$$\boldsymbol{z}_{t,k} = \boldsymbol{h}_{t,k}(\boldsymbol{x}_t) + \boldsymbol{v}_{t,k}, \quad k = 1, 2, \dots, K \tag{2}$$

where t is the time index, k is the sensor index, $\boldsymbol{x}_t \in \mathbf{R}^n$ denotes the state vector, $\boldsymbol{z}_{t,k} \in \mathbf{R}^{m_k}$ denotes the ideal measurement vector collected by the kth sensor. Define $\boldsymbol{z}_t = \{\boldsymbol{z}_{t,1}, \boldsymbol{z}_{t,2}, \dots, \boldsymbol{z}_{t,K}\}$. Let $\boldsymbol{w}_t \in \mathbf{R}^n$ and $\boldsymbol{v}_{t,k} \in \mathbf{R}^{m_k}$ denote the white process noise and measurement noise of the kth sensor respectively. The probability density functions of $p_w(\boldsymbol{w}_t)$ and $p_{v,k}(\boldsymbol{v}_{t,k})$, as well as the functions $\boldsymbol{f}_t(\cdot)$ and $\boldsymbol{h}_{t,k}(\cdot)$, are assumed to be known. The K measurement noise sequences $\{\boldsymbol{v}_{t,k}, t > 1\}_{k=1}^{K}$ are assumed independent of each other and also independent of the process noise sequence $\{\boldsymbol{w}_t, t > 0\}$.

The multi-sensor asynchronous random delay model can be formulated as

$$\boldsymbol{y}_{t,k} = \sum_{d=0}^{S_k} \lambda_{t,k}^d \boldsymbol{z}_{t-d,k}, \quad k = 1, 2, \dots, K \tag{3}$$

where $\boldsymbol{y}_{t,k} \in \mathbf{R}^{m_k}$ is the actual measurement received by the filter from the kth sensor, d is the delay step, $S_k = \min(t-1, l_k)$ where l_k is the maximum possible delay step of the kth sensor, $\lambda_{t,k}^d$ is a binary random variable taking the value of 0 or 1. For any given t and k, only one element in $\{\lambda_{t,k}^d\}_{d=0}^{S_k}$ takes the value of 1, while others take 0. The K actual measurements $\{\boldsymbol{y}_{t,1}, \boldsymbol{y}_{t,2}, \dots, \boldsymbol{y}_{t,K}\}$ are collectively represented as $\boldsymbol{y}_t$.

Define $\boldsymbol{r}_{t,k} = [\lambda_{t,k}^0, \lambda_{t,k}^1, \dots, \lambda_{t,k}^{S_k}]^{\mathrm{T}}$. It is assumed that $\boldsymbol{r}_{t_1,k}$ is independent of $\boldsymbol{r}_{t_2,k}$ if $t_1 \neq t_2$, $\boldsymbol{r}_{t,k_1}$ is independent of $\boldsymbol{r}_{t,k_2}$ if $k_1 \neq k_2$. Furthermore, the vector sequence $\{\boldsymbol{r}_{t,k}, t > 0\}$ is independent of $\{\boldsymbol{w}_t, t \geqslant 0\}$, $\{\boldsymbol{v}_{t,k}, t > 0\}_{k=1}^{K}$ and $\boldsymbol{x}_0$. For any given t and k, the state space of $\boldsymbol{r}_{t,k}$ consists of $S_k + 1$ column vectors $\{\boldsymbol{r}_{t,k}^d\}_{d=0}^{S_k}$. In $\boldsymbol{r}_{t,k}^d$, the $(d+1)$th element takes the value of 1, while others take 0. For example, if $S_k = 3$, then $\boldsymbol{r}_{t,k}^2 = [0, 0, 1, 0]^{\mathrm{T}}$. Random vector $\boldsymbol{r}_{t,k}$ obeys the discrete distribution $p(\boldsymbol{r}_{t,k} = \boldsymbol{r}_{t,k}^d) = p_{t,k}^d$, $d = 0, 1, \dots, S_k$, where $p_{t,k}^d \geqslant 0$ and $\sum_{d=0}^{S_k} p_{t,k}^d = 1$. The random event $\boldsymbol{r}_{t,k} = \boldsymbol{r}_{t,k}^d$ indicates that, for the sensor k, the measurement delayed by d sampling periods, i.e., $\boldsymbol{z}_{t-d,k}$, will be used to estimate the state at time t.

Our aim is to recursively estimate $\boldsymbol{x}_t$ based on all available actual measurements $\boldsymbol{y}_{1:t} = \{\boldsymbol{y}_\tau\}_{\tau=1}^{t}$. From the beginning of Section 3, particle filtering solutions for the two cases corresponding to known or unknown delay probability will be discussed.

Remark 1 As an extension to the one-step or two-step delay model in [7,8], a multi-step delay model has been reported in [9], which, however, can be seen as the special case of the model given by (1) – (3). Specifically, if for all $k \in \{1, 2, \dots, K\}$, l_k is a constant and $r_{t,1} = r_{t,2} = \cdots = r_{t,K}$ holds for any time index, then the proposed model reduces to the model in [9]. With the ability to represent the multi-sensor asynchronous delay, the proposed model can be used to describe more complex situations of measurement delay, including some special cases, e.g., measurements from some sensors undergo delay while those from others do not.

3. PF for multi-sensor asynchronous delay with known delay probability

PF is an approximate method to implement the Bayesian filter by Monte Carlo simulation, where the joint distribution $p(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$ is usually approximated by a set of weighted particle-trajectories as follows:

$$p(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1}) \approx \sum_{i=1}^{N} \pi_{t-1}^i \delta(\boldsymbol{x}_{0:t-1} - \boldsymbol{x}_{0:t-1}^i) \tag{4}$$

where $\delta(\cdot)$ is the Dirac delta function, $\boldsymbol{x}_{0:t-1}^i = \{\boldsymbol{x}_0^i, \boldsymbol{x}_1^i, \dots, \boldsymbol{x}_{t-1}^i\}$ is the ith particle-trajectory from time 0 to $t-1$ with normalized weight π_{t-1}^i. If $\boldsymbol{x}_{0:t-1}^i$ is drawn from the importance density function $q(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$, then

$$\pi_{t-1}^i \propto \frac{p(\boldsymbol{x}_{0:t-1}^i|\boldsymbol{y}_{1:t-1})}{q(\boldsymbol{x}_{0:t-1}^i|\boldsymbol{y}_{1:t-1})} \tag{5}$$

where $\propto$ stands for 'proportional to'. With the joint distribution $p(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$ available, it is straightforward to represent the filtering distribution $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ by the weighted particles $\{\boldsymbol{x}_{t-1}^i, \pi_{t-1}^i\}_{i=1}^N$ as $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} \pi_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i)$, where $\boldsymbol{x}_{t-1}^i$ is the component of $\boldsymbol{x}_{0:t-1}^i$ corresponding to time $t-1$.

Suppose, at time $t-1$, the weighted particle-trajectories $\{\boldsymbol{x}_{0:t-1}^i, \pi_{t-1}^i\}_{i=1}^N$ are available and the posterior joint distribution $p(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$ can be approximated by (4). When the new measurement $\boldsymbol{y}_t$ arrives, a new weighted particle-trajectory set $\{\boldsymbol{x}_{0:t}^i, \pi_t^i\}_{i=1}^N$ is required to approximate $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$, where $\boldsymbol{x}_{0:t}^i$ is drawn from the importance density $q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$. If $q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ is chosen to factorize such that

$$q(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t})q(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1}) \tag{6}$$

then one can obtain $\boldsymbol{x}_{0:t}^i$ by augmenting each of the existing particle-trajectories $\boldsymbol{x}_{0:t-1}^i \sim q(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$ with the new state $\boldsymbol{x}_t^i \sim q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}^i, \boldsymbol{y}_{1:t})$. To derive the weight update equation, we manipulate $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ as follows:

$$\begin{gathered}p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) = \\ \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_{0:t}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} = \\ \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_{0:t}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \propto \\ p(\boldsymbol{y}_t|\boldsymbol{x}_{0:t}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1}).\end{gathered} \tag{7}$$

Substituting (6) and (7) into (5), we have

$$\begin{gathered}\pi_t^i \propto \\ \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_{0:t}^i, \boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{0:t-1}^i, \boldsymbol{y}_{1:t})} \cdot \frac{p(\boldsymbol{x}_{0:t-1}^i|\boldsymbol{y}_{1:t-1})}{q(\boldsymbol{x}_{0:t-1}^i|\boldsymbol{y}_{1:t-1})} = \\ \pi_{t-1}^i \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_{0:t}^i, \boldsymbol{y}_{1:t-1})p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{q(\boldsymbol{x}_t^i|\boldsymbol{x}_{0:t-1}^i, \boldsymbol{y}_{1:t})}.\end{gathered} \tag{8}$$

If we choose $q(\boldsymbol{x}_t|\boldsymbol{x}_{0:t-1}^i, \boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$, and perform resampling at each filter cycle to tackle the particle degeneracy problem [12], which results in $\pi_{t-1}^i = 1/N$, $i = 1, 2, \dots, N$, then (8) becomes

$$\pi_t^i \propto p(\boldsymbol{y}_t|\boldsymbol{x}_{0:t}^i, \boldsymbol{y}_{1:t-1}). \tag{9}$$

For no delay systems, according to the assumption that $\boldsymbol{y}_t$ only depends on the current state, $p(\boldsymbol{y}_t|\boldsymbol{x}_{0:t}^i, \boldsymbol{y}_{1:t-1})$ can be simplified as $p(\boldsymbol{y}_t|\boldsymbol{x}_t^i)$. However, in the presence of multi-step random delay, $\boldsymbol{y}_t$ may depend on previous states and measurements. Define $S = \max\{S_k\}_{k=1}^K$, then $p(\boldsymbol{y}_t|\boldsymbol{x}_{0:t}^i, \boldsymbol{y}_{1:t-1}) = p(\boldsymbol{y}_t|\boldsymbol{x}_{t-S:t}^i, \boldsymbol{y}_{t-S:t-1})$. If we further assume that $p(\boldsymbol{y}_t|\boldsymbol{x}_{t-S:t}^i, \boldsymbol{y}_{t-S:t-1})$ can be approximated by $p(\boldsymbol{y}_t|\boldsymbol{x}_{t-S:t}^i)$ for simplicity [9,18], and multi-sensor measurements are mutually independent, then (9) can be written as

$$\begin{gathered}\pi_t^i \propto p(\boldsymbol{y}_t|\boldsymbol{x}_{t-S:t}^i) = \prod_{k=1}^{K} p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S:t}^i) = \\ \prod_{k=1}^{K} p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S_k:t}^i)\end{gathered} \tag{10}$$

where

$$\begin{gathered}p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S_k:t}^i) = \\ \sum_{d=0}^{S_k} p(\boldsymbol{y}_{t,k}, \boldsymbol{r}_{t,k} = \boldsymbol{r}_{t,k}^d|\boldsymbol{x}_{t-S_k:t}^i) = \\ \sum_{d=0}^{S_k} p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k} = \boldsymbol{r}_{t,k}^d, \boldsymbol{x}_{t-S_k:t}^i)p(\boldsymbol{r}_{t,k} = \boldsymbol{r}_{t,k}^d) = \\ \sum_{d=0}^{S_k} p_{t,k}^d p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-d}^i) = \\ \sum_{d=0}^{S_k} p_{t,k}^d p_{v,k}(\boldsymbol{y}_{t,k} - \boldsymbol{h}_{t-d,k}(\boldsymbol{x}_{t-d}^i)).\end{gathered} \tag{11}$$

Obviously, to evaluate $p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S_k:t}^i)$, not only the particle $\boldsymbol{x}_t^i$ but also its ancestors $\boldsymbol{x}_{t-1}^i, \boldsymbol{x}_{t-2}^i, \dots, \boldsymbol{x}_{t-S_k}^i$ should be available. To this end, when performing resampling at each filtering step, we define an intermediate array to record the relationship of particle indexes between the current particle set and the previous one. By querying these intermediate arrays, $\boldsymbol{x}_{t-1}^i$ can be found in $\{\boldsymbol{x}_{t-1}^j\}_{j=1}^N$ with $\boldsymbol{x}_t^i$ known, and then $\boldsymbol{x}_{t-2}^i$ can be found in $\{\boldsymbol{x}_{t-2}^j\}_{j=1}^N$ with $\boldsymbol{x}_{t-1}^i$ known. This process continues until all the ancestors $\boldsymbol{x}_{t-1}^i, \boldsymbol{x}_{t-2}^i, \dots, \boldsymbol{x}_{t-S_k}^i$ of $\boldsymbol{x}_t^i$ are obtained.

Substitution of (11) into (10) yields

$$\pi_t^i \propto \prod_{k=1}^{K} \left(\sum_{d=0}^{S_k} p_{t,k}^d p_{v,k}(\boldsymbol{y}_{t,k} - \boldsymbol{h}_{t-d,k}(\boldsymbol{x}_{t-d}^i)) \right). \tag{12}$$

With $\{\boldsymbol{x}_{0:t}^i, \pi_t^i\}_{i=1}^N$ available, the joint posterior density $p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})$ can be formulated as

$$p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t}) \approx \sum_{i=1}^{N} \pi_t^i \delta(\boldsymbol{x}_{0:t} - \boldsymbol{x}_{0:t}^i). \tag{13}$$

The optimal estimate of $\boldsymbol{x}_t$ in the minimum-mean-square-error sense can be obtained by

$$\widehat{\boldsymbol{x}}_t = \int \boldsymbol{x}_t p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})\mathrm{d}\boldsymbol{x}_t \approx$$

$$\int \boldsymbol{x}_t \int p(\boldsymbol{x}_{0:t}|\boldsymbol{y}_{1:t})\mathrm{d}\boldsymbol{x}_{0:t-1}\mathrm{d}\boldsymbol{x}_t =$$

$$\int \boldsymbol{x}_t \int \sum_{i=1}^{N} \pi_t^i \delta(\boldsymbol{x}_{0:t} - \boldsymbol{x}_{0:t}^i)\mathrm{d}\boldsymbol{x}_{0:t-1}\mathrm{d}\boldsymbol{x}_t =$$

$$\sum_{i=1}^{N} \pi_t^i \boldsymbol{x}_t^i. \tag{14}$$

The PF for multi-sensor asynchronous delay with known delay probability can be summarized by Algorithm 1.

Algorithm 1

Step 1 Draw N particles from the initial distribution $\boldsymbol{x}_0^i \sim p(\boldsymbol{x}_0), i = 1, 2, \ldots, N$; set $t = 1$ and $\pi_0^i = 1/N$ for all $i = 1, 2, \ldots, N$.

Step 2

For $t = 1, 2, 3, \ldots$

(i) Draw particle $\boldsymbol{x}_t^i$ from $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$ for all $i = 1, \ldots, N$;

(ii) Calculate the normalized weight π_t^i according to (12) for all $i = 1, \ldots, N$, where $\boldsymbol{x}_{t-d}^i$ is the point on $\boldsymbol{x}_{0:t}^i$ corresponding to time $t - d$;

(iii) Calculate the state estimate $\widehat{\boldsymbol{x}}_t = \sum_{i=1}^{N} \boldsymbol{x}_t^i \pi_t^i$;

(iv) Perform resampling.

End For

Note that Step 2(i) is essentially the prediction step in Bayesian filtering, and can be implemented without the measurement. The purpose of this step is to obtain the particle at time t. In fact, the measurement delay is only considered in the update stage, which is reflected in the new likelihood evaluating method and particle weighting method.

4. PF for multi-sensor asynchronous delay with unknown delay probability

It is infeasible to evaluate the particle weights by (12) without the knowledge of delay probability, and consequently Algorithm 1 is invalid. To tackle this problem, inspired by the idea in [19] where a PF is developed to deal with the systems in presence of process noise with unknown statistics, we propose to use the posterior distribution $p(\boldsymbol{r}_{t,k}|\boldsymbol{y}_{1:t})$ to approximate $p(\boldsymbol{r}_{t,k})$. Thus, the resulting PF is the same as Algorithm 1, except that the calculation module to determine $p(\boldsymbol{r}_{t,k}|\boldsymbol{y}_{1:t}), k = 1, 2, \ldots, K$ should be introduced in.

According to the Bayesian rule, we have

$$p(\boldsymbol{r}_{t,k}|\boldsymbol{y}_{1:t}) = p(\boldsymbol{r}_{t,k}|\boldsymbol{y}_{1:t,k}) =$$

$$\frac{p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}, \boldsymbol{y}_{1:t-1,k})p(\boldsymbol{r}_{t,k}|\boldsymbol{y}_{1:t-1,k})}{p(\boldsymbol{y}_{t,k}|\boldsymbol{y}_{1:t-1,k})}. \tag{15}$$

The first equality comes from the fact that the delay of the kth sensor does not depend on the measurements of other sensors. Given that $\boldsymbol{r}_{t,k}$ does not depend on $\boldsymbol{y}_{1:t-1,k}$, and $p(\boldsymbol{y}_{t,k}|\boldsymbol{y}_{1:t-1,k})$ is a normalizing constant denoted as c, (15) can be rewritten as

$$p(\boldsymbol{r}_{t,k}|\boldsymbol{y}_{1:t}) = \frac{1}{c}p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}, \boldsymbol{y}_{1:t-1,k})p(\boldsymbol{r}_{t,k}). \tag{16}$$

Since there is no knowledge of $p(\boldsymbol{r}_{t,k})$, it is reasonable to assume that $\boldsymbol{r}_{t,k}$ is uniformly distributed in its state space, i.e.,

$$p(\boldsymbol{r}_{t,k}) = \frac{1}{S_k + 1}\sum_{d=0}^{S_k} \delta(\boldsymbol{r}_{t,k} - \boldsymbol{r}_{t,k}^d). \tag{17}$$

Substitution of (17) into (16) yields

$$p(\boldsymbol{r}_{t,k}|\boldsymbol{y}_{1:t}) =$$

$$\frac{1}{c(S_k + 1)}\sum_{d=0}^{S_k} p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d, \boldsymbol{y}_{1:t-1,k})\delta(\boldsymbol{r}_{t,k} - \boldsymbol{r}_{t,k}^d) =$$

$$\sum_{d=0}^{S_k} \alpha_{t,k}^d \delta(\boldsymbol{r}_{t,k} - \boldsymbol{r}_{t,k}^d) \tag{18}$$

where

$$\alpha_{t,k}^d = \frac{p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d, \boldsymbol{y}_{1:t-1,k})}{c(S_k + 1)}. \tag{19}$$

That is to say, the posterior delay probability of the kth sensor can be written as

$$p(\boldsymbol{r}_{t,k} = \boldsymbol{r}_{t,k}^d|\boldsymbol{y}_{1:t}) = \alpha_{t,k}^d, \quad d = 0, 1, \ldots, S_k. \tag{20}$$

If $\alpha_{t,k}^d$ ($d = 0, 1, \ldots, S_k$) is available, then $p_{t,k}^d$ in (12) can be approximated by $\alpha_{t,k}^d$. Next, the algorithm to determine $\alpha_{t,k}^d$ is given.

It is easy to verify that $c(S_k + 1) = \sum_{d=0}^{S_k} p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d, \boldsymbol{y}_{1:t-1,k})$. Hence, we have

$$\alpha_{t,k}^d = \frac{p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d, \boldsymbol{y}_{1:t-1,k})}{\sum_{d=0}^{S_k} p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d, \boldsymbol{y}_{1:t-1,k})}. \tag{21}$$

Equations (18) and (21) reflect the fact that $p(\boldsymbol{r}_{t,k}|\boldsymbol{y}_{1:t})$ depends solely on $p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d, \boldsymbol{y}_{1:t-1,k}), d = 0, 1, \ldots, S_k$, which can be expressed as

$$p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d, \boldsymbol{y}_{1:t-1,k}) =$$

$$\int p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S_k:t},\boldsymbol{r}_{t,k}^d,\boldsymbol{y}_{1:t-1,k})\cdot p(\boldsymbol{x}_{t-S_k:t}|\boldsymbol{r}_{t,k}^d,\boldsymbol{y}_{1:t-1,k})d\boldsymbol{x}_{t-S_k:t}= \int p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S_k:t},\boldsymbol{r}_{t,k}^d,\boldsymbol{y}_{1:t-1,k})\cdot p(\boldsymbol{x}_{t-S_k:t}|\boldsymbol{y}_{1:t-1,k})d\boldsymbol{x}_{t-S_k:t}. \tag{22}$$

In order to simplify the evaluation of the likelihood, as has been done in Section 3 and [9,18], we assume that $p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S_k:t},\boldsymbol{r}_{t,k}^d,\boldsymbol{y}_{1:t-1,k})$ can be approximated by $p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S_k:t},\boldsymbol{r}_{t,k}^d)$, thus

$$p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d,\boldsymbol{y}_{1:t-1,k})= \int p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S_k:t},\boldsymbol{r}_{t,k}^d)p(\boldsymbol{x}_{t-S_k:t}|\boldsymbol{y}_{1:t-1,k})d\boldsymbol{x}_{t-S_k:t}. \tag{23}$$

By running a separate PF for each sensor, the joint distribution $p(\boldsymbol{x}_{t-S_k:t}|\boldsymbol{y}_{1:t-1,k})$ can be represented by weighted local particle-trajectories as the discrete sum form, by which $p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d,\boldsymbol{y}_{1:t-1,k})$ and then $\alpha_{t,k}^d$ can be evaluated. However, it is computationally expensive to obtain the discrete representations for all $p(\boldsymbol{x}_{t-S_k:t}|\boldsymbol{y}_{1:t-1,k})$ $(k=0,1,\dots,K)$. To reduce the computational complexity, the following approximation strategy is adopted.

Specifically, after the filter cycle at time $t-1$, the trajectory-weight set $\{\boldsymbol{x}_{0:t-1}^i,\pi_{t-1}^i\}_{i=1}^N$ will be obtained to represent $p(\boldsymbol{x}_{0:t-1}|\boldsymbol{y}_{1:t-1})$. Since the resampling step is introduced in each filter cycle, the weight π_{t-1}^i equals $1/N$ for all $i=1,2,\dots,N$. By simply extracting the required components from these particle-trajectories and keeping the weights unchanged, local trajectory-weight-set $\{\boldsymbol{x}_{t-S_k:t-1}^i,\pi_{t-1}^i=1/N\}_{i=1}^N$ can be obtained to represent the joint distribution $p(\boldsymbol{x}_{t-S_k:t-1}|\boldsymbol{y}_{1:t-1})$, where $\boldsymbol{x}_{t-S_k:t-1}^i$ is the segment of $\boldsymbol{x}_{0:t-1}^i$ from time $t-S_k$ to $t-1$. Then, we use (1) to propagate each particle of $\{\boldsymbol{x}_{t-1}^i\}_{i=1}^N$ to time t and get particle set $\{\boldsymbol{x}_t^i\}_{i=1}^N$, where $\boldsymbol{x}_{t-1}^i$ is the point on $\boldsymbol{x}_{t-S_k:t-1}^i$ corresponding to time $t-1$. Letting $\boldsymbol{x}_{t-S_k:t}^i=\{\boldsymbol{x}_{t-S_k:t-1}^i,\boldsymbol{x}_t^i\}$ for each $i\in\{1,2,\dots,N\}$, we obtain the new local trajectory-weight set $\{\boldsymbol{x}_{t-S_k:t}^i,\pi_{t-1}^i=1/N\}_{i=1}^N$, which is used to approximate $p(\boldsymbol{x}_{t-S_k:t}|\boldsymbol{y}_{1:t-1,k})$ for simplicity, i.e.,

$$p(\boldsymbol{x}_{t-S_k:t}|\boldsymbol{y}_{1:t-1,k})\approx\sum_{i=1}^N\frac{1}{N}\delta(\boldsymbol{x}_{t-S_k:t}-\boldsymbol{x}_{t-S_k:t}^i). \tag{24}$$

Substitution of (24) into (23) yields

$$p(\boldsymbol{y}_{t,k}|\boldsymbol{r}_{t,k}^d,\boldsymbol{y}_{1:t-1,k})= \frac{1}{N}\sum_{i=1}^N p(\boldsymbol{y}_{t,k}|\boldsymbol{x}_{t-S_k:t}^i,\boldsymbol{r}_{t,k}^d)= \frac{1}{N}\sum_{i=1}^N p_{v,k}(\boldsymbol{y}_{t,k}-\boldsymbol{h}_{t-d,k}(\boldsymbol{x}_{t-d}^i)). \tag{25}$$

Combining (25) with (21), we have

$$\alpha_{t,k}^d=\frac{\displaystyle\sum_{i=1}^N p_{v,k}(\boldsymbol{y}_{t,k}-\boldsymbol{h}_{t-d,k}(\boldsymbol{x}_{t-d}^i))}{\displaystyle\sum_{b=0}^{S_k}\sum_{i=1}^N p_{v,k}(\boldsymbol{y}_{t,k}-\boldsymbol{h}_{t-b,k}(\boldsymbol{x}_{t-b}^i))}. \tag{26}$$

The PF for multi-sensor asynchronous delay with unknown delay probability can be summarized by Algorithm 2.

Algorithm 2

Step 1 Draw N particles from the initial distribution $x_0^i\sim p(x_0), i=1,2,\dots,N$; set $t=1$ and $\pi_0^i=1/N$ for all $i=1,2,\dots,N$.

Step 2

For $t=1,2,3,\dots$

(i) For all $i=1,2,\dots,N$, draw particle $\boldsymbol{x}_t^i$ from $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^i)$, and then define the ith particle-trajectory up to time t as $\boldsymbol{x}_{0:t}^i=\{\boldsymbol{x}_{0:t-1}^i,\boldsymbol{x}_t^i\}$;

(ii) For $k=1,2,\dots,K$

For $d=0,1,\dots,S_k$

Calculate $\alpha_{t,k}^d$ by (26) and set $p_{t,k}^d=\alpha_{t,k}^d$;

End For

End For

(iii) Calculate the normalized weight π_t^i according to (12) for all $i=1,\dots,N$;

(iv) Calculate the state estimate $\widehat{\boldsymbol{x}}_t=\displaystyle\sum_{i=1}^N\boldsymbol{x}_t^i\pi_t^i$;

(v) Perform resampling.

End For

5. Numerical examples

Example 1 Univariate non-stationary growth model

This example is used to test Algorithm 1. Consider the following model with state and measurement equations given by

$$x_t=0.5x_{t-1}+25\frac{x_{t-1}}{1+x_{t-1}^2}+8\cos(1.2t)+w_{t-1} \tag{27}$$

$$\text{Sensor 1}: z_{t,1}=\frac{(x_t+1)^2}{20}+v_{t,1} \tag{28}$$

$$\text{Sensor 2}: z_{t,2}=\frac{(x_t-1)^2}{20}+v_{t,1} \tag{29}$$

where $w_{t-1}\sim\mathcal{N}(0,0.3)$ is the white process noise, $v_{t,1}\sim\mathcal{N}(0,2)$ and $v_{t,2}\sim\mathcal{N}(0,2)$ are the white measurement noises of Sensor 1 and Sensor 2 respectively, $x_0\sim\mathcal{N}(1,0.3)$ is the initial state. The actual measurement $\boldsymbol{y}_t$ is related to the ideal measurement $\boldsymbol{z}_t=[z_{t,1},z_{t,2}]^{\mathrm{T}}$ by (3) with $l_1=l_2=2$.

The delay probabilities are set as follows: $p^0_{t,1} = p^0_{t,2} = 1$ when $t = 1$; $p^0_{t,1} = p^0_{t,2} = 0.7$, $p^1_{t,1} = p^1_{t,2} = 0.3$ when $t = 2$; $p^0_{t,1} = 0.5$, $p^1_{t,1} = 0.3$, $p^2_{t,1} = 0.2$, $p^0_{t,2} = 0.6$, $p^1_{t,2} = 0.3$, $p^2_{t,2} = 0.1$ when $t > 2$. Four algorithms are used to estimate the state sequence for $t = 1, 2, \ldots, 100$ with $N = 100$, which are abbreviated to SIR12, PF-Kn1, PF-Kn2 and PF-Kn12 respectively. In this paper, we have a convention that the number suffix appended to the algorithm name indicates which sensors are used. SIR12 is the sampling importance resampling (SIR) filter [12] with Sensor 1 and Sensor 2 used, but the measurement delay is ignored. Other three algorithms are implemented with the assumption that the delay probability is known. Among them, PF-Kn1 and PF-Kn2 are essentially the PF proposed in [9] with Sensor 1 and Sensor 2 used respectively. PF-Kn12 is the proposed fusion filter with the two sensors used.

Each of the four algorithms is run 1 000 times. For clarity, only the root mean squared errors (RMSEs) of the first 80 independent runs are shown in Fig. 1. Table 1 summarizes the overall performances of these algorithms, where the second and third columns list the average RMSE (ARMSE) and the variance of RMSE respectively, and the fourth column lists the average run time per run. It is obvious that, although there is a mechanism to fuse multi-sensor information, SIR12 produces the poorest result, which indicates that multi-step delay of measurement makes the traditional PF face great difficulty. Arbitrary neglect of the delay will significantly deteriorate the performance. Although both PF-Kn1 and PF-Kn2 only use a single sensor, they produce more accurate results than SIR12 (Note PF-Kn2 performs better slightly than PF-Kn1, which is attributed to the more mild measurement delay of sensor 2). The proposed PF-Kn12 gives the best results, which is benefit from the proper treatment of multi-sensor asynchronous delay problem and, consequently, more efficient fusion of multi-sensor information. However, its run time is twice as much as that of SIR12. Hence, the improved performance of PF-Kn12 is achieved at the price of the increased computational cost.

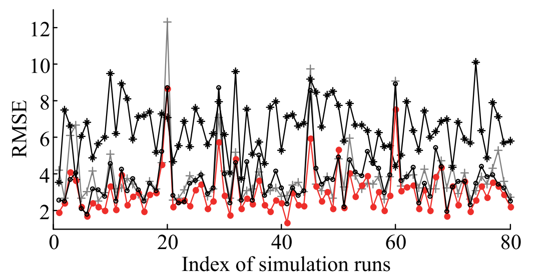

Fig. 1 RMSEs of SIR12, PF-Kn1, PF- Kn2 and PF-Kn12

Table 1 Performance comparison of SIR12, PF-Kn1, PF-Kn2 and PF-Kn12

Filter	ARMSE	RMSE variance	Run time per run/s
SIR12	6.537	2.087	0.169
PF-Kn1	4.061	3.490	0.200
PF-Kn2	3.951	2.765	0.199
PF-Kn12	3.187	2.535	0.328

Example 2 Bearing only tracking

This example is used to test Algorithm 2. Consider the problem of tracking an object moving in a two-dimensional plane, the measurements, gathered at fixed intervals, are the bearings with respect to two sensors. Assume that the object moves according to the following process model

$$\boldsymbol{x}_t = \boldsymbol{F}\boldsymbol{x}_{t-1} + \boldsymbol{G}\boldsymbol{v}_{t-1} \tag{30}$$

with $\boldsymbol{x}_t = [x_t, x'_t, y_t, y'_t]^{\mathrm{T}}$, $\boldsymbol{v}_t = [v_{xt}, v_{yt}]^{\mathrm{T}}$,

$$\boldsymbol{F} = \boldsymbol{I}_2 \otimes \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{G} = \boldsymbol{I}_2 \otimes \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$$

where x_t and y_t denote the Cartesian coordinates of the target at time t, x'_t and y'_t denote the velocities in two directions respectively, $\boldsymbol{I}_2$ denotes the 2×2 identity matrix, $\otimes$ denotes the Kronecker product, $\boldsymbol{v}_{t-1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ is the white process noise, where $\boldsymbol{Q} = \sigma_v^2 \boldsymbol{I}_2$ is the covariance matrix. The initial distribution of the state vector is Gaussian, i.e., $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{P}_0)$.

The measurement equations are given by

$$z_{t,1} = \arctan(y_t/x_t) + w_{t,1} \tag{31}$$

$$z_{t,2} = \arctan(y_t/(x_t - 80)) + w_{t,2} \tag{32}$$

where the measurement noises $w_{t,1} \sim \mathcal{N}(0, \sigma^2_{w1})$ and $w_{t,2} \sim \mathcal{N}(0, \sigma^2_{w2})$ are assumed to be mutually independent. The actual measurement y_t is related to the ideal measurement $\boldsymbol{z}_t = [z_{t,1}, z_{t,2}]^{\mathrm{T}}$ by (3) with $l_1 = l_2 = 2$.

Numerical simulations are implemented 500 times over 50 time steps. The required parameters are specified as $\sigma_v = 0.1\ \mathrm{m/s^2}$, $\sigma_{w1} = 5$ mrad, $\sigma_{w2} = 5$ mrad, $\boldsymbol{\mu}_0 = [-0.05\ \mathrm{m}, 0.001\ \mathrm{m/s}, 0.7\ \mathrm{m}, -0.055\ \mathrm{m/s}]^{\mathrm{T}}$, $\boldsymbol{P}_0 = \mathrm{diag}[(0.1\ \ \mathrm{m})^2, (0.005\ \ \mathrm{m/s})^2, (0.1\ \ \mathrm{m})^2, (0.01\ \ \mathrm{m/s})^2]$, $N = 1\,000$. When $t \leqslant 2$, the delay probabilities are set to be identical to Example 1; when $t > 2$, they are set as $p^0_{t,1} = 0.2$, $p^1_{t,1} = 0.6$, $p^2_{t,1} = 0.2$, $p^0_{t,2} = 0.2$, $p^1_{t,2} = 0.6$, $p^2_{t,2} = 0.2$.

Three filters, including SIR12, PF-Kn12 and PF-Un12, are used to estimate the state sequence. (Note PF-Un12 stands for the PF given by Algorithm 2). The RMSEs of the first 80 simulation runs are shown in Fig. 2 with four subplots illustrating the RMSEs of position and velocity in X and Y directions, respectively. Table 2 lists the ARMSEs and average run-times per run. Due to the consideration of measurement delay, the estimation accuracies of PF-Kn12 and PF-Un12 are much higher than that of SIR12. By using the precise delay probability, PF-Kn12 produces the best results among the three filters. In comparison, PF-Un12, although no knowledge of delay probability is used, can yield decent results comparable to PF-Kn12 with slightly increased computational cost.

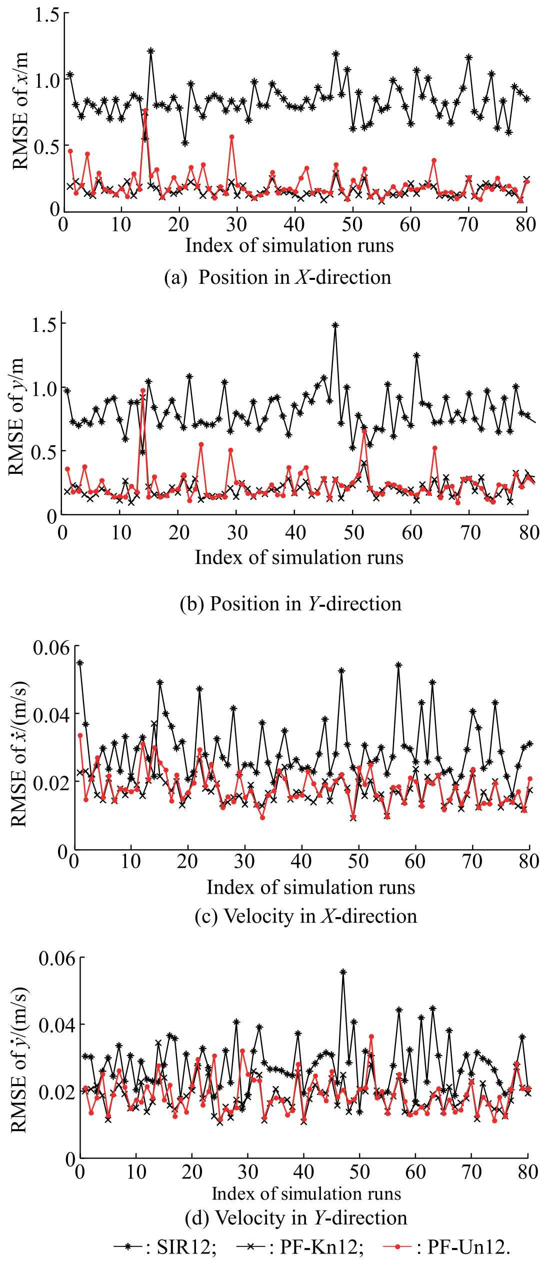

Fig. 2 RMSEs of position and velocity for bearing only tracking problem

Table 2 Performance comparison of SIR12, PF-Kn12 and PF-Un12 for bearing only tracking

Algorithm	ARMSE-x/m	ARMSE-y/m	ARMSE-$\dot{x}$/(m/s)	ARMSE-$\dot{y}$/(m/s)	Run time per run/s
SIR12	0.815 9	0.803 7	0.028 3	0.027 7	1.541
PF-Kn12	0.166 8	0.207 9	0.017 0	0.018 5	3.336
PF-Un12	0.227 7	0.258 3	0.018 3	0.019 1	4.080

It is of course true that the filters of identifying the delay probability first and then estimating the state cannot outperform PF-Kn12, where the true delay probability is used. Therefore, even if there are some EM based methods with the ability to deal with multi-sensor multi-step delay (in fact, they do not exist so far), they would not substantially outperform the proposed filter in terms of estimation accuracy. On the contrary, large amounts of computational cost would be introduced in them. In this sense, this paper provides a simple and practical method, which has acceptable estimation accuracy for the problem of interest.

6. Conclusions

In this paper, the multi-sensor information fusion framework of PF is extended to the situation of multi-step asynchronous measurement delay. Two PFs, i.e., PF-Kn and PF-Un, are proposed to deal with the two cases with the delay probability known and unknown, respectively. Their effectiveness has been demonstrated by two typical numerical examples. Since multi-sensor asynchronous delay is very common in practice, the proposed PF-Kn has important application value. In addition, in networked remote-sensing systems, it is often required to rapidly process multi-sensor multi-step delay with delay probability unavailable. This paper provides a practical method for this problem, which not only can produce decent results but also have low additional computational cost due to no complex iterative optimization involved.

References

[1] E. Besada-Portas, J. A. Lopez-Orozco, J. A. Besada, et al. Multisensor out of sequence data fusion for estimating the state of discrete control systems. *IEEE Trans. on Automatic Control*, 2009, 54(7): 1728 – 1732.

[2] F. Deng, J. Chen, C. Chen. Adaptive unscented Kalman filter for parameter and state estimation of nonlinear high-speed objects. *Journal of Systems Engineering and Electronics*, 2013, 24(4): 655 – 665.

[3] S. Zhang, Y. Zhao, M. Li, et al. Optimal filtering for uncertain systems with stochastic nonlinearities, correlated noises and missing measurements. *Journal of Systems Engineering and Electronics*, 2015, 26(5): 1052 – 1059.

[4] L. Xu, X. Li, Z. Duan. Hybrid grid multiple-model estimation with application to maneuvering target tracking. *IEEE Trans. on Aerospace and Electronic Systems*, 2016, 52(1): 122 – 136.

[5] K. Berntorp, A. Robertsson, E. K. Årzé. Blackwellized particle filters with out-of-sequence measurement processing. *IEEE Trans. on Signal Processing*, 2014, 62(24): 6454 – 6467.

[6] Z. Shuo, Y. Bar-Shalom. Out-of-sequence measurement processing for particle filter: exact Bayesian solution. *IEEE Trans. on Aerospace and Electronic Systems*, 2012, 48(4): 2818 – 2831.

[7] A. Hermoso-Carazo, J. Linares-Pérez. Extended and unscented filtering algorithms using one-step randomly delayed observations. *Applied Mathematics and Computation*, 2007, 190(2): 1375 – 1393.

[8] A. Hermoso-Carazo, J. Linares-Pérez. Unscented filtering algorithm using two-step randomly delayed observations in nonlinear systems. *Applied Mathematical Modelling*, 2009, 33(9): 3705 – 3717.

[9] Y. Huang, Y. Zhang, N. Li, et al. Particle filter for nonlinear systems with multiple step randomly delayed measurements. *Electronics Letters*, 2015, 51(23): 1859 – 1861.

[10] X. Wang, Y. Liang, Q. Pan, et al. Gaussian filter for nonlinear systems with one-step randomly delayed measurements. *Automatica*, 2013, 49(4): 976 – 986.

[11] W. Zhang, J. Zuo, Q. Guo, et al. Multisensor information fusion scheme for particle filter. *Electronics Letters*, 2015, 51(6): 486 – 488.

[12] M. S. Arulampalam, S. Maskell, N. Gordon, et al. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. on Signal Processing*, 2002, 50(2): 174 – 188.

[13] Y. Wu, J. Wang, X. Lv, et al. Modified sequential importance resampling filter. *Journal of Systems Engineering and Electronics*, 2015, 26(3): 441 – 449.

[14] J. H. Kotecha, P. M. Djuric. Gaussian particle filtering. *IEEE Trans. on Signal Processing*, 2003, 51(10): 2592 – 2601.

[15] Y. Yang, Y. Liang, Q. Pan, et al. Gaussian-consensus filter for nonlinear systems with randomly delayed measurements in sensor networks. *Information Fusion*, 2016, 30: 91 – 102.

[16] Y. Huang, Y. Zhang, N. Li. Latency probability estimation of non-linear systems with one-step randomly delayed measurements. *IET Control Theory & Applications*, 2016, 10(7): 843 – 852.

[17] X. Wang, Y. Liang, Q. Pan, et al. Measurement random latency probability identification. *IEEE Trans. on Automatic Control*, 2016, 61(12): 4210 – 4216.

[18] Y. Zhang, Y. Huang, N. Li, et al. Particle filter with one-step randomly delayed measurements and unknown latency probability. *International Journal of Systems Science*, 2016, 47(1): 209 – 221.

[19] J. Liu, C. Z. Han, V. Prahlad. Process noise identification based particle filter: an efficient method to track highly manoeuvring targets. *IET Signal Processing*, 2011, 5(6): 538 – 546.

Biographies

Junyi Zuo was born in 1975. He received his B.S. and M.S. degrees from the School of Aeronautics, Northwestern Polytechnical University, in 2000 and 2003 respectively, and Ph.D. degree from the School of Automation, Northwestern Polytechnical University in 2008. Currently, he is an associated professor at the School of Aeronautics, Northwestern Polytechnical University. His research interests include optimal estimation theory and its application, nonlinear filtering, target tracking, and aircraft system identification.
E-mail: zuojunyi@163.com

Xiaoping Zhong was born in 1977. He received his B.S. degree from the School of Transportation, Wuhan University of Technology in 2000, M.S. and Ph.D. degrees from the School of Aeronautics, Northwestern Polytechnical University in 2003 and 2007 respectively. Currently, he is a lecturer at the School of Aeronautics, Northwestern Polytechnical University. His research interests are optimization theory and its application, and flight control technique.
E-mail: zxp_proteus@nwpu.edu.cn