

Article

# Particle Filter for Randomly Delayed Measurements with Unknown Latency Probability

Ranjeet Kumar Tiwari <sup>1,\*</sup> , Shovan Bhaumik <sup>1</sup>, Paresh Date <sup>2</sup> and Thiagalingam Kirubarajan <sup>3</sup>

<sup>1</sup> Department of Electrical Engineering, Indian Institute of Technology Patna, Patna 801106, India; shovan.bhaumik@iitp.ac.in

<sup>2</sup> Department of Mathematics, Brunel University London, Uxbridge UB83PH, UK; paresh.date@brunel.ac.uk

<sup>3</sup> Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S4L8, Canada; kiruba@mcmaster.ca

\* Correspondence: ranjeet.pee16@iitp.ac.in

Received: 19 August 2020; Accepted: 15 September 2020; Published: 6 October 2020



**Abstract:** This paper focuses on developing a particle filter based solution for randomly delayed measurements with an unknown latency probability. A generalized measurement model that includes measurements randomly delayed by an arbitrary but fixed maximum number of time steps along with random packet drops is proposed. Owing to random delays and packet drops in receiving the measurements, the measurement noise sequence becomes correlated. A model for the modified noise is formulated and subsequently its probability density function (pdf) is derived. The recursion equation for the importance weights is developed using pdf of the modified measurement noise in the presence of random delays. Offline and online algorithms for identification of the unknown latency parameter using the maximum likelihood criterion are proposed. Further, this work explores the conditions that ensure the convergence of the proposed particle filter. Finally, three numerical examples, one with a non-stationary growth model and two others with target tracking, are simulated to show the effectiveness and the superiority of the proposed filter over the state-of-the-art.

**Keywords:** nonlinear estimation; particle filter; randomly delayed measurements; latency probability

## 1. Introduction

State estimation for nonlinear discrete-time stochastic systems has received considerable attention from researchers because of its application in various fields of science, including navigation and localization [1,2], surveillance [3], agriculture [4], econometrics [5], and meteorology [6], for example. The Bayesian approach [7] gives a recursive relationship for the computation of the posterior probability density functions (pdf) of the unobserved states. But the computation of the posterior pdf in case of a nonlinear system is often numerically intractable, and hence suboptimal approximations of these pdf are often used. The particle filter (PF) is a powerful sequential Monte Carlo method under the Bayesian framework to solve nonlinear and non-Gaussian estimation problems by approximating the posterior pdf empirically [8]. The particle filter often outperforms other approximate Bayesian filters such as the extended Kalman filter (EKF) and the grid-based filters in solving nonlinear state estimation problems [9]. However, most works on the EKF [10] and as well as on the traditional PF [8,9,11] typically assume that measurements are available at each time step without any delay. In practice, in many aerospace and underwater target tracking [12], control [13] and communication [14] subsystems, random delays in receiving the measurements are inevitable. These delays, usually caused by the limitations of the network channel, need to be accounted for while designing the state estimator.

In the literature, the random delays have been addressed in the context of linear estimators [15–20]. A linear networked estimator is proposed in Reference [21] to tackle irregularly-spaced and delayed

measurements in a multisensor environment. On the other hand, the research on random delays and packet drops in nonlinear state estimation is limited. In Reference [22] and Reference [23], improved versions of the EKF and the unscented Kalman filter (UKF) are proposed for one-time step and two-time step randomly delayed measurements. In Reference [24], quadrature filters have been modified to solve nonlinear filtering problem with one-step randomly delayed measurements. In Reference [25], the cubature Kalman filter (CKF) [26] is used to tackle one-step randomly delayed measurements for nonlinear systems. In Reference [27], a methodology to solve nonlinear estimation problems with multi-step randomly delayed measurements is proposed. However, all these non-linear filters are restricted to Gaussian approximations. Moreover, they assume that the latency probability of delayed measurements is known. In Reference [28] and Reference [29], a modified PF that deals with one-step randomly delayed measurement with unknown latency probability and a PF for multi-step randomly delayed measurements with a known latency probability are presented, respectively. In References [30,31], the estimation of the unknown latency parameter with one-step randomly delayed measurements is addressed using data log-likelihood function within the Expectation-Maximization (EM) framework. However, none of these works considered the presence of random packet drops. Further,  $H_\infty$  filtering techniques are used to tackle network-induced delays and packet drops in References [32–34].

The specific contributions of this paper to the state-of-the art, specifically over References [28–30] are as follows: (i) We propose a new PF with an explicit expression for the importance weight for randomly delayed measurements of any number of time steps with an unknown latency probability and in the presence of packet drops. In Reference [29], the design of a PF with multi-step randomly delayed measurements is addressed. However, the latency probability is assumed to be known and packet drop is not considered while deriving the expression for the importance weight. (ii) The latency parameter for the random delays and packet drops is assumed to be unknown and we present a method to estimate it both in offline and online manners by maximizing the likelihood function. The sequential Monte Carlo (SMC) method is used here to approximate the likelihood function in the presence of randomly delayed measurements of any number of time steps and packet drops. In References [28,30,31] the latency probability for the measurements with random delays of maximum one step is estimated without considering packet drops. Moreover, while Gaussian approximation is used in the E-steps of the EM framework in References [30,31], the SMC approximation is used in Reference [28], but only for measurements with random delays of maximum one time step and without considering any packet drops. (iii) Due to presence of random delays and packet drops, the measurement noise sequence at different time steps becomes correlated. This work first formulates the modified noise model and derives a general pdf for the modified noise. The proposed PF is then developed using the pdf of the modified measurement noise. In References [25,35], randomly delayed measurements with the correlated measurement noise for a nonlinear system is addressed. However, they have considered a maximum delay of one time step and used the Gaussian approximation to develop the filtering algorithm.

Finally, with the help of three numerical examples, the effectiveness and the superiority of the proposed PF are demonstrated in comparison with the state-of-the-art algorithms. Table 1 lists the features of the previous works and of the proposed work.

**Table 1.** Features comparison for proposed work and previous works.

Work	Random Delays	Packet Drops	Latency Estimation	Filtering
[28]	Single step	×	✓	SMC
[29]	Multi-step	×	×	SMC
[30]	Single-step	×	✓	Gaussian
[31]	Single-step	×	✓	Gaussian
Proposed work	Multi-step	✓	✓	SMC

The rest of this paper is organized as follows. The problem statement is defined in Section 2. In Section 3, the modified PF is proposed and its convergence is discussed. Section 4 deals with the estimation of the unknown latency probability. In Section 5, simulation results are presented to demonstrate the superiority of the proposed PF. Finally, in Section 6, some conclusions are discussed.

## 2. Problem Statement

Consider a nonlinear dynamic system that can be described by the following equations:

$$\text{State equation: } x_k = f_{k-1}(x_{k-1}, k-1) + q_{k-1}, \quad (1)$$

$$\text{Measurement equation: } z_k = h_k(x_k, k) + v_k, \quad (2)$$

where  $x_k \in \mathfrak{R}^{n_x}$  denotes the state vector of the system and  $z_k \in \mathfrak{R}^{n_z}$  is the measurement at any discrete time  $k \in (0, 1, \dots)$ , while  $q_{k-1} \in \mathfrak{R}^{n_x}$  and  $v_k \in \mathfrak{R}^{n_z}$  are mutually independent white noises with arbitrary but known pdf. Here, we consider the case where actual measurement received at a particular time step may be a randomly delayed measurement from a previous time step. This delay (in number of integer time steps) can be between 0 and  $N$  at the  $k$ th sampling instant. If any measurement gets delayed by more than  $N$  steps, that measurement is discarded and no measurement is received at the estimator. Here,  $N$  is the maximum (in number of integer time steps) delay that is determined as discussed in Sections 3.2 and 3.3.

To model delayed measurements at the  $k$ th instant, we choose the independent and identically distributed Bernoulli random numbers  $\beta_k^i$  ( $i = 1, 2, \dots, N+1$ ) that take values either 0 or 1 with an unknown probability  $P(\beta_k^i = 1) = p = E[\beta_k^i]$  and  $P(\beta_k^i = 0) = 1 - p$ , where  $p$  is the unknown latency parameter. If  $y_k$  is the measurement received at the  $k$ th instant [27], then

$$\begin{aligned} y_k &= (1 - \beta_k^1)z_k + \beta_k^1(1 - \beta_k^2)z_{k-1} + \beta_k^1\beta_k^2(1 - \beta_k^3)z_{k-2} + \dots + \prod_{i=1}^N \beta_k^i(1 - \beta_k^{N+1})z_{k-N} + \left(1 - (1 - \beta_k^1) \right. \\ &\quad \left. - \beta_k^1(1 - \beta_k^2) - \dots - \prod_{i=1}^N \beta_k^i(1 - \beta_k^{N+1})\right) y_{k-1}, \quad (3) \\ &= \sum_{j=0}^N \alpha_k^j z_{k-j} + \left(1 - \sum_{j=0}^N \alpha_k^j\right) y_{k-1}; \quad k \geq 2, \end{aligned}$$

where

$$\alpha_k^j = \prod_{i=0}^j \beta_k^i (1 - \beta_k^{j+1}). \quad (4)$$

Here,  $\beta_k^0$  is considered to be 1. A measurement received at the  $k$ th time instant is  $j$  step delayed if  $\alpha_k^j = 1$ . Additionally, at time instant  $k$ , at most one of  $\alpha_k^j$  ( $0 \leq j \leq N$ ) can be 1. If all  $\alpha_k^j$  are zeros, the estimator buffer keeps the measurement  $y_{k-1}$  received at the previous step. This results in the loss of a measurement (packet drop) when that measurement is delayed by more than  $N$  steps due to buffer-size limitation.

**Remark 1.** Bernoulli random variable  $\beta_k^i$  and its function  $\alpha_k^j$  are used to represent the real-time randomness of delays in measurements in practical systems [15]. Inclusion of the possibility that at a time step  $k$  no measurement may be received (when the delay is longer than  $N$  steps) corresponds to the practical limit on the buffer size in real systems. In our algorithm, if a received measurement matches with any of the measurements in the buffer, it is discarded.

**Remark 2.** The latency parameter of received measurements,  $p$ , that is, the mean of random variable  $\beta_k^i$  is unknown in real systems. Contrary to the assumption of single-step delay in Reference [28], the unknown latency probability in this paper is for arbitrary step delays along with the packet drops due to buffer limitation.

Further, since both the ideal measurement and the measurement noise are modified in (3),  $y_k$  needs to be rewritten for its subsequent use in characterizing the densities. Hereafter in this

article,  $h_k(x_k, k)$  and  $f_k(x_k, k)$  will be written as  $h_k(x_k)$  and  $f_k(x_k)$  respectively, for brevity. Then, (3) can be restructured as

$$y_k = G1_k(h_{k-j}(x_{k-j}), \alpha_k^j) + G2_k(G1_{k-1}(\cdot), \alpha_k^j, G2_{k-1}(\cdot)) + v'_k; \quad k \geq 2 \quad \text{and} \quad 0 \leq j \leq N. \quad (5)$$

Here,  $G1_k(h_{k-j}(x_{k-j}), \alpha_k^j)$  and  $G2_k(G1_{k-1}(\cdot), \alpha_k^j, G2_{k-1}(\cdot))$  denote the ideal measurement parts (without any noise) of  $y_k$  from the non-delayed measurements  $z_{k-N:k}$  and the previous step measurement  $y_{k-1}$ , respectively, and are defined as

$$G1_k(h_{k-j}(x_{k-j}), \alpha_k^j) = \sum_{j=0}^N \alpha_k^j h_{k-j}(x_{k-j}); \quad k \geq 2 \quad (6)$$

$$G2_k(G1_{k-1}(\cdot), \alpha_k^j, G2_{k-1}(\cdot)) = \left(1 - \sum_{j=0}^N \alpha_k^j\right) (G1_{k-1}(\cdot) + G2_{k-1}(\cdot)); \quad k \geq 2, \quad (7)$$

where  $G1_1(\cdot) = h_1(x_1)$  and  $G2_1(\cdot) = 0$ . Again,  $v'_k$  is the additive measurement noise in (3) defined as

$$v'_k = \sum_{j=0}^N \alpha_k^j v_{k-j} + \left(1 - \sum_{j=0}^N \alpha_k^j\right) v'_{k-1}; \quad k \geq 2, \quad (8)$$

where  $v'_{k-1}$  is the noise received along with observation  $y_{k-1}$  and  $v'_1 = v_1$ .

Now, the objective is to develop a PF algorithm for the system in (1) with measurement model (3) that assumes the knowledge of latency probability  $p$ . Additionally, we propose offline as well as online algorithms to estimate  $p$ .

### 3. Modified Particle Filter for Randomly Delayed Measurements

#### 3.1. Particle Filter

In a sequential importance sampling filter, the posterior probability density function  $P(x_{0:k}|z_{1:k})$  is replaced by its equivalent series of weighed particles, which can be represented as [9]

$$\hat{P}(x_{0:k}|z_{1:k}) = \sum_{i=1}^{N_s} w_k^i \delta[x_{0:k} - x_{0:k}^i], \quad (9)$$

where particles  $\{x_{0:k}^i\}_{i=1}^{N_s}$  are drawn from a proposal density  $q(x_{0:k}|z_{1:k})$  and then the weights of the particles are chosen using the importance principle. The unnormalized importance weight of the  $i$ th particle is given by

$$w_k^i = \frac{P(x_{0:k}^i|z_{1:k})}{q(x_{0:k}^i|z_{1:k})}. \quad (10)$$

The recursive weight update at each time step is given by

$$P(x_{0:k}|z_{1:k}) = \frac{P(z_k|x_{0:k})P(x_{0:k}|z_{1:k-1})}{P(z_k|z_{1:k-1})} \propto P(z_k|x_k)P(x_k|z_{1:k-1}), \quad (11)$$

where  $P(z_k|z_{1:k-1})$  is a normalizing constant. Similarly, the proposal density is assumed to be decomposed as

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{0:k-1}, z_{1:k})q(x_{0:k-1}|z_{1:k-1}). \quad (12)$$

Assuming that the state vector  $x_k$  follows the Markov process, the importance weight of (10), with the help of (11) and (12), can be written as

$$\begin{aligned} w_k^i &\propto \frac{P(z_k|x_k^i)P(x_k^i|x_{k-1}^i)P(x_{k-1}^i|z_{1:k-1})}{q(x_k^i|x_{0:k-1}^i, z_{1:k})q(x_{k-1}^i|z_{1:k-1})} \\ &= w_{k-1}^i \frac{P(z_k|x_k^i)P(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})}, \end{aligned}$$

where the predicted density  $P(x_k^i|x_{k-1}^i)$  and the likelihood density  $P(z_k|x_k^i)$  can be evaluated using the system model, previously estimated posterior  $P(x_{k-1}^i|z_{1:k-1})$  and the joint noise density  $P(q_{k-1}, v_k|x_{k-1}^i)$  [7].

### 3.2. Modified PF for Randomly Delayed Measurements

A recursive computation of the importance weights can be obtained for a nonlinear system with measurement model of (3). From (3), it can be seen that  $y_k$  is stochastically dependent on the non-delayed measurements  $z_{k-N:k}$  and the previous step measurement  $y_{k-1}$  and, therefore, we need to relax the standard assumption of  $P(z_k|x_{1:k}, z_{1:k-1}) = P(z_k|x_k)$  as discussed below.

**Assumption 1.** The received measurement  $y_k$ , conditionally on  $x_{k-N:k}$  and  $y_{k-1}$ , is independent of the state vectors  $x_{1:k-N-1}$  and the measurements  $y_{1:k-2}$ , that is,  $P(y_k|x_{1:k}, y_{1:k-1}) = P(y_k|x_{k-N:k}, y_{k-1})$ .

**Lemma 1.** Recursion equation of the importance weight  $w_k^i$  for model (1) and (3), can be obtained as

$$w_k^i = w_{k-1}^i \frac{P(y_k|x_{k-N:k}^i, y_{k-1})P(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, y_{1:k})}, \quad (13)$$

where  $x_k^i$  is drawn from the proposal density  $q(x_k|x_{0:k-1}^i, y_{1:k})$ .

**Proof.** Using the Bayes' theorem and assuming that the states do not depend on the future measurements, the proposal density can be decomposed as

$$q(x_{0:k}|y_{1:k}) = q(x_k|x_{1:k-1}, y_{1:k})q(x_{1:k-1}|y_{1:k-1}). \quad (14)$$

Particles  $x_k^i$  and  $x_{1:k-1}^i$  can be sampled from  $q(x_k|x_{1:k-1}, y_{1:k})$  and  $q(x_{1:k-1}|y_{1:k-1})$ , respectively. Again, using the Bayes' rule, the joint pdf,  $P(x_{1:k}, y_{1:k})$ , can be decomposed as follows:

$$\begin{aligned} P(x_{1:k}, y_{1:k}) &= P(y_k|x_k, x_{1:k-1}, y_{1:k-1})P(x_k, x_{1:k-1}, y_{1:k-1}) \\ &= P(y_k|x_k, x_{1:k-1}, y_{1:k-1})P(x_k|x_{1:k-1}, y_{1:k-1})P(x_{1:k-1}, y_{1:k-1}). \end{aligned} \quad (15)$$

By Assumption 1 and the first-order Markov property of the state vectors, (15) can be rewritten as

$$P(x_{1:k}, y_{1:k}) = P(y_k|x_{k-N:k}, y_{k-1})P(x_k|x_{k-1})P(x_{1:k-1}, y_{1:k-1}). \quad (16)$$

Using (14) and (16), the importance weight can be written as

$$\begin{aligned} w_k &= \frac{P(y_k|x_{k-N:k}, y_{k-1})P(x_k|x_{1:k-1})}{q(x_k|x_{0:k-1}, y_{1:k})} \frac{P(x_{1:k-1}, y_{1:k-1})}{q(x_{0:k-1}|y_{1:k-1})} \\ &= w_{k-1} \frac{P(y_k|x_{k-N:k}, y_{k-1})P(x_k|x_{1:k-1})}{q(x_k|x_{0:k-1}, y_{1:k})}. \end{aligned} \quad (17)$$

Now, with the help of (17),  $w_k^i$  can be finally written as (13).  $\square$

Now, the predicted density  $P(x_k^i|x_{k-1}^i)$  and the likelihood density  $P(y_k|x_{k-N:k}^i, y_{k-1})$  can be characterized using the joint noise density  $P(q_{k-1}, v'_k|x_{k-1})$  [7,36]. As given in Section 2,  $q_{k-1}$  and  $v_k$  are independent noise processes with  $E[q_{k-1-i}(v_{k-j})^\top] = E[q_{k-1-i}]E[v_{k-j}] = 0$ , for all integer values of  $i$  and  $j$ . Hence, using the independence property and (8), it can be shown that  $q_{k-1}$  and the modified measurement noise  $v'_k$  are also independent. Therefore, assuming  $q_{k-1}$  and  $v'_k$  are independent of the previous state  $x_{k-1}$ , we can decompose the joint noise density as  $P(q_{k-1}, v'_k|x_{k-1}) = P(q_{k-1})P(v'_k)$ . Moreover, given that the pdf of  $q_{k-1}$  is known,  $P(x_k^i|x_{k-1}^i)$  can be evaluated, whereas the pdf of  $v'_k$  is unknown and needs to be calculated for the evaluation of the likelihood.

Further, for the computation of the likelihood density  $P(y_k|x_{k-N:k}, y_{k-1})$ , the probability related to the number of random delays in the received measurement needs to be evaluated. Note that the probability of a received measurement being delayed by  $j$  time steps, at any instant  $k$ , is [27]  $P(\alpha_k^j = 1) = p^j(1 - p)$ ,  $0 \leq j \leq N$ . Note that as the number of delay steps ( $j$ ) increases, the associated probability ( $P(\alpha_k^j = 1)$ ) decreases. The probability that the estimator receives  $y_{k-1}$  at the  $k$ th instant of time is [27]  $P(\sum_{j=0}^N \alpha_k^j = 0) = p^{N+1}$ . It can be observed that for a high value of  $p$ ,  $N$  should be kept sufficiently large to reduce the probability of a packet being lost.

**Lemma 2.** *The likelihood density  $P(y_k|x_{k-N:k}^i, y_{k-1})$  can be computed recursively as*

$$P(y_k|x_{k-N:k}^i, y_{k-1}) = \sum_{j=0}^N p^j(1 - p)P_{v_{k-j}}(y_k - h_{k-j}(x_{k-j}^i)) + p^{N+1}P(y_{k-1}|x_{k-1-N:k-1}^i, y_{k-2}), \quad (18)$$

where  $P_{v_{k-j}}(\cdot)$  represents the pdf of the measurement noise  $v_{k-j}$ .

**Proof.** Let  $\gamma_k$  be a Bernoulli random variable that denotes the event that a measurement is received (with a step delay between 0 and  $N$ ). The probability that a measurement is received with a delay less than or equal to  $N$  steps, is

$$P(\gamma_k = 1) = P\left(\sum_{j=0}^N \alpha_k^j = 1\right) = \sum_{j=0}^N p^j(1 - p). \quad (19)$$

The probability that no measurement is received and the estimator keeps measurement  $y_{k-1}$ , is

$$P(\gamma_k = 0) = P\left(\sum_{j=0}^N \alpha_k^j = 0\right) = p^{N+1}. \quad (20)$$

Now, as the modified likelihood density should be characterized by the pdf of the modified measurement noise  $v'_k$ , using (5), we have

$$P(y_k|x_{k-N:k}, y_{k-1}) = P_{v'_k}(y_k - G1_k(h_{k-j}(x_{k-j}^i), \alpha_k^j) - G2_k(G1_{k-1}(\cdot), \alpha_k^j, G2_{k-1}(\cdot))), \quad (21)$$

where  $P_{v'_k}(\cdot)$  denotes the pdf of  $v'_k$ . Further, assuming that  $v'_k$ , conditionally on  $v_{k-N:k}$  and  $v'_{k-1}$ , is an independent noise sequence over time, it can be calculated using the known pdf of  $v_{k-N:k}$  and the intermediate Bernoulli random variable  $\gamma_k$  as follows:

$$\begin{aligned}
P(v'_k) &= \int P(v'_k, \gamma_k) d\gamma_k \\
&= \sum_{\gamma_k=0}^1 P(v'_k | \gamma_k) P(\gamma_k) \\
&= P(v'_k | \gamma_k = 1) P(\gamma_k = 1) + P(v'_k | \gamma_k = 0) P(\gamma_k = 0) \\
&\text{(using (19) and (20))} \\
&= \sum_{j=0}^N p^j (1-p) \times P(v'_k | \gamma_k = 1) + p^{N+1} \times P(v'_k | \gamma_k = 0).
\end{aligned} \tag{22}$$

Using the expression for  $v'_k$  in (8), if  $\gamma_k = 1$  (i.e.,  $\alpha_k^j = 1$  for any  $j$ ), we can write  $v'_k = v_{k-j}$  and

$$P(v'_k | \gamma_k = 1) = P(v_{k-j}). \tag{23}$$

Similarly, when  $\gamma_k = 0$  (i.e.,  $\sum_{j=0}^N \alpha_k^j = 0$ ), the measurement noise  $v'_k = v'_{k-1}$  and we have

$$P(v'_k | \gamma_k = 0) = P(v'_{k-1}). \tag{24}$$

Substituting (23) and (24) into (22) and then, (22) and (21) results in

$$\begin{aligned}
P(y_k | x_{k-N:k}, y_{k-1}) &= \sum_{j=0}^N p^j (1-p) P_{v_{k-j}}((y_k - G1_k(h_{k-j}(x_{k-j}), \alpha_k^j) - G2_k(G1_{k-1}(\cdot), \alpha_k^j, G2_{k-1}(\cdot))) |_{\gamma_k=1}) \\
&\quad + p^{N+1} P_{v'_{k-1}}((y_k - G1_k(h_{k-j}(x_{k-j}), \alpha_k^j) - G2_k(G1_{k-1}(\cdot), \alpha_k^j, G2_{k-1}(\cdot))) |_{\gamma_k=0}).
\end{aligned} \tag{25}$$

By definition,  $\gamma_k = \sum_{j=0}^N \alpha_k^j$ . Again, using (6), we can write  $(y_k - G1_k(\cdot) - G2_k(\cdot)) |_{\gamma_k=1} = y_k - h_{k-j}(x_{k-j})$ ,  $0 \leq j \leq N$ . Similarly, using (3) and (7), we have  $(y_k - G1_k(\cdot) - G2_k(\cdot)) |_{\gamma_k=0} = y_{k-1} - G1_{k-1}(\cdot) - G2_{k-1}(\cdot)$ . Now, rewriting (25) for particle  $i$ , we have

$$\begin{aligned}
P(y_k | x_{k-N:k}^i, y_{k-1}) &= \sum_{j=0}^N p^j (1-p) P_{v_{k-j}}(y_k - h_{k-j}(x_{k-j}^i)) + p^{N+1} P_{v'_{k-1}}(y_{k-1} - G1_{k-1}(\cdot) - G2_{k-1}(\cdot)), \\
&\text{(applying (21) for } k := k-1 \text{ to the second term on the right hand side)} \\
&= \sum_{j=0}^N p^j (1-p) P_{v_{k-j}}(y_k - h_{k-j}(x_{k-j}^i)) + p^{N+1} P(y_{k-1} | x_{k-1-N:k-1}^i, y_{k-2}).
\end{aligned}$$

□

**Remark 3.** Note that (18) is similar to the sum of the product densities used in the probabilistic data association (PDA) algorithm [37,38]. Further, it can be observed that if there are no random delays and packet drops in the received measurements (i.e.,  $N = 0$  and  $p = 0$ ), (18) reduces to  $P_{v_k}(z_k - h_k(x_k^i))$ , the expression for the likelihood density of the standard PF.

In this work, to mitigate the effect of degeneracy on the importance weights of particles, the resampling is carried out at each step. A general proposal density  $q(x_k | x_{0:k-1}, y_{1:k})$  obtained from a nonlinear filter such as the EKF or the UKF [39], can be used to draw samples. However, the predicted density  $P(x_k | x_{k-1})$  is a common choice to implement the sequential importance resampling (SIR) PF [9] despite the fact that the current measurement is not used to locate new samples. Note that the current measurement in this work is randomly delayed and is unaccounted for in the proposal density, which may not necessarily push the sampled particles towards higher likelihood regions. On the other hand,  $P(x_k | x_{k-1})$  includes the measurements up to the last time-step that have been corrected with the modified importance weight. The steps to implement a standard SIR filter [9] with the modified

importance weight for this work are outlined in Algorithm 1. It uses a standard resampling technique, for example, multinomial [40] or, systematic [9] to suppress the particles with negligible weight. A flow diagram to obtain the modified PF estimate is shown in Figure 1.

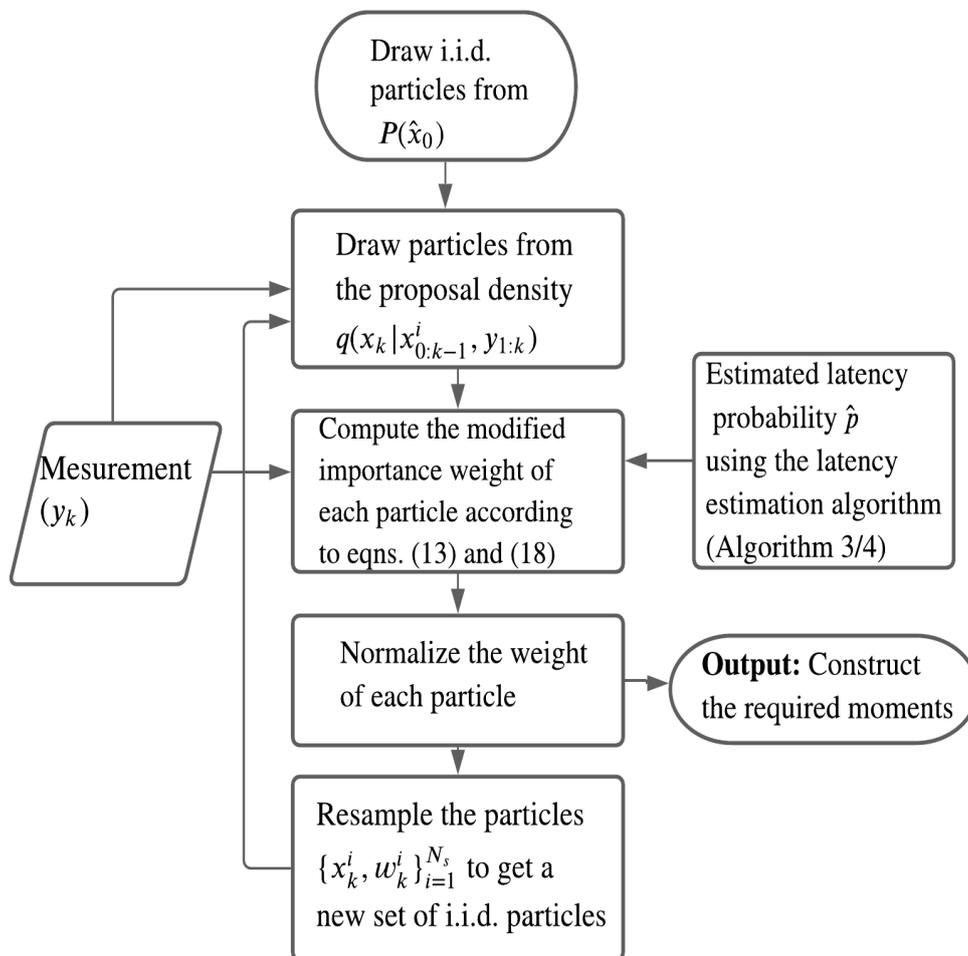
---

**Algorithm 1** Modified particle filter.

---

$$[\{x_k^i, w_k^i\}_{i=1}^{N_s}] := \text{SIR}[\{x_{k-N:k}^i, w_{k-1}^i\}_{i=1}^{N_s}, \hat{p}, y_k]$$

- for  $i = 1 : N_s$ 
    - Draw  $x_k^i \sim q(x_k | x_{0:k-1}^i, y_{1:k})$
    - Compute the importance weight  $w_k^i$  according to (13) and (18)
    - Normalize the importance weight:  $w_k^i := w_k^i / \text{SUM}[\{w_k^i\}_{i=1}^{N_s}]$
  - end for
  - Resample the particles at each step
    - $[\{x_k^i, w_k^i\}_{i=1}^{N_s}] := \text{RESAMPLE}[\{x_k^i, w_k^i\}_{i=1}^{N_s}]$
- 



**Figure 1.** Flow diagram for PF with the modified importance weight.

### 3.3. Convergence of the PF for Randomly Delayed Measurements

In this subsection, we explore the conditions for the convergence of the modified PF derived for randomly delayed measurements. A PF is said to be converging if its empirical approximation

follows a mean square error of order  $1/N_s$  at each step of filtering [8]. The prime requisite for simple convergence is that likelihood function  $P(z_k|\cdot)$  should be upper bounded, that is,  $\|P(z_k|\cdot)\| < \infty$ , for all  $x_k \in \mathfrak{R}^{n_x}$  [8]. The following lemma is an extension of the results in Reference [8] for our case.

**Lemma 3.** *If the likelihood function*

$$\|P(y_k|x_{k-N:k}, y_{k-1})\| < \infty, \quad \forall x_{k-N:k} \in \mathfrak{R}^{n_x} \text{ and } y_k \in \mathfrak{R}^{n_z}, \quad (26)$$

then,  $\forall k \geq N$ , there exists  $c_{k/k}$  independent of  $N_s$ , such that for any  $\Phi \in B(\mathfrak{R}^{n_x \times (k+1)})$

$$E \left[ \left( \frac{1}{N_s} \sum_{i=1}^{N_s} \Phi(x_{0:k}^i) - \int \Phi(x_{0:k}) P(dx_{0:k}|y_{1:k}) \right)^2 \right] \leq c_{k/k} \frac{\|\Phi\|^2}{N_s}, \quad (27)$$

where  $x_{0:k}^i$  are the unweighted particles obtained using the modified PF algorithm.

**Proof.** Attributing to the random delays and packet drops in measurements, we investigate the impact of the modified likelihood density on the simple convergence of the particle filter that is otherwise converging with non-delayed measurements. That is,  $\|P(z_k|x_k)\| < \infty$ ,  $\forall x_k \in \mathfrak{R}^{n_x}$  and  $z_k \in \mathfrak{R}^{n_z}$ , is given. Now, using (2), we can write

$$P(z_k|x_k) = P_{v_k}(z_k - h_k(x_k)).$$

Thus, the pdf of noise  $v_k$ , that is,  $P_{v_k}(z_k - h_k(x_k))$ , is bounded for all its real-valued inputs. Rearranging the terms of (18) on the both sides, we have

$$P(y_k|x_{k-N:k}, y_{k-1}) - p^{N+1}P(y_{k-1}|x_{k-1-N:k-1}) = \sum_{j=0}^N p^j(1-p)P_{v_{k-j}}(y_k - h_{k-j}(x_{k-j})). \quad (28)$$

As  $v_k$  is a stationary process, its pdf is not affected by time shift, that is, if  $P_{v_k}(\cdot)$  is bounded,  $P_{v_{k-j}}(\cdot)$  must also be bounded. Again, since  $p^j(1-p) < 1$  for all values of  $j$ , we can write

$$\sum_{j=0}^N p^j(1-p)P_{v_{k-j}}(y_k - h_{k-j}(x_{k-j})) \leq (N+1)P_{v_{k-j}}(y_k - h_{k-j}(x_{k-j})). \quad (29)$$

Given that  $N$  is a finite number, (28) and (29) can be used to establish (26). Now, results of Reference [8] can be used to verify (27).  $\square$

Theorem 3 of Reference [8] suggests that  $c_{k/k}$  is independent of the number of particles  $N_s$ , but represents the dependency of mixed dynamics of the system on the initial conditions or past values. That is, if the optimal filter associated with the system dynamics has a long memory,  $c_{k/k}$  will continue to increase the mean square error with each step. In our case, the filter does have a memory due to arbitrary delays in measurements. Therefore, to avoid the large mean square error of convergence, the value of  $N$  should be kept small. On the other hand, to counter the increase in the value of  $c_{k/k}$ , number of particles  $N_s$  needs to be large.

#### 4. Identification of Latency Probability

In practice, when a set of randomly delayed measurements is given, we may not know channel properties and its random parameters. Hence, the latency probability that is required to design the filter may be unknown. Here, we use the ML criterion to identify the unknown latency probability for

received measurements. This method involves the maximization of the joint density  $P_p(y_{1:m})$  of the received measurements, which is a function of latency parameter  $p$  represented by [28]

$$\hat{p} = \arg \max_{p \in [0,1]} P_p(y_1, \dots, y_m), \quad (30)$$

where  $m$  is the number of measurements used for the identification of parameter  $p$  and  $\hat{p}$  is the estimated value of  $p$ . Now, we assume that the first received measurement  $y_1$  is independent of parameter  $p$  and is equal to  $z_1$ . Again, using the Bayes' theorem the above joint pdf can be reformulated as

$$P_p(y_1, \dots, y_m) = P(y_1) \prod_{k=2}^m P_p(y_k | y_{1:k-1}). \quad (31)$$

For computational simplicity the above maximization of likelihood is expressed in terms of the log-likelihood (LL). The LL of (31) can be formulated as

$$\begin{aligned} L_p(y_{1:m}) &= \log P_p(y_{1:m}) \\ &= \log P(y_1) + \sum_{k=2}^m \log P_p(y_k | y_{1:k-1}), \end{aligned} \quad (32)$$

where  $L_p(y_{1:m})$  is the LL function of the received measurements. Now, to solve the maximization problem of (31), first the computation of likelihood  $P_p(y_k | y_{1:k-1})$  and then the maximization of  $L_p(y_{1:m})$  need to be carried out.

#### 4.1. Computation of Likelihood Density

State likelihood can be used to compute  $P_p(y_k | y_{1:k-1})$  with the help of the SMC approximation [41]. We can express the likelihood density  $P_p(y_k | y_{1:k-1})$  as the marginal density of a joint pdf that includes delayed measurement and previous states that are correlated using the Bayes' theorem and Assumption 1 as follows:

$$\begin{aligned} P_p(y_k | y_{1:k-1}) &= \int \dots \int P_p(y_k, x_{k-N:k} | y_{1:k-1}) dx_k \dots dx_{k-N} \\ &= \int \dots \int P_p(y_k | x_{k-N:k}, y_{1:k-1}) P_p(x_{k-N:k} | y_{1:k-1}) dx_k \dots dx_{k-N} \\ &= \int \dots \int P_p(y_k | x_{k-N:k}, y_{k-1}) P_p(x_{k-N:k} | y_{1:k-1}) dx_k \dots dx_{k-N}. \end{aligned} \quad (33)$$

Using Bayes' rule, the joint state prior  $P_p(x_{k-N:k} | y_{1:k-1})$  can be decomposed as

$$P_p(x_{k-N:k} | y_{1:k-1}) = P_p(x_k | x_{k-N:k-1}, y_{1:k-1}) P_p(x_{k-N:k-1} | y_{1:k-1})$$

Since the state vectors follow the first-order Markov property, using the chain rule and (9), we can write the joint pdf as

$$\begin{aligned} P_p(x_{k-N:k} | y_{1:k-1}) &= P(x_k | x_{k-1}) P_p(x_{k-1} | x_{k-2}, y_{k-1}) \dots P_p(x_{k-N} | x_{k-N-1}, y_{k-N}), \\ &\approx \frac{1}{N_s} \sum_{i=1}^{N_s} \delta[x_k - x_k^i] \delta[x_{k-1} - x_{k-1}^i] \dots \delta[x_{k-N} - x_{k-N}^i], \end{aligned} \quad (34)$$

where  $x_{k-N}^i, x_{k-N+1}^i, \dots$ , and  $x_k^i$  are the unweighted particles and drawn from  $P_p(x_{k-N} | x_{k-N-1}, y_{k-N})$ ,  $P_p(x_{k-N+1} | x_{k-N}, y_{k-N+1})$ ,  $\dots$ , and  $P(x_k | x_{k-1})$ , respectively. The accuracy of the SMC approximation in this case depends on the choice of the proposal density, the number of

particles and the value of  $N$ . This would be equivalent to that of the modified SIR as discussed in Section 3.3. Now, substituting (34) into (33),  $P_p(y_k|y_{1:k-1})$  can be computed as

$$\hat{P}_p(y_k|y_{1:k-1}) = \frac{1}{N_s} \sum_{i=1}^{N_s} P_p(y_k|x_k^i, \dots, x_{k-N}^i, y_{k-1}), \quad (35)$$

where  $P_p(y_k|x_{k-N:k}^i, y_{k-1})$  is given in (18). Algorithm 2 illustrates the steps to compute the LL function.

---

**Algorithm 2** Computation of log-likelihood function.

---

$$[L_p, \{x_k^i, w_{p,k}^i\}_{i=1}^{N_s}] := \text{LL}[L_p, \{x_{k-N:k-1}^i, w_{p,k-1}^i\}_{i=1}^{N_s}, y_k]$$

- for ( $i = 1 : N_s$ )

- Draw  $x_k^i \sim q(x_k|x_{0:k-1}^i, y_{1:k})$
- Assign particle a weight:

$$w_{p,k}^i = w_{p,k-1}^i \frac{P_p(y_k|x_{k-N:k}^i, y_{k-1})P(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, y_{1:k})}$$

- end for

- Compute the LL function:

$$L_p := L_p + \log \left( \frac{1}{N_s} \sum_{i=1}^{N_s} P_p(y_k|x_{k-N:k}^i, y_{k-1}) \right)$$

- for ( $i = 1 : N_s$ )

- Normalize the importance weight:  $w_{p,k}^i := w_{p,k}^i / \text{SUM}[\{w_{p,k}^i\}_{i=1}^{N_s}]$

- end for

- Resample the drawn particles at each step:

$$[\{x_k^j, w_{p,k}^j\}_{j=1}^{N_s}] := \text{RESAMPLE}[\{x_k^i, w_{p,k}^i\}_{i=1}^{N_s}]$$


---

#### 4.2. Maximization of Log-Likelihood Function

Substituting (35) into (32), we can rewrite the LL function in (32) as

$$\hat{L}_p(y_{1:m}) = \sum_{k=2}^m \log \left( \frac{1}{N_s} \sum_{i=1}^{N_s} P_p(y_k|x_{k-N:k}^i, y_{k-1}) \right), \quad (36)$$

where  $y_1$  is independent of parameter  $p$  and can be neglected in the maximization of the likelihood density. Equation (36) can be maximized numerically over  $p \in [0, 1]$ .

There are two options for the numerical search of the latency parameter: offline and online identification. In the offline method, we can use more measurements for higher parameter estimation accuracy. A higher value of  $m$  and smaller steps ( $sl$ ) for  $p$  can yield a more accurate estimate of the latency probability, at the expense of higher computational burden. The offline algorithm can only be started after the first  $m$  measurements and it can be repeated with further measurements to improve the estimate. Algorithm 3 outlines the steps for offline identification.

In case of online identification, the latency probability is estimated at each time step and the estimated value of the parameter is used in the proposed PF algorithm. The running average of the estimated values at each step can be evaluated to improve the accuracy of the identified parameter. Algorithm 4 outlines the online method.

**Algorithm 3** Offline identification of latency probability.

$$[L_{max}, \hat{p}] := \text{OFFLINE}[\{x_{k-N:k}^i, w_{k-1}^i\}_{i=1}^{N_s}, y_{1:m}]$$

- Select the values for  $sl$  and  $m$
- for ( $p = 0 : sl : 1$ )
  - Set  $L_p = 0$
  - for ( $k = 1 : m$ )
    - $[L_p, \{x_k^i, w_{p,k}^i\}_{i=1}^{N_s}] := \text{LL}[L_p, \{x_{k-N:k}^i, w_{p,k-1}^i\}_{i=1}^{N_s}, y_k]$
  - end for
  - Initialization: if  $p = 0$ 
    - $L_{max} = L_p$  and  $\hat{p} = p$
  - Update: else
    - If  $L_p > L_{max}$ 
      - $L_{max} = L_p$  and  $\hat{p} = p$
- end for

**Algorithm 4** Online identification of latency probability.

$$[L_{max}, \hat{p}] := \text{ONLINE}[\{x_{k-N:k}^i, w_{k-1}^i\}_{i=1}^{N_s}, y_{1:k}]$$

- Select the value for  $sl$
- for ( $p = 0 : sl : 1$ )
  - Set  $L_p = 0$
  - for ( $t = 1 : k$ )
    - $[L_p, \{x_t^i, w_{p,t}^i\}_{i=1}^{N_s}] := \text{LL}[L_p, \{x_{t-N:t}^i, w_{p,t-1}^i\}_{i=1}^{N_s}, y_t]$
  - end for
  - Initialization: if  $p = 0$ 
    - $L_{max} = L_p$  and  $\hat{p} = p$
  - Update: else
    - If  $L_p > L_{max}$ 
      - $L_{max} = L_p$  and  $\hat{p} = p$
- end for

### 4.3. Computational Complexity

Attributing to the use of numerical search method to maximize the likelihood, the estimation of latency parameter is a computationally involved process. However, once the latency parameter is estimated, the computational cost of the proposed PF algorithm is not substantially higher than that of the standard PF. In the offline mode of identification, the proposed particle filter will run  $1/sl$  times slower than the standard PF, where  $sl$  ( $0 < sl < 1$ ) is the step length used for the search algorithm. That is, if the standard PF is  $\mathcal{O}(N_s n_x^2)$  [42], then proposed PF in offline mode is  $\mathcal{O}(N_s n_x^2 / sl)$ . Similarly, for online identification, the modified particle filter will be  $\mathcal{O}(N_s n_x^2 k)$ , where  $k$  is the time step.

## 5. Simulation Results

To demonstrate the superiority of the proposed PF over the standard PF and the state-of-the-art algorithms for randomly delayed measurements, three different types of filtering problems are simulated. Although any general proposal density [39] can be chosen for the implementation of the proposed PF, in our simulations, the transitional prior density  $P(x_k | x_{k-1})$  is used as the proposal density for its simplicity. The latency probability of delayed measurements is first identified by maximizing (36) over  $p \in [0, 1]$  with the help of the proposed PF algorithm. Subsequently, the estimated probability

$p$  is used to implement the proposed PF for the given problems. To ensure a fair comparison of the proposed PF with the other state-of-the-art algorithms, two sets of simulation are carried out for each problem based on the measurement model used in the chosen filtering algorithm. In the first set of simulations, the proposed PF with estimated latency probability is compared against the standard PF and the cubature Kalman filter for the randomly delayed measurements (CKF-RD) [27] by using the randomly delayed measurements with packet drops. The proposed PF with other than the true value of  $N$  is also implemented to investigate the impact of selecting a wrong value for the maximum number of delay steps. The performances of all filters are compared in terms of the root mean square error (RMSE). Note that the proposed PF with the wrong value of  $N$  and the CKF-RD are implemented with the true value of the latency probability. In the second set of simulations, the proposed PF with the estimated latency probability is compared against the PF for multi-step randomly delay measurements (PF-MD) [29] by using the set of randomly delayed measurements with *no* packet drops. To demonstrate the importance of the latency estimation, PF-MD is implemented with both the true latency probability and the incorrect latency probability. Further, recognizing the practical limitation on the buffer-length and to avoid the large convergence error as discussed in Section 3.3, the true value of the maximum step delay is taken as  $N = 2$ . However, the simulation work can be carried out with the larger value of  $N$  as shown for Problem 1. For brevity, the case with the larger  $N$  for other problems is not included in the paper.

MATLAB 2017a software is used to carry out the simulation work. No in-built subroutine for the PF is used and all the figures presented in this work are generated by running the standard instructions in MATLAB following the Algorithms 1–4. Both the truth and measurements are generated prior to filter implementation. The filtering algorithm uses the software generated measurement values for the PF to implement.

### 5.1. Problem 1

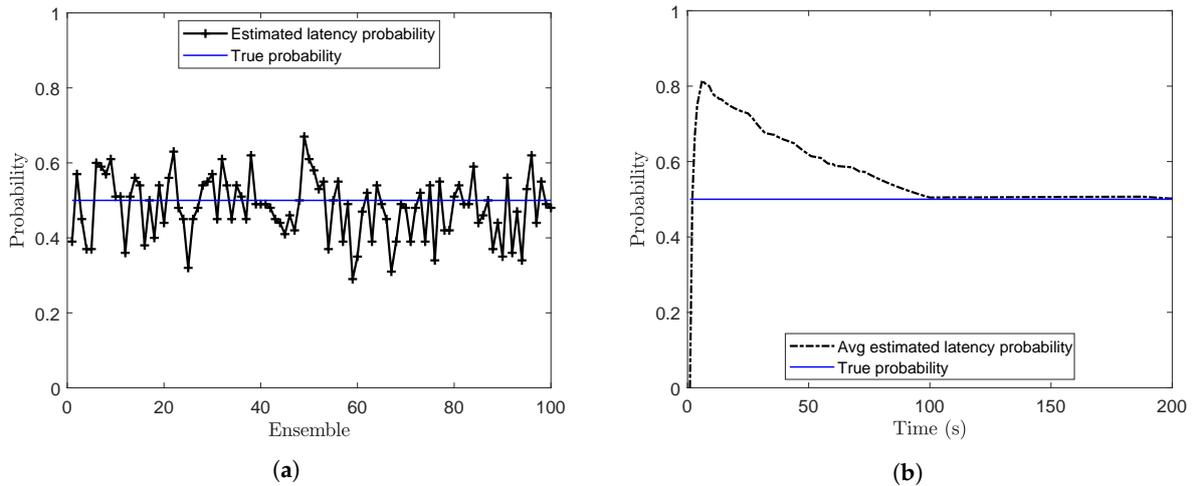
We consider a time-varying growth model that is widely used in literature because of its non-stationary property for validation of performances by nonlinear filters [9,23,43,44]. Nonlinear dynamics of the system are as follows:

$$\begin{aligned}x_k &= 0.5x_{k-1} + 25 \frac{x_{k-1}}{1 + x_{k-1}^2} + 8 \cos(1.2k) + q_{k-1}, \\z_k &= x_k^2/20 + v_k,\end{aligned}\tag{37}$$

where  $q_{k-1}$  and  $v_k$  are independent zero mean Gaussian processes with  $E[q_k^2] = 10$  and  $E[v_k^2] = 1$ , respectively. The initial state  $x_0 \sim \mathcal{N}(0, 1)$ , and the received measurements,  $y_{1:k}$  are generated by using (37) and (3) with  $N = 2$ . The distribution of the initial estimated state  $P(\hat{x}_0) \sim \mathcal{N}(0, 1)$  and the number of particles used for the simulation of this problem is,  $N_s = 1000$ .

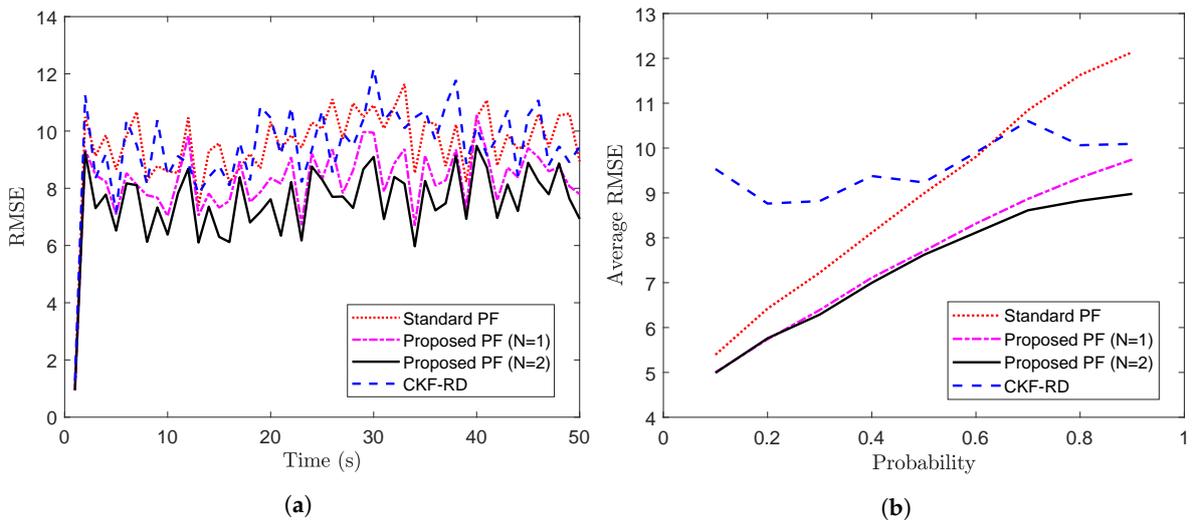
Offline estimation of the latency probability is carried out by using Algorithm 3 with  $sl = 0.01$  and  $m = 500$ . The latency probability ( $p$ ) at the end of each ensemble is calculated and plotted in Figure 2a. For this case, when the true value of  $p$  is 0.5, the mean of the estimated latency probability over 100 ensembles is calculated as 0.481. Online estimates of the latency probability are shown in Figure 2b. Here, the estimated probability at each time step is the running average of the estimated probabilities.

The proposed PF is implemented with  $N = 1$  and  $N = 2$  on the measurements as described above, and the results are compared against that of the CKF-RD and the standard PF. To compare the results, the RMSE calculated over 100 Monte Carlo (MC) runs are plotted over 50 time steps in Figure 3a. It can be seen that the proposed PF with  $N = 2$  outperforms the other three filters. Moreover, it is interesting to observe that the performance of the proposed PF with the wrong value of the maximum number of step delay ( $N = 1$ ) is closer to that of the proposed PF with  $N = 2$ .



**Figure 2.** Estimation of the latency probability (a) Offline estimation (b) Online estimation (problem 1).

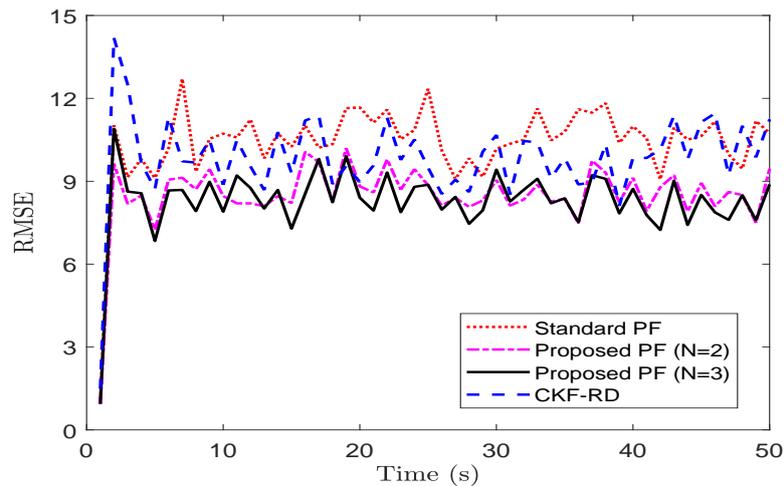
Further, the average RMSE calculated over 50 time steps for different values of true latency probability is shown in Figure 3b. As expected, for a higher probability value where packet drop is more likely, filter designed for  $N = 2$  performs better than the other filters.



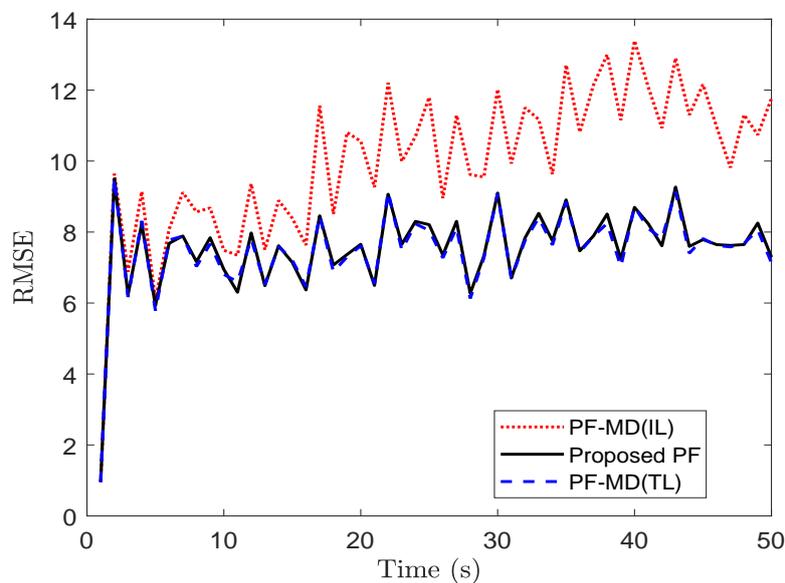
**Figure 3.** (a) RMSE vs. time steps considering  $p = 0.5$  (b) Avg RMSE vs. probability (Problem 1).

To observe the impact on the performance of the proposed filter when the maximum number of step delay is increased, we take a case of  $N = 3$ . Figure 4 shows the RMSE for the different filter while considering the true latency,  $p = 0.50$ . The average RMSE of the proposed filter in Figure 3a with  $N = 3$  is 8.79 whereas that in Figure 4 with  $N = 2$  is 7.48. The increase in the length of memory for the proposed filter explains the increase in the RMSE.

For simulation with the set of randomly delayed measurements without any packet drops, the RMSE value for PF-MD with the true latency (PF-MD(TL)), the proposed PF and PF-MD with the incorrect latency (PF-MD(IL)) are plotted in Figure 5. It can be observed that the RMSE value of the proposed PF and PF-MD(TL) remain almost the same whereas the filtering performance of PF-MD degrades when the latency probability is unknown and considered an incorrect value.



**Figure 4.** RMSE of different filters for  $N = 3$  and  $p = 0.50$  (problem 1).



**Figure 5.** RMSE of different filters for measurements with no packet drops considering  $p = 0.50$  (problem 1).

Table 2 shows the comparison of the computational burden for the different filters. To make the computational comparison independent of the software and clock speed used for simulation, we have calculated the relative computational time for each filter. All computational time is calculated with respect to the computational time of the standard PF algorithm. Note that the given computational time for the proposed PF is after the estimation of the latency parameter. However, the estimation of the latency parameter requires  $1/sl$  times running of the proposed PF algorithm.

**Table 2.** Relative computational time for different algorithms.

Algorithms	Relative Computational Time
CKF-RD	0.42
Standard PF	1
PF-MD	1.59
Proposed PF	1.65

## 5.2. Problem 2

Consider a ground surveillance problem where a moving target is tracked from a noisy sensor mounted on a moving platform at an approximately known altitude by using only bearing angle observations. This bearing-only tracking (BOT) problem has mainly two components, namely, the target kinematics and the tracking platform kinematics. A representative scenario is given in References [3,45]. The tracking platform motion may be represented by the following equations:

$$\begin{aligned}x_{tp,k} &= \bar{x}_{tp,k} + \Delta x_{tp,k}, \\y_{tp,k} &= \bar{y}_{tp,k} + \Delta y_{tp,k}, \quad k = 0, 1, \dots, n_{step},\end{aligned}$$

where  $x_{tp,k}$  and  $y_{tp,k}$  represent the X and Y coordinates of the tracking platform at  $k$ th time-step, respectively.  $\bar{x}_{tp,k}$  and  $\bar{y}_{tp,k}$  are the known mean coordinates of the platform position and  $\Delta x_{tp,k}$  and  $\Delta y_{tp,k}$  are independent zero-mean Gaussian white noises with variances,  $r_x = 1 \text{ m}^2$  and  $r_y = 1 \text{ m}^2$ , respectively. The mean values for position coordinates (in meters) are  $\bar{x}_{tp,k} = 4kT$  and  $\bar{y}_{tp,k} = 20$ , where  $T$  is sampling time for the discretization expressed in seconds (s).

The target moves in X direction according to following discrete state space relations.

$$x_k = Fx_{k-1} + Gq_{k-1}, \quad (38)$$

where

$$x_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix}, \quad F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}$$

with  $x_{1,k}$  and  $x_{2,k}$  denoting the position (in m) and velocity (in m/s), respectively, of the target.  $q_{k-1}$  is an independent zero-mean white Gaussian noise with variance  $r_q = 0.01 \text{ m}^2\text{s}^4$ . The initial true states are assumed to be  $x_0 = [80 \ 1]^T$ .

The sensor measurement is given by

$$z_k = z_{m,k} + v_k, \quad (39)$$

where

$$z_{m,k} = h(x_{tp,k}, y_{tp,k}, x_{1,k}) = \arctan \frac{y_{tp,k}}{x_{1,k} - x_{tp,k}}$$

is the angle between the X axis and the line of sight from the sensor to target and  $v_k$  is independent Gaussian white noise with zero mean and variance  $r_v = (3^\circ)^2$ . For the implementation of CKF-RD, the measurement noise statistics, on account of the uncertainties in the position of the observer platform, need to be modified as given in Reference [3].

The randomly delayed measurements with packet drops,  $y_{1:k}$ , are generated for 21s by using (3) and (39) with  $N = 2$ . The number of particles used for approximating the pdf is  $N_s = 3000$ . At the beginning of simulation, the latency probability of received measurements is identified offline with  $T = 0.05 \text{ s}$ ,  $m = 400$ , and  $sl = 0.01$ . Latency probability at the end of each ensemble is calculated. The mean value of estimated  $p$  over 100 ensembles is calculated as 0.460, whereas its true value is 0.5. For online identification,  $T$  is taken as 0.1s and the running average of latency probability is calculated at each time step. The plots for the offline and online estimations are shown in the Figure 6a,b, respectively.

Now, the proposed PF with  $N = 1$  and  $N = 2$  are implemented with the true and estimated latency probabilities, respectively. The results of CKF-RD and the standard PF for same set of delayed measurements are compared against that of proposed PF. The RMSE of four filters with the true latency probability  $p = 0.5$  and sampling time  $T = 0.2 \text{ s}$ , which have been calculated over 100 MC runs, are plotted in Figure 7a,b. From the plots, it can be seen that using a filter designed for delayed measurements is a better choice than a standard PF where the delays are not accounted.

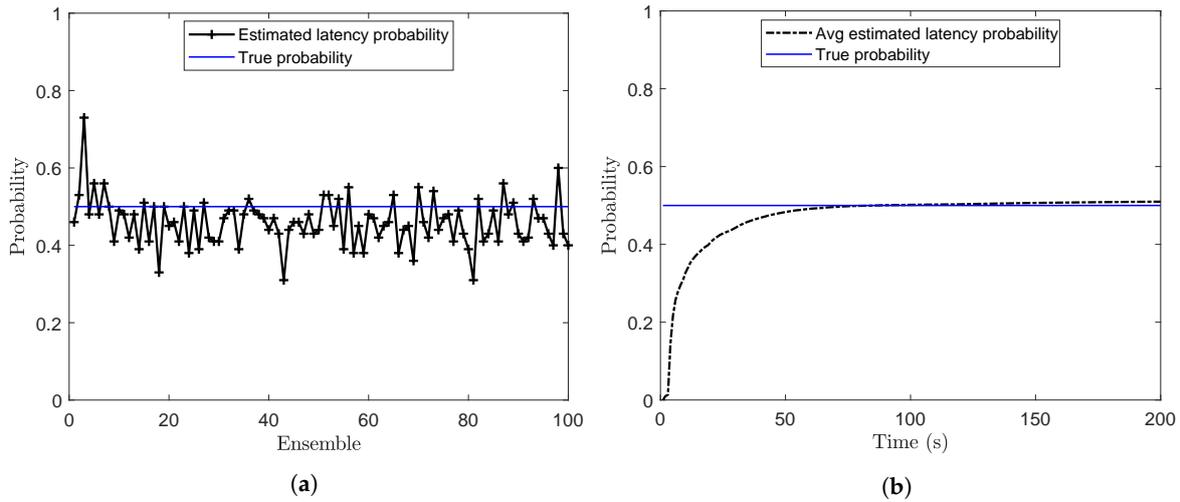


Figure 6. Estimated latency probability (a) Offline estimation (b) Online estimation (Problem 2).

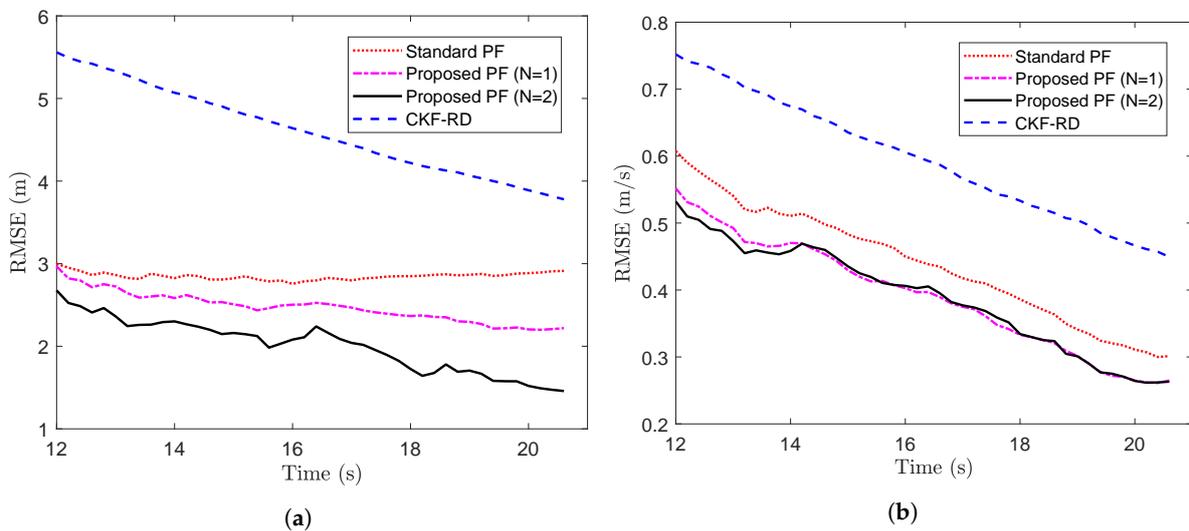


Figure 7. RMSE vs. time steps with  $p = 0.5$ . (a) State  $x_{1,k}$  (b) State  $x_{2,k}$  (Problem 2).

Further, the average RMSE is calculated over 105 time steps for the different values of  $p$ . The average RMSE for two states,  $x_{1,k}$  and  $x_{2,k}$  are plotted in Figure 8a,b respectively. It can be seen that the difference in performance of the proposed PF and the standard PF becomes pronounced as the value of probability increases. However, at higher probabilities, the performance of the proposed filter also gets deteriorated on account of the higher rate of the packet loss.

To compare the performance of the proposed PF with the estimated latency, and PF-MD with the unknown latency, randomly delayed measurements with no packet drops are generated using the measurement model of Reference [29]. The RMSE for three filters are plotted in Figure 9. It can be observed that estimation of the latency affects the filtering process and the use of its incorrect value degrades the performance of a filter. The relative computational burden is given in the Table 3.

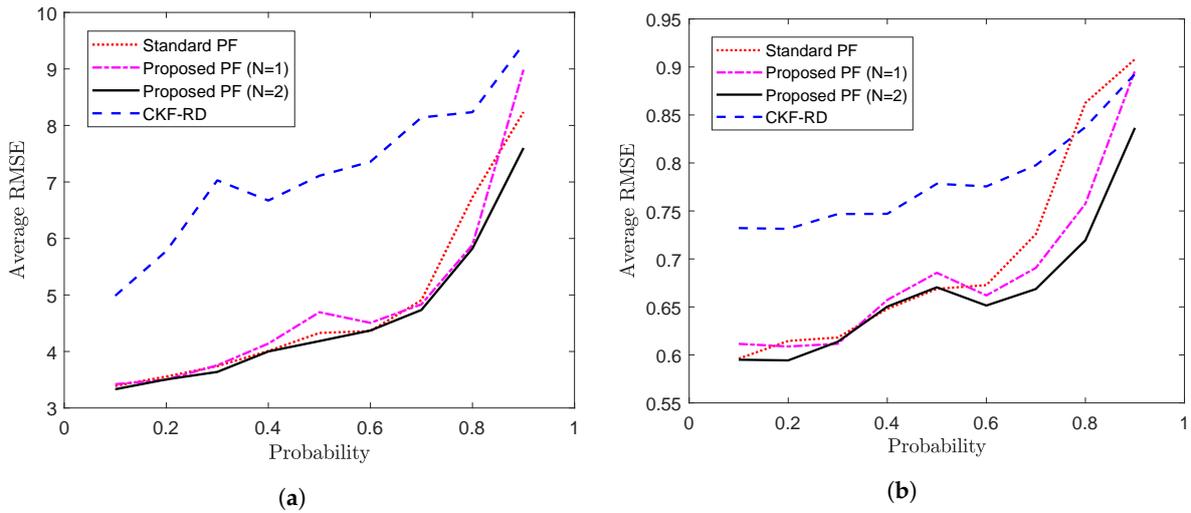


Figure 8. Average RMSE vs. probability (a) State  $x_{1,k}$  (b) State  $x_{2,k}$  (Problem 2).

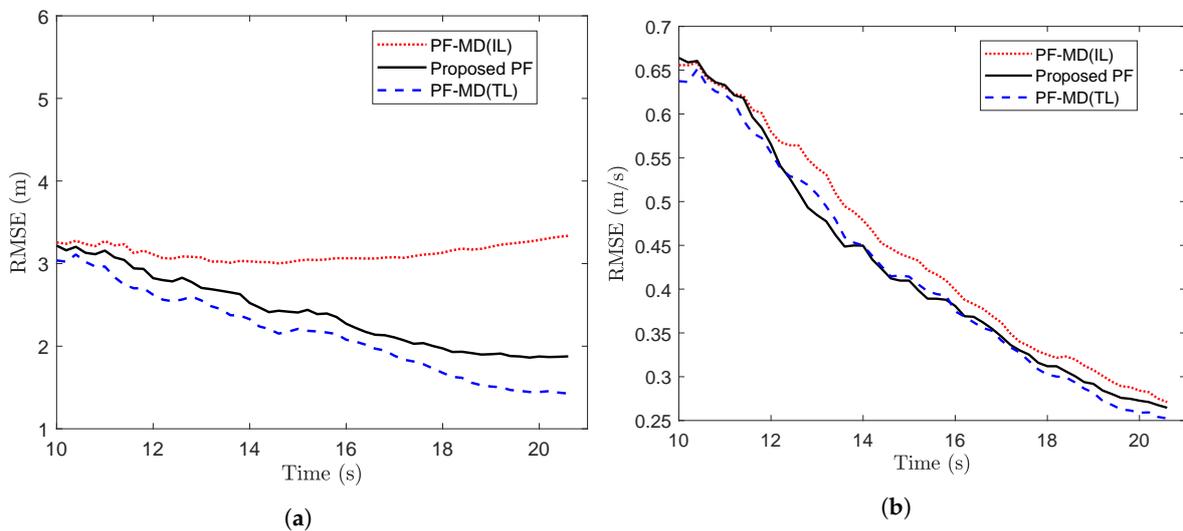


Figure 9. RMSE of different filters for measurements with no packet drops considering  $p = 0.5$ . (a) State  $x_{1,k}$  (b) State  $x_{2,k}$  (Problem 2).

Table 3. Relative computational time for different algorithms.

Algorithms	Relative Computational Time
CKF-RD	0.053
Standard PF	1
PF-MD	1.285
Proposed PF	1.330

5.3. Problem 3

Consider a coordinated turn model for an aerospace target tracking problem for an aircraft that executes a maneuvering turn in a two-dimensional plane with a fixed, but unknown turn rate  $\Omega$ . This uses the bearing and the range measurement observed from a radar to estimate the kinematics of the target. The five-dimensional state vector for the kinematics of aircraft is considered as  $x_k = [\zeta \ \dot{\zeta} \ \eta \ \dot{\eta} \ \Omega]^T$ , where  $\zeta$  and  $\eta$  represent positions, and  $\dot{\zeta}$  and  $\dot{\eta}$  represent velocities along the X and Y co-ordinates, respectively. The discrete-time state dynamics of the aircraft is given as [46]:

$$x_k = \begin{bmatrix} 1 & \frac{\sin \Omega T}{\Omega} & 0 & -\left(\frac{1 - \cos \Omega T}{\Omega}\right) & 0 \\ 0 & \cos \Omega T & 0 & -\sin \Omega T & 0 \\ 0 & \frac{1 - \cos \Omega T}{\Omega} & 1 & \frac{\sin \Omega T}{\Omega} & 0 \\ 0 & \sin \Omega T & 0 & \cos \Omega T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_{k-1} + q_{k-1}, \quad (40)$$

where  $T$  is the time between two successive measurements.  $q_{k-1}$  is a zero mean Gaussian noise with covariance  $Q = \text{diag}[q_1 M \ q_1 M \ q_2 T]$ , where  $q_1$  and  $q_2$  are the noise intensity parameters, and  $M = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^3}{2} & T \end{bmatrix}$ . The range,  $r$ , and bearing,  $\theta$  are the measurement available for target tracking, which are being measured by a radar placed at the origin. The noise-corrupted measurements can be expressed as

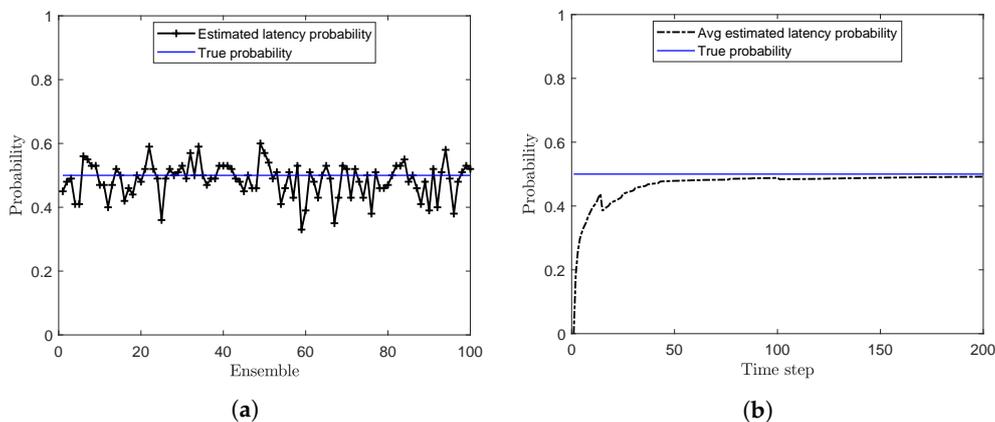
$$z_k = [r_k \ \theta_k]^\top = \left[ \sqrt{\zeta_k^2 + \eta_k^2} \ \tan^{-1} \left( \frac{\eta_k}{\zeta_k} \right) \right]^\top + v_k, \quad (41)$$

where  $v_k$  is an independent zero-mean Gaussian noise with covariance  $R = \text{diag}[\sigma_r^2 \ \sigma_\theta^2]$ . Data for the different parameters used in this simulation are given in Table 4. Initial values for the state and covariance are  $x_0 = [1000 \text{ m} \ 300 \text{ ms}^{-1} \ 1000 \text{ m} \ 0 \text{ ms}^{-1} \ -3^\circ \text{s}^{-1}]^T$  and  $\mathbb{P}_{0|0} = \text{diag}[100 \text{ m}^2 \ 10 \text{ m}^2 \text{s}^{-2} \ 100 \text{ m}^2 \ 10 \text{ m}^2 \text{s}^{-2} \ 100 \text{ mrad}^2 \text{s}^{-2}]$ , respectively.

**Table 4.** Parameters used in tracking

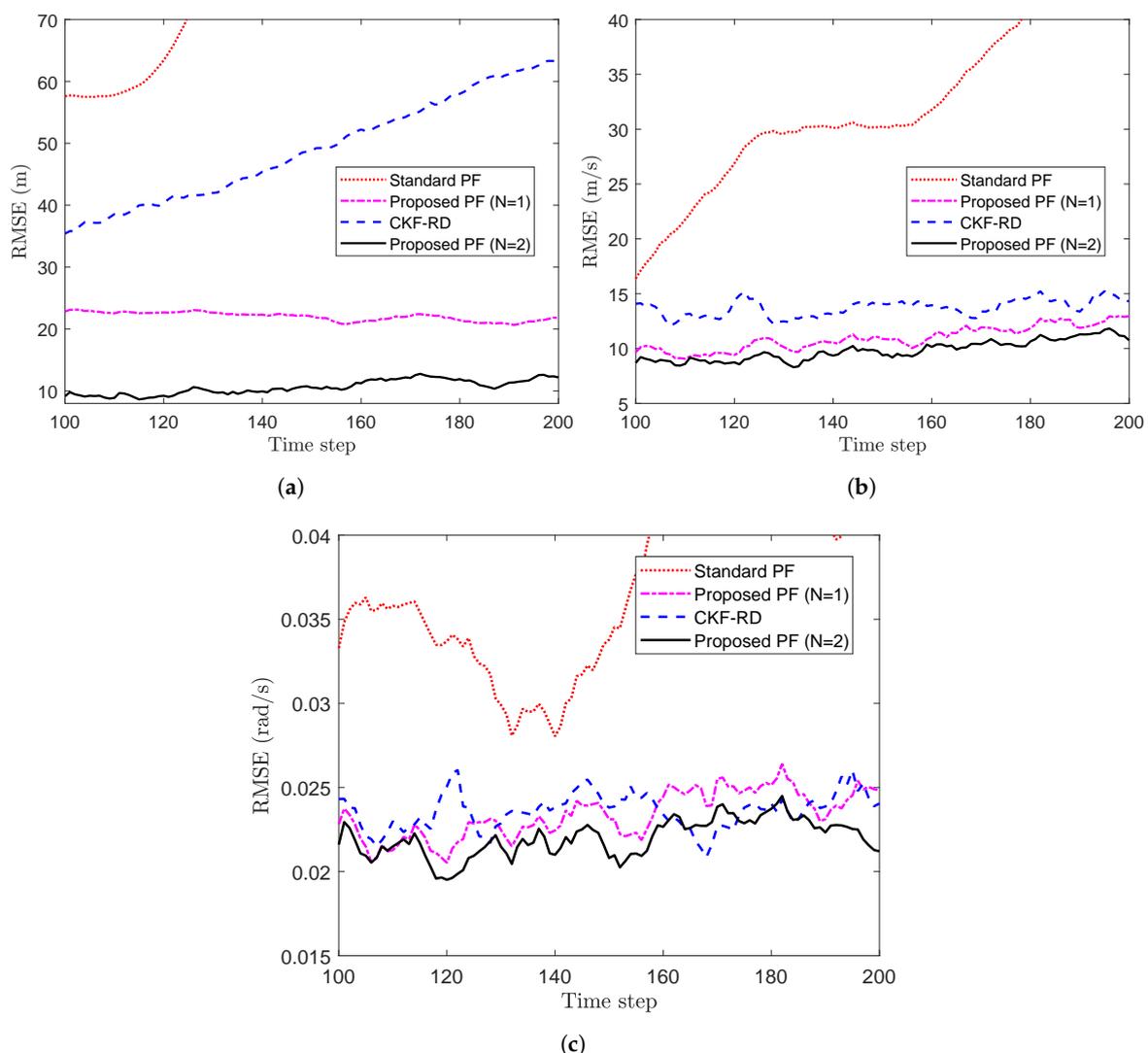
Sampling Time ( $T$ )	0.125 s
Turn Rate ( $\Omega$ )	$-3^\circ \text{s}^{-1}$
$q_1$	$0.1 \text{ m}^2 \text{s}^{-3}$
$q_2$	$1.75 \times 10^{-4} \text{ s}^{-3}$
$\sigma_r$	10 m
$\sigma_\theta$	$\sqrt{10}$ mrad

The randomly delayed measurements with packet drops,  $y_{1:k}$ , are generated by using (41) and (3) with  $N = 2$  and a probability,  $p = 0.5$ . The number of particles used for approximating the pdf is  $N_s = 8000$ . In the beginning of simulation, the latency probability of received measurements is identified offline with  $m = 200$  and  $sl = 0.01$ . The latency probability at the end of each ensemble is calculated and plotted in Figure 10a. The mean value of the estimated probabilities  $\hat{p}$  over 100 ensembles is calculated as 0.4864 against its true value 0.5. Figure 10b shows the online estimation of latency probability.



**Figure 10.** Estimated latency probability (a) Offline estimation (b) Online estimation (Problem 3).

To track the maneuvering target, the proposed PF with  $N = 2$  and  $N = 1$  (a wrong value of the maximum number of delay steps) are implemented. To demonstrate the superiority of the proposed PF, simulation results of the CKF-RD and the standard PF for the same set of measurements and with the true value of latency probability, are compared against that of the proposed PF. The RMSE of the position, velocity and turn rate are the performance metrics as defined in Reference [46]. The RMSE of position, velocity and turn rate for the four filters with  $p = 0.5$ , are plotted in Figure 11a–c, respectively. Since the standard PF does not consider the correlation of current measurement with previously observed states due to random delays, it diverges, particularly in case of the RMSE in position. It can also be observed in the RMSE plots that the proposed PF with  $N = 1$ , which has accounted at least partially for the presence of random delays, performs better than the standard PF. This is significant since the maximum number of delay steps will have to be decided by the user and setting  $N = 1$  might still yield some benefit over assuming no delay. Note that the difference in RMSE of position between the proposed PF and the standard PF or, the CKF-RD is considerable and is worth the relatively higher computational cost paid to obtain it.

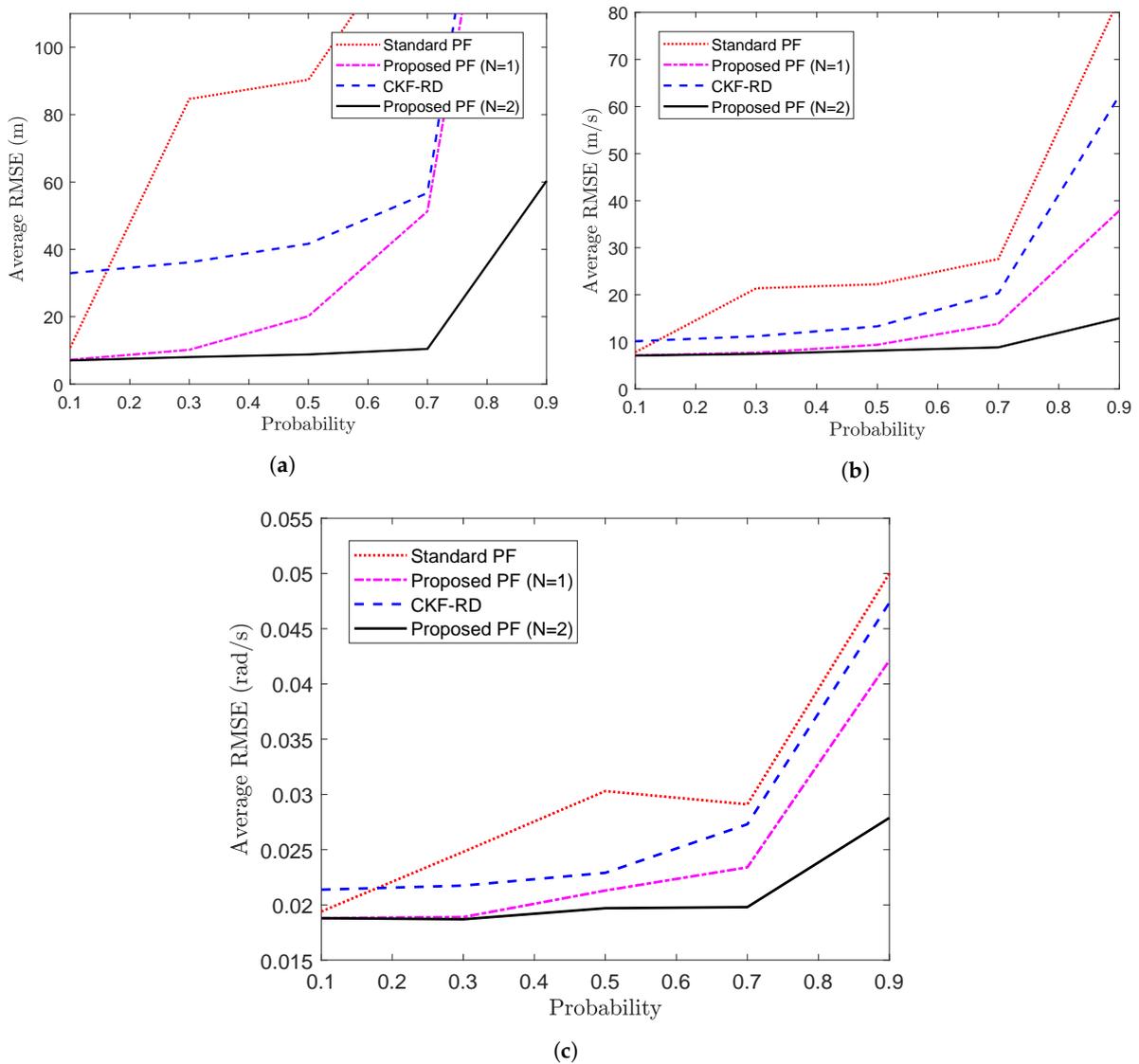


**Figure 11.** RMSE vs. time steps considering  $p = 0.5$ . (a) RMSE of position (b) RMSE of velocity (c) RMSE of turn rate (Problem 3).

Further, to observe the impact of latency probability on the performances of the different filters, the time-averaged value of the RMSEs of the position, velocity and turn rate are plotted against different values of the  $p$  in Figure 12a–c, respectively. It can be seen that with high value of probability,

that is, when the random delays are more likely, the standard PF and proposed PF with  $N = 1$  become significantly less applicable than the proposed PF with  $N = 2$ . It can also be observed that when the probability is high with insufficiently higher value of  $N$ , the rate of packet drops increases and even the performance of the proposed PF ( $N = 2$ ) gets deteriorated.

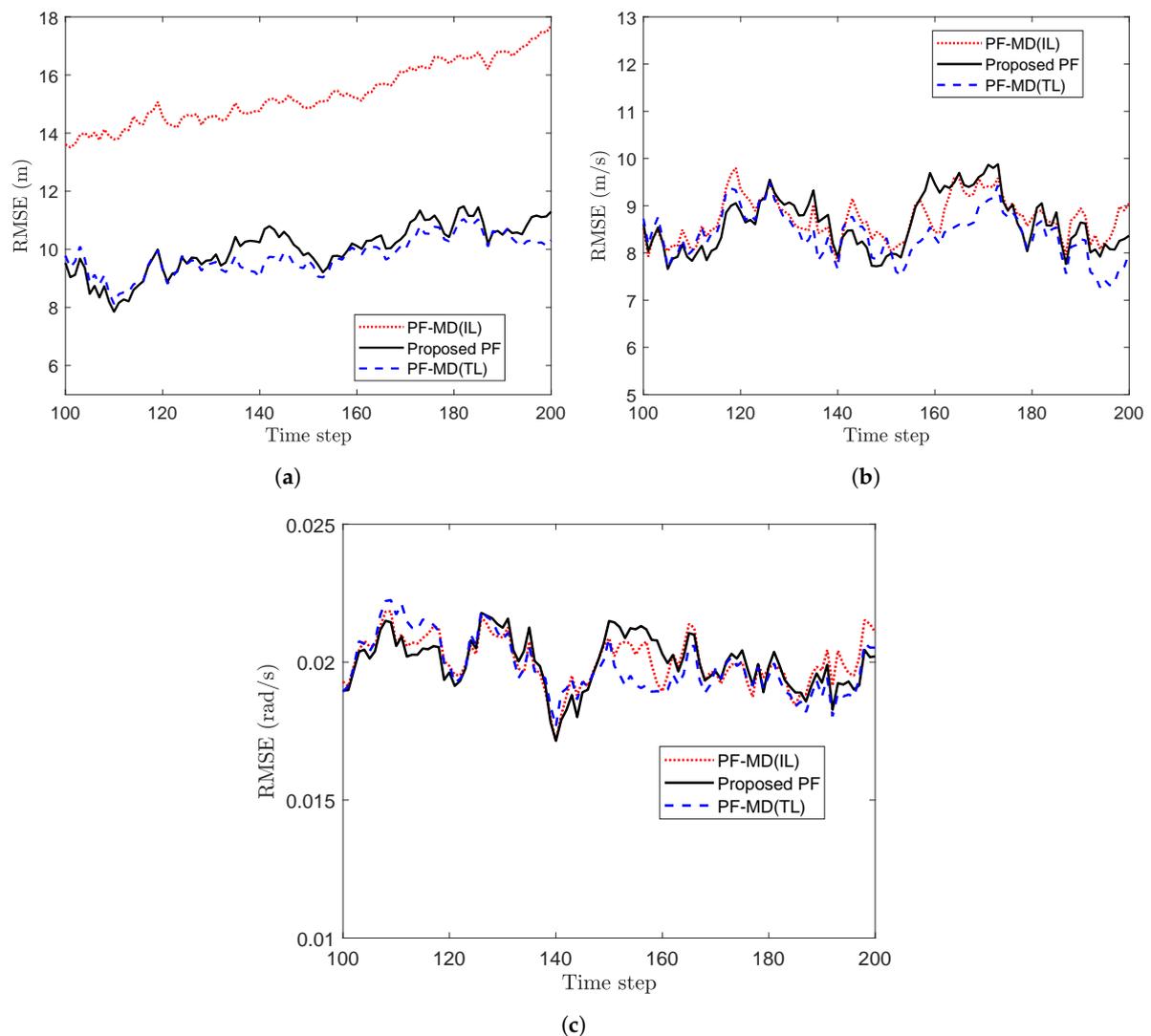
The performance of the proposed PF and PF-MD is compared in terms of the RMSE in Figure 13. To be in synchronization with the model of Reference [29], the measurements used for the estimation are without any packet drops. It can be observed from the plot that without the knowledge of latency, the performance of PF-MD degrades considerably, which suggests estimation of the latency is needed as it is presented in the proposed PF. Table 5 shows the relative computational burden for different filters.



**Figure 12.** Average RMSE vs. probability (a) Average RMSE of position (b) Average RMSE of velocity (c) Average RMSE of turn rate (Problem 3).

**Table 5.** Relative computational time for different algorithms.

Algorithms	Relative Computational Time
CKF-RD	0.025
Standard PF	1
PF-MD	1.512
Proposed PF	1.610



**Figure 13.** RMSE of different filters for measurements with no packet drops considering  $p = 0.5$ . (a) RMSE of position (b) RMSE of velocity (c) RMSE of turn rate (Problem 3).

## 6. Conclusions and Discussion

The standard PF loses its applicability if measurements are randomly delayed at the receiver. The random delays are possible in a system where the sensor and the receiver are connected through a network with bandwidth limitation.

In this paper, a recursion equation of the importance weight was developed to handle delays in measurements. A practical measurement model based on i.i.d. Bernoulli random variables, which includes the possibility of random delays in receiving the observations along with the packet drop situation if any measurement suffers a delay more than the data buffer length, was adopted. Moreover, the latency probability of the received measurements is usually unknown in practical cases.

Hence, this paper presented a method to identify the latency probability in the randomly delayed measurement environment. Further, this paper explored the conditions to ensure the convergence of the proposed PF and the trade-off in selecting the maximum delay.

To validate the performance of the proposed filter and demonstrate its superiority, three numerical examples are simulated using the standard PF, the CKF-RD, the proposed PF with wrong selection of maximum delay and the proposed PF with the correct value of maximum delay. Simulation results show that a filter designed for the delayed measurements, even considering less delay than the actual, performs better than the standard PF. If the random delay is more likely in a system, number of the maximum possible delay should be chosen such that it strikes a balance between avoiding the information loss and minimizing the convergence error. Further, the proposed PF is compared with PF-MD, which is designed for the multi-step randomly delayed measurements without any packet drops and assumes that the latency probability is known. Simulation results showed that the performance of a filter depends on the correct value of latency and its estimation is necessary for the better accuracy of the filtering process.

**Author Contributions:** R.K.T., S.B., P.D. and T.K. conceptualized the work and developed the methodology. Software implementation is done by R.K.T. The results both theoretical and simulation are validated by S.B., P.D. and T.K. The first version of the draft is prepared by R.K.T. and later it was reviewed and modified by S.B., P.D. and T.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Visvesvaraya PhD Scheme, MeitY, Govt. of India, MEITY-PHD-2530. The first and second authors gratefully acknowledge the financial support extended by the organization.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

pdf	Probability density function
PF	Particle filter
EKF	Extended Kalman filter
UKF	Unscented Kalman filter
EM	Expectation-Maximization
SMC	Sequential Monte Carlo
SIR	Sequential importance resampling
LL	Log-Likelihood
CKF-RD	Cubature Kalman filter for randomly delayed measurements
PF-MD(IL)	Particle filter for multiple delayed measurements with incorrect latency
PF-MD(TL)	Particle filter for multiple delayed measurements with true latency
RMSE	Root mean square error
BOT	Bearing-only tracking
i.i.d.	Independently and identically distributed

## References

1. Lee, K.; Johnson, E.N. Robust Outlier-Adaptive Filtering for Vision-Aided Inertial Navigation. *Sensors* **2020**, *20*, 2036. [[CrossRef](#)]
2. Yang, Y.; Shen, Q.; Li, J.; Deng, Z.; Wang, H.; Gao, X. Position and Attitude Estimation Method Integrating Visual Odometer and GPS. *Sensors* **2020**, *20*, 2121. [[CrossRef](#)]
3. Lin, X.; Kirubarajan, T.; Bar-Shalom, Y.; Maskell, S. Comparison of EKF, pseudomeasurement and particle filters for a bearing-only target tracking problem. In *Signal and Data Processing of Small Targets 2002*; International Society for Optics and Photonics: San Diego, CA, USA, 2002; Volume 4728.
4. Li, Y.; Jia, H.; Qi, J.; Sun, H.; Tian, X.; Liu, H.; Fan, X. An Acquisition Method of Agricultural Equipment Roll Angle Based on Multi-Source Information Fusion. *Sensors* **2020**, *20*, 2082. [[CrossRef](#)]
5. Lopes, H.F.; Tsay, R.S. Particle filters and Bayesian inference in financial econometrics. *J. Forecast.* **2011**, *30*, 168–209. [[CrossRef](#)]

6. Stordal, A.S.; Karlsen, H.A.; Nævdal, G.; Skaug, H.J.; Vallès, B. Bridging the ensemble Kalman filter and particle filters: The adaptive Gaussian mixture filter. *Comput. Geosci.* **2011**, *15*, 293–305. [[CrossRef](#)]
7. Ho, Y.; Lee, R. A Bayesian approach to problems in stochastic estimation and control. *IEEE Trans. Autom. Control.* **1964**, *9*, 333–339. [[CrossRef](#)]
8. Crisan, D.; Doucet, A. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Process.* **2002**, *50*, 736–746. [[CrossRef](#)]
9. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [[CrossRef](#)]
10. Nelson, L.; Stear, E. The simultaneous on-line estimation of parameters and states in linear systems. *IEEE Trans. Autom. Control.* **1976**, *21*, 94–98. [[CrossRef](#)]
11. Zuo, J.Y.; Jia, Y.N.; Zhang, Y.Z.; Lian, W. Adaptive iterated particle filter. *Electron. Lett.* **2013**, *49*, 742–744. [[CrossRef](#)]
12. Moose, R.; Dailey, T. Adaptive underwater target tracking using passive multipath time-delay measurements. *IEEE Trans. Acoust. Speech Signal Process.* **1985**, *33*, 778–787. [[CrossRef](#)]
13. Kolmanovsky, I.V.; Maizenberg, T.L. Optimal control of continuous-time linear systems with a time-varying, random delay. *Syst. Control. Lett.* **2001**, *44*, 119–126. [[CrossRef](#)]
14. Nilsson, J.; Bernhardsson, B.; Wittenmark, B. Stochastic analysis and control of real-time systems with random time delays. *Automatica* **1998**, *34*, 57–64. [[CrossRef](#)]
15. Ma, J.; Sun, S. Optimal linear estimators for systems with random sensor delays, multiple packet dropouts and uncertain observations. *IEEE Trans. Signal Process.* **2011**, *59*, 5181–5192.
16. Ma, J.; Sun, S. Linear estimators for networked systems with one-step random delay and multiple packet dropouts based on prediction compensation. *IET Signal Process.* **2016**, *11*, 197–204. [[CrossRef](#)]
17. Liang, X.; Zhang, H. Linear optimal estimator for system subject to random delay and packet dropout without time stamping. In Proceedings of the 2016 35th Chinese Control Conference (CCC), Chengdu, China, 27–29 July 2016; pp. 1843–1847.
18. Schenato, L. Optimal estimation in networked control systems subject to random delay and packet drop. *IEEE Trans. Autom. Control.* **2008**, *53*, 1311–1317. [[CrossRef](#)]
19. Chen, B.; Yu, L.; Zhang, W.A. Robust Kalman filtering for uncertain state delay systems with random observation delays and missing measurements. *IET Control. Theory Appl.* **2011**, *5*, 1945–1954. [[CrossRef](#)]
20. Ray, A.; Liou, L.; Shen, J. State estimation using randomly delayed measurements. *J. Dyn. Syst. Meas. Control.* **1993**, *115*, 19–26. [[CrossRef](#)]
21. Yan, B.; Lev-Ari, H.; Stanković, A.M. Networked State Estimation With Delayed and Irregularly Spaced Time-Stamped Observations. *IEEE Trans. Control. Netw. Syst.* **2017**, *5*, 888–900. [[CrossRef](#)]
22. Hermoso-Carazo, A.; Linares-Pérez, J. Extended and unscented filtering algorithms using one-step randomly delayed observations. *Appl. Math. Comput.* **2007**, *190*, 1375–1393. [[CrossRef](#)]
23. Hermoso-Carazo, A.; Linares-Pérez, J. Unscented filtering algorithm using two-step randomly delayed observations in nonlinear systems. *Appl. Math. Model.* **2009**, *33*, 3705–3717. [[CrossRef](#)]
24. Singh, A.K.; Bhaumik, S.; Date, P. Quadrature filters for one-step randomly delayed measurements. *Appl. Math. Model.* **2016**, *40*, 8296–8308. [[CrossRef](#)]
25. Wang, X.; Liang, Y.; Pan, Q.; Zhao, C. Gaussian filter for nonlinear systems with one-step randomly delayed measurements. *Automatica* **2013**, *49*, 976–986. [[CrossRef](#)]
26. Bhaumik, S.; SWATI. Cubature quadrature Kalman filter. *IET Signal Process.* **2013**, *7*, 533–541. [[CrossRef](#)]
27. Singh, A.K.; Date, P.; Bhaumik, S. A modified Bayesian filter for randomly delayed measurements. *IEEE Trans. Autom. Control.* **2017**, *62*, 419–424. [[CrossRef](#)]
28. Zhang, Y.; Huang, Y.; Li, N.; Zhao, L. Particle filter with one-step randomly delayed measurements and unknown latency probability. *Int. J. Syst. Sci.* **2016**, *47*, 209–221. [[CrossRef](#)]
29. Huang, Y.; Zhang, Y.; Li, N.; Zhao, L. Particle filter for nonlinear systems with multiple step randomly delayed measurements. *Electron. Lett.* **2015**, *51*, 1859–1861. [[CrossRef](#)]
30. Huang, Y.; Zhang, Y.; Li, N. Latency probability estimation of non-linear systems with one-step randomly delayed measurements. *IET Control. Theory Appl.* **2016**, *10*, 843–852. [[CrossRef](#)]
31. Wang, X.; Liang, Y.; Pan, Q.; Wang, Y. Measurement random latency probability identification. *IEEE Trans. Autom. Control.* **2016**, *61*, 4210–4216. [[CrossRef](#)]

32. Wang, Z.; Shen, B.; Liu, X.  $H_\infty$  filtering with randomly occurring sensor saturations and missing measurements. *Automatica* **2012**, *48*, 556–562. [[CrossRef](#)]
33. Li, X.Y.; Sun, S.L.  $H_\infty$  filtering for network-based systems with delayed measurements, packet losses, and randomly varying nonlinearities. *IEEE Sens. J.* **2016**, *16*, 4909–4918. [[CrossRef](#)]
34. Yan, H.; Qian, F.; Yang, F.; Shi, H.  $H_\infty$  filtering for nonlinear networked systems with randomly occurring distributed delays, missing measurements and sensor saturation. *Inf. Sci.* **2016**, *370*, 772–782. [[CrossRef](#)]
35. Zhang, Y.; Huang, Y. Gaussian approximate filter for stochastic dynamic systems with randomly delayed measurements and colored measurement noises. *Sci. China Inf. Sci.* **2016**, *59*, 92–97. [[CrossRef](#)]
36. Saha, S.; Gustafsson, F. Particle filtering with dependent noise processes. *IEEE Trans. Signal Process.* **2012**, *60*, 4497–4508. [[CrossRef](#)]
37. Meyer, F.; Kropfreiter, T.; Williams, J.L.; Lau, R.; Hlawatsch, F.; Braca, P.; Win, M.Z. Message passing algorithms for scalable multitarget tracking. *Proc. IEEE* **2018**, *106*, 221–259. [[CrossRef](#)]
38. Bar-Shalom, Y.; Willett, P.; Tian, X. *Tracking and Data Fusion: A Handbook of Algorithms*; YBS Publishing: Storrs, CT, USA, 2011.
39. Van Der Merwe, R.; Doucet, A.; De Freitas, N.; Wan, E.A. The unscented particle filter. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2001; pp. 584–590.
40. Li, T.; Bolic, M.; Djuric, P.M. Resampling methods for particle filtering: Classification, implementation, and strategies. *IEEE Signal Process. Mag.* **2015**, *32*, 70–86. [[CrossRef](#)]
41. Doucet, A. *On Sequential Simulation-Based Methods for Bayesian Filtering*; Technical Report; Department of Engineering, University of Cambridge: Cambridge, UK, 1998.
42. Gustafsson, F.; Gunnarsson, F.; Bergman, N.; Forssell, U.; Jansson, J.; Karlsson, R.; Nordlund, P.J. Particle filters for positioning, navigation, and tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 425–437. [[CrossRef](#)]
43. Kotecha, J.H.; Djuric, P.M. Gaussian sum particle filtering. *IEEE Trans. Signal Process.* **2003**, *51*, 2602–2612. [[CrossRef](#)]
44. Wang, X.; Liang, Y.; Pan, Q.; Zhao, C.; Yang, F. Design and implementation of Gaussian filter for nonlinear system with randomly delayed measurements and correlated noises. *Appl. Math. Comput.* **2014**, *232*, 1011–1024. [[CrossRef](#)]
45. Sadhu, S.; Mondal, S.; Srinivasan, M.; Ghoshal, T.K. Sigma point Kalman filter for bearing only tracking. *Signal Process.* **2006**, *86*, 3769–3777. [[CrossRef](#)]
46. Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control.* **2009**, *54*, 1254–1269. [[CrossRef](#)]

