CrossMark

SURVEY AND STATE OF THE ART

# Particle filtering with applications in networked systems: a survey

Wenshuo Li[1] · Zidong Wang[2,3] · Yuan Yuan[1] · Lei Guo[1]

**Abstract** The particle filtering algorithm was introduced in the 1990s as a numerical solution to the Bayesian estimation problem for nonlinear and non-Gaussian systems and has been successfully applied in various fields including physics, economics, engineering, etc. As is widely recognized, the particle filter has broad application prospects in networked systems, but network-induced phenomena and limited computing resources have led to new challenges to the design and implementation of particle filtering algorithms. In this survey paper, we aim to review the particle filtering method and its applications in networked systems. We first provide an overview of the particle filtering methods as well as networked systems, and then investigate the recent progress in the design of particle filter for networked systems. Our main focus is on the state estimation problems in this survey, but other aspects of particle filtering approaches are also highlighted.

✉ Zidong Wang
Zidong.Wang@brunel.ac.uk

Lei Guo
lguo@buaa.edu.cn

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

[2] College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao 266590, China

[3] Department of Computer Science, Brunel University London, UB8 3PH, Uxbridge Middlesex, UK

## Introduction

In recent years, many fruitful results have been published regarding Bayesian inference. Analytically tractable solutions to the Bayesian inference, however, exist only for a limited class of models. Enormous research efforts have, therefore, been put to develop approximate solutions for the Bayesian inference, among which the particle filtering methods, using the sampling-based approximation techniques, have gained increasing popularity due to their ability to provide an arbitrarily close approximation to the true probability density function (PDF). Particle filtering approaches have been found to be especially attractive for nonlinear/non-Gaussian filtering problems where the assumptions enabling the Kalman-type filters are violated.

On another research frontier, with the rapid development of wireless communication technology, networked systems have found applications in a wide range of areas such as process monitoring, formation control, tele-operations, etc. Despite their advantages such as structural flexibility and low cost, networked systems have given rise to new challenges to traditional state estimation approaches. The challenges come mainly from two aspects: (i) there may be certain degree of information loss due to the limited network resources and, (ii) the trade-off between accuracy and communication cost must

مدينة الملك عبدالعزيز
KACST للعلوم والتقنية

Springer

be addressed, especially for large-scale networks. Furthermore, the constraints on communication cost are highlighted when particle filtering methods are applied to networked systems.

The purpose of this paper is to provide a thorough and timely review of existing results on particle filtering methods and their applications to networked systems. The rest of this paper is organized as follows. In Sect. 2, the basic idea of the particle filter is introduced, and some improvements on the existing particle filtering algorithms are discussed. State estimation for networked systems are studied in Sect. 3, where existing filtering algorithms addressing network-induced phenomena are discussed and distributed filtering methods are briefly reviewed. In Sect. 4, the applications of particle filtering methods to the state estimation of networked systems are investigated. Particle filtering under network-induced phenomena and particle filtering for networked systems are reviewed, respectively. Conclusions and future research topics are presented in Sect. 5.

## Particle filter

### Basic ideas

It is known that the Kalman-type filters are not suitable for state estimation for systems with non-Gaussian noises and/or strong nonlinearities since the Gaussian assumption on the state posterior is no longer valid. Particle filtering (PF), with the capability of approximating PDFs of any form, has received considerable attention among researchers and engineers since it was proposed in the 1990s. For some excellent research reviews on particle filtering method, the readers can refer to [1–3], to name just a few.

Next we will briefly introduce the basic principles of particle filtering. Consider the state space model given by

$$\begin{cases} x_{k+1} = f_k(x_k, v_k) \\ z_k = h_k(x_k, n_k) \end{cases} \tag{1}$$

where $\{x_k, k \in N\}$ is the state sequence which is of interest to us; $\{z_k, k \in N\}$ is the observed signal sequence; $f_k : R^{n_x} \times R^{n_v} \to R^{n_x}$ and $h_k : R^{n_x} \times R^{n_n} \to R^{n_z}$ are both nonlinear functions; $v_k$ and $n_k$ are the process noise and measurement noise, respectively, both of which are assumed to be an independent identical distributed (i.i.d.) process.

Our aim here is to estimate, in a recursive manner, the current state $x_k$ given the measurements up to time $k$, denoted by $z_{1:k} := \{z_i, i = 1, 2, ..., k\}$. The Bayesian method for state estimation seeks to construct the posterior PDF $p(x_k|z_{1:k})$ in which the complete statistical information of $x_k$ is contained. Suppose that $p(x_{k-1}|z_{1:k-1})$ is available at time $k$, then $p(x_k|z_{1:k})$ can be obtained through the following two-stage calculation:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\mathrm{d}x_{k-1} \tag{2}$$

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{\int p(z_k|x_k)p(x_k|z_{1:k-1})\mathrm{d}x_k} \tag{3}$$

Therefore, as long as we know the initial PDF $p(x_0|z_0)$, we can, in theory, obtain $p(x_k|z_{1:k})$ using formulas (2) and (3) recursively at each filtering period.

For linear systems with Gaussian additive noise, (2) and (3) can be solved analytically to obtain $p(x_k|z_{1:k})$ and the state estimate $\hat{x}_k$ which maximizes $p(x_k|z_{1:k})$ is exactly the solution of Kalman filter. For systems with a more general form, however, the analytical expression of $p(x_k|z_{1:k})$ is usually unavailable due to the computationally intractable integrals in (2) and (3). When this is the case, one can apply particle filtering algorithm to obtain an approximation of $p(x_k|z_{1:k})$. Particle filter is a sequential Monte Carlo method, the basic idea of which is to represent the state posterior $p(x_k|z_{1:k})$ by a set of particles $\{x_k^i, i = 1, 2, \ldots, N_s\}$ with associated weights $\{w_k^i, i = 1, 2, \ldots, N_s\}$:

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i) \tag{4}$$

where $N_s$ is the number of particles, $\delta(\cdot)$ is the Dirac delta function, and the weights are normalized such that $\sum_i w_k^i = 1$. In this way, the integrals in (2) and (3) are transformed into summations which are easier to calculate.

Now it is natural for one to ask that how could we sample from $p(x_k|z_{1:k})$ which is unknown to us. Particle filter does this by means of importance sampling. Suppose that there is a probability density $q(x_k|z_{1:k})$ which is known to us and from which we can draw samples easily. Typically we choose $q(x_k|z_{1:k})$ such that the recursion

$$q(x_k|z_{1:k}) = q(x_k|x_{k-1}^i, z_k)q(x_{k-1}|z_{1:k-1}) \tag{5}$$

is satisfied. Then, the particle representation of $p(x_k|z_{1:k})$ is given by

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} \tilde{w}_k^i \delta(x_k - \tilde{x}_k^i) \tag{6}$$

where $\tilde{x}_k^i \sim q(x_k|z_{1:k})$ $(i = 1, 2, \ldots, N_s)$ is the set of particles; $\tilde{w}_k^i \propto \frac{p(x_k^i|z_{1:k})}{q(x_k^i|z_{1:k})}$ is the normalized weight of the $i$th particle. The density $q(x_k|z_{1:k})$ is called *importance density* and its distribution is called *proposal distribution*.

In practical applications, one is normally more interested in sequential implementation of the Monte Carlo approximation in (6), that is, to represent $p(x_k|z_{1:k})$ using

$p(x_{k-1}|z_{1:k-1})$. Suppose that at time $k$, we have the following discrete approximation of $p(x_{k-1}|z_{1:k-1})$:

$$p(x_{k-1}|z_{1:k-1}) \approx \sum_{i=1}^{N_s} \tilde{w}_{k-1}^i \delta(x_{k-1} - \tilde{x}_{k-1}^i) \qquad (7)$$

Note that

$$
\begin{aligned}
p(x_k|z_{1:k-1}) &= \int p(x_k, x_{k-1}|z_{1:k-1}) dx_{k-1} \\
&= \int p(x_k|x_{k-1}) p(x_{k-1}|z_{1:k-1}) dx_{k-1} \\
&\approx \int p(x_k|x_{k-1}) \sum_{i=1}^{N_s} \tilde{w}_{k-1}^i \delta(x_{k-1} - \tilde{x}_{k-1}^i) dx_{k-1} \\
&= \sum_{i=1}^{N_s} \left\{ \int \tilde{w}_{k-1}^i p(x_k|x_{k-1}) \delta(x_{k-1} - \tilde{x}_{k-1}^i) dx_{k-1} \right\} \\
&= \sum_{i=1}^{N_s} \tilde{w}_{k-1}^i p(x_k|\tilde{x}_{k-1}^i)
\end{aligned}
$$

Therefore, if we draw samples $\tilde{x}_k^i \sim p(x_k|\tilde{x}_{k-1}^i)$ for $i = 1, 2, \ldots, N_s$, the prediction density $p(x_k|z_{1:k-1})$ can be approximated by $\{\tilde{x}_k^i, \tilde{w}_{k-1}^i\}, i = 1, 2, \ldots, N_s$. Furthermore, since

$$
\begin{aligned}
p(x_k|z_{1:k}) &= \frac{p(z_k, x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} = \frac{p(z_k|x_k) p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \\
&\propto p(z_k|x_k) p(x_k|z_{1:k-1})
\end{aligned}
$$

the importance weights of $\tilde{x}_k^i$ should be updated as follows:

$$\tilde{w}_k^i \propto \tilde{w}_{k-1}^i p(z_k|\tilde{x}_k^i) \qquad (8)$$

for $i = 1, 2, \ldots, N_s$.

The weight update formula in (8) can be extended directly to the more general case where the importance density is selected as $q(x_k|z_{1:k})$ satisfying (5):

$$\tilde{w}_k^i \propto \tilde{w}_{k-1}^i \frac{p(z_k|\tilde{x}_k^i) p(\tilde{x}_k^i|\tilde{x}_{k-1}^i)}{q(\tilde{x}_k^i|\tilde{x}_{k-1}^i, z_k)} \qquad (9)$$

The standard particle filtering algorithm introduced above is also termed as *sequential importance sampling* (SIS) algorithm. For clarity, we present its pseudo code in Algorithm 1.

---

**Algorithm 1** Sequential Importance Sampling Algorithm

Initialize $\{\tilde{x}_0^i, \tilde{w}_0^i\}$.
**for** $k = 1, \ldots, N - 1$ **do**
   Sample $\tilde{x}_k^i \sim q(\tilde{x}_k^i|\tilde{x}_{k-1}^i, z_k)$.
   Update weights according to $\tilde{w}_k^i \propto \tilde{w}_{k-1}^i \frac{p(z_k|\tilde{x}_k^i) p(\tilde{x}_k^i|\tilde{x}_{k-1}^i)}{q(\tilde{x}_k^i|\tilde{x}_{k-1}^i, z_k)}$.
**end for**

---

Although [4] is widely recognized to be the work which lays the foundation for modern particle filtering, the history of the Monte Carlo method can trace back to the 1940s [5]. In [5], the Monte Carlo method is introduced by Metropolis as a branch of statistical mechanics where one's major concern is the collective behaviour of a group of particles. A statistical study based on samples (particles) drawn from all possible events is suggested to avoid dealing with multiple integrals or multiplications of the probability matrices. Soon after that, the Markov Chain Monte Carlo (MCMC) method was proposed by Metropolis inspired by the search of thermodynamic equilibrium through simulation [6]. The major finding is that one does not have to know the exact dynamics of the system in the simulation; instead, he only needs to simulate a Markov chain which has the same equilibrium as the original system. This scheme for simulation is then referred to as *Metropolis algorithm*. The generalized version of Metropolis Algorithm, also known as *Metropolis-Hasting MCMC* (MH-MCMC) algorithm, has been proposed in 1970 [7]. A thorough introduction of the MCMC method could be found in [8] and [9]. In the following, we will briefly introduce MH-MCMC algorithm to show how the idea of MCMC algorithm is implemented.

To draw samples from $\pi(x)$, we construct a Markov chain with $\pi(x)$ as its invariant distribution, i.e.,

$$p(x_k) \to \pi(x) \text{ask} \to \infty, \quad \text{for any } p(x_0)$$

where $p(x_k)$ is the marginal probability of the Markov chain and $p(x_0)$ is its initial value. It is known that a Markov chain is characterized by the initial probability $p(x_0)$ and the transition probability $T(x_{k+1}|x_k)$. In the MH-MCMC algorithm, $p(x_0)$ can be chosen arbitrarily and the state transition is realized by sampling from a proposal distribution $q(x'|x_k)$ and accepting the new sample $x'$ with the following probability:

$$A(x', x_k) = \min\left(1, \frac{p(x')q(x_k|x')}{p(x_k)q(x'|x_k)}\right).$$

If the proposal is accepted, the Markov chain state will be updated by $x'$, i.e., $x_{k+1} = x'$; otherwise it remains at $x_k$, i.e., $x_{k+1} = x_k$.

The pseudo code of MH-MCMC algorithm is given in Algorithm 2.

The introduction of MCMC method here is out of three considerations. First, it shares the similar idea with particle filtering. Specifically, both of them represent the unknown target distribution by a set of randomly drawn samples to avoid the intractable integrals. Second, it has been combined with the standard particle filter, as shown later, to reduce the problem of sample depletion in particle filter. Third, the missing data problems, which are one major concern of this survey due to their universality in networked systems, have

**Algorithm 2** Metropolis-Hasting MCMC Algorithm

Initialize $x_0$.
**for** $k = 0, 1, ..., N - 1$ **do**
  Sample $x' \sim q(x'|x_k)$.
  Sample $u \sim Uniform(0, 1)$.
  **if** $u < A(x', x_k) = \min\left(1, \frac{p(x')q(x_k|x')}{p(x_k)q(x'|x_k)}\right)$ **then**
    $x_{k+1} = x'$.
  **else**
    $x_{k+1} = x_k$.
  **end if**
**end for**

been tackled by researchers using certain forms of MCMC method, such as the Gibbs sampler [10] and other data augmentation methods.

Comparing the importance sampling with the MCMC method, we see that both can obtain samples from a PDF that is known to us up to a normalizing constant. Nevertheless, some key differences between them should be noted. In the importance sampling, no iterations are involved, but the samples obtained are associated with different weights, which implies a lower computational efficiency since particles with different weights take up the same amount of computational resources. By contrast, all the samples obtained are equally weighted in the MCMC method. However, the MCMC method requires iterative sampling from the proposal distribution to ensure that the invariant distribution is finally reached, which can be time prohibitive in some real-time applications.

Before moving on to discuss some related problems of particle filtering, we will at first give a brief review on the convergence results. In practice, one cannot apply the particle filtering method with confidence until satisfactory answers are given to the following questions:

- Could the particle approximation of $p(x_k|z_{1:k})$ converge asymptotically to the true $p(x_k|z_{1:k})$ and in what sense?
- Is there error accumulation over time?

In [11], the law of large numbers for particle filtering has been established. It is proved that the particle representation converges almost surely to the quantity of interest as the number of particles tends towards infinity. The law of large numbers, however, does not provide a measure of the approximation error which is usually of more interest to practitioners. Here it is natural for one to think of the central limit theorem which can offer the probability distribution of approximation error. Unfortunately, the classical central limit theorem, assuming that samples are drawn independently from the same distribution, does not apply to the analysis of particle filtering where there is interaction between particles.

A comprehensive survey of convergence results on particle filters could be found in [12] where almost sure convergence and mean square convergence of particle filtering are studied, respectively. Note that the mean square convergence results stated in [12] relies on some strict assumptions. For example, the convergence of mean square error has only been established for bounded functions, which has excluded $f(x) = x$, meaning that this convergence result does not apply to the classical mean square estimation. Furthermore, it has been shown in [12] that the error accumulation seems to be inevitable, unless certain mixing conditions on the dynamic model (thus the true optimal filter) are satisfied. This also explains why particle filtering is not suitable for (fixed) parameter estimation. To avoid error accumulation, one has to increase the number of particles with time, which may lead to a formidable computation load.

The central limit theorem for particle filters has been established in [13] which, due to its minimal assumptions on the distribution of interest, applies to various forms of particle filtering algorithms. The asymptotic variance allows us to compare the relative efficiency of different algorithms and assess the stability of a given particle filter. More recently, the convergence result for a rather general class of unbounded functions has been obtained (see [14]). Shortly the result was extended to $L_p$-convergence in [15]. Notably, both the results in [14,15] require that the unnormalized importance weights are point-wise bounded. This constraint has been relaxed in [16] where only boundedness of the second (for mean square convergence) or fourth (for $L_4$-convergence) order moment of importance weights is required.

## Related problems and improvements

### Degeneracy problem

Theoretically, there are an infinite number of possible choices for the importance density $q(x_k|z_{1:k})$. In practice, some choices are superior over the others since they are closer to the optimal importance density $p(x_k|x_{k-1}^i, z_k)$ (see [17]). Here the optimality is defined in the sense that the variance of the important weights is minimized. In fact, it has been shown (see [18]) that the unconditional variance of importance weights can only increase over time, which leads to a common problem with particle filtering methods, i.e., the degeneracy phenomenon. Intuitively, we hope that all the particles are evenly weighted to guarantee the efficiency of the algorithm. The actual situation, however, is that after a few filtering iterations, only one particle will have a significant weight, while all the others play a negligible role in the representation of state posterior. Kong et al. introduced in [18] the effective sample size (ESS) as a measure of degeneracy, defined as

$$ESS = \frac{N_s}{1 + \text{var}(\tilde{w}_k^i)} \tag{10}$$

where $\text{var}(\tilde{w}_k^i)$ is the variance of importance weights. It is clear from (10) that a large variance of importance weights implies a small effective sample size, hence a severe degeneracy.

One way to reduce particle degeneracy is to use optimal importance density $p(x_k|x_{k-1}^i, z_k)$ so that $\text{var}(\tilde{w}_k^i)$ in (10) is minimized. With very few exceptions, however, it is impossible in practice to evaluate $p(x_k|x_{k-1}^i, z_k)$ analytically. Therefore, many suboptimal schemes for importance density selection have been proposed. One idea they have in common is that current measurements should be taken into account when constructing the importance density. The proposal distribution is said to be an adapted one if the current measurements are incorporated. The bootstrap particle filter proposed by Gordon et al. [4] uses the state transition probability $p(x_k|x_{k-1})$ as the importance density. Even though the algorithm is simple to realize, it has ignored the current measurements, which might cause a large deviation between the predicted particles and the actual support of the posterior PDF. Compared with the true optimal importance density, the Gaussian approximation of it is much easier to evaluate. A variety of tools for the calculation of Gaussian approximation of $p(x_k|x_{k-1}^i, z_k)$ are available, including extended Kalman filter (EKF), unscented Kalman filter (UKF), etc. The procedure is rather simple: when new measurements are received, a Kalman-type propagation is performed to obtain the Gaussian approximation of $p(x_k|x_{k-1}^i, z_k)$. This approximation is then used as the importance density from which the new set of particles is drawn. The filtering method is termed as extended particle filter (EPF)/unscented particle filter (UPF) when EKF/UKF is employed to calculate the importance density function (see [19,20], respectively).

It has been observed that the mixture structure of particle filters will cause a great increase in running time when adaptation is performed [21]. To adapt the proposal distribution without a great loss in efficiency, the auxiliary particle filter (APF) has been introduced, whose basic idea is to carry out the particle filtering algorithm in a higher dimension. The motivation is that it is wasteful to draw particles which will at last be abandoned with a large probability. In the APF algorithm, an auxiliary variable $j^i$, serving as the index of the particle $\tilde{x}_{k-1}^i$, is weighted at the beginning of the $k$th iteration according to the compatibility of $\tilde{x}_{k-1}^i$ given $z_{1:k}$. The new set of particles, $\{\tilde{x}_k^i, i = 1, 2, \ldots, N_s\}$, is then sampled from the modified state transition probability in which the weighted index is incorporated. It is revealed in [22] that essentially the APF method is equivalent to adding a well-designed resampling step (see details below) before each iteration of the standard SIS procedure.

Generally, the suboptimal proposal distribution should be constructed on a case by case basis. In [23], a problem-specific proposal distribution has been designed for radar tracking based on particle filter. The particle swarm optimization (PSO) has been used in [24] to optimize the proposal distribution for the simultaneous localization and mapping (SLAM) problem. In [25], the ensemble Kalman filter (EnKF) has been employed to define the proposal density of particle filter for soil moisture estimation.

Another way to reduce degeneracy is to perform resampling at each filtering iteration. Resampling is a procedure in which particles $\{\tilde{x}_k^i, i = 1, 2, \ldots, N_s\}$ are reselected in accordance with their weights $\{\tilde{w}_k^i, i = 1, 2, \ldots, N_s\}$. In this way, the particles with larger weights will have a greater number of offspring while those with negligible weights are simply discarded. The motivation is to conserve our computing resources for the particles which will play greater roles. After resampling, one gets a new set of particles which are equally weighted and distributed according to the state posterior $p(x_k|z_{1:k})$. If resampling is performed after each iteration of the SIS procedure, such algorithm is referred to as *sampling importance resampling* (SIR) algorithm. There are various resampling schemes, such as stratified sampling [26], residual sampling [27], systematic sampling [28], exquisite resampling [29], etc. A recent review of existing resampling algorithms could be found in [30].

One problem caused by the resampling step is the so-called *sample impoverishment*. It is found that when resampling is performed, all the particles will locate at the same point in the state space after a few iterations, implying that the diversity of particles is lost, which may lead to a severe deterioration in the capability of particles for representing the state posterior. Theoretically, the problem of sample impoverishment can be avoided if we are able to resample from a continuous distribution rather than a discrete one. Based on the above consideration, the regularized particle filter (RPF) has been proposed in [31] where the Kernel density is introduced to approximate the true posterior density with a continuous density function. In [32], the regularized auxiliary particle filter (RAPF) which combines the RPF and APF methods has been presented to diversify the particles.

The effect of particle impoverishment is especially significant in smoothing problems where the estimation is derived based on particle paths. In smoothing estimation, each particle denotes a complete realization of state evolution rather than the current state only, which implies that the elements of a particle, in particular those corresponding to earlier states, have to be resampled many times with the running of smoothing algorithm. As a result, most particles will share a common path, thus incapable of representing the smoothing distribution. To address this problem, we hope to obtain a different set of particles which are still distributed according to the smoothing distribution. As discussed in the previous section, this can be done by the MCMC method. A key step to implement the MCMC method is to construct a Markov chain with $p(x_k|z_{1:k})$ as its invariant distribution. In [33], a Gibbs sam-

pler has been used to update the state of Markov chain. The Metropolis-Hasting (MH) sampler has been adopted in [34] to generate new particles. It is also shown in [34] that the support of the smoothing distribution could be improved through the MCMC procedure.

Another problem brought up by the resampling step is the increased computational complexity. This is due to the fact that resampling is the only step in the particle filtering algorithm that hinders a parallel implementation. Based on this observation, it is suggested in [35] that the resampling step should be abandoned when its disadvantages outweigh advantages. In [35], the Gaussian particle filtering (GPF) method has been proposed where the state posterior is approximated by a Gaussian distribution whose mean and covariance are propagated using sequential importance sampling. Since an average is calculated in each iteration over the entire set of particles, we do not need to worry about the problem of particle degeneracy any more. Hence there is no need to resample the particles, i.e., a fully parallel implementation becomes possible.

*Variance reduction*

For a given estimation $I_{k|k}(g(x_k))$ of $g(x_k)$ based on $z_{1:k}$, we hope its variance is as small as possible. Numerous results have been reported on variance reduction for particle filtering (see [33] and references therein). In [36], the methods of SIS, SIR and APF are compared from the perspective of variance reduction. It is discovered that the resampling step in the SIR procedure has led to an increase in variance from two aspects: First, the fact that resampling is performed on a discrete distribution has introduced dependence among samples, which further leads to a larger variance compared with the fully adapted APF algorithm; Second, the randomness of the resampling step itself has produced an extra variance term. To avoid the extra variance caused by resampling while coping with particle degeneracy, a hybrid algorithm is proposed which automatically switches between SIS and APF according to whether a serious decrease in the effective sample size is detected.

For state estimation of the jump Markov linear systems (JMLS), it is suggested that the salient model structure should be made good use of [33]. It is shown that given the current state of Markov chain, the state estimation problem for JMLS reduces to Kalman filtering, which allows for a closed-form solution. Taking advantage of this property, one can draw samples from a lower dimensional distribution, where the continuous states are marginalized out. A lower dimensional distribution means a reduced number of particles and thus a lower computational complexity. Further, it is revealed that sampling from a lower dimensional distribution will result in a smaller variance. This approach has borrowed the idea of Rao-Blackwellization [37] which is especially useful for

nonlinear model with a linear substructure. The well-known variance decomposition formula

$$\text{var}[\tau(U, V)] = \text{var}[E\{\tau(U, V)|V\}] + E\left[\text{var}\{\tau(U, V)|V\}\right] \tag{11}$$

forms the theoretical basis of the Rao-Blackwellization method. From (11), it is not difficult to see that $E\{\tau(U, V)|V\}$, in which the random variable $V$ is integrated out, has the same mean with $\tau(U, V)$, yet a reduced variance. It is therefore advantageous to integrate out any redundant random variable present in the estimation. Specifically, when state can be decomposed as $x = [x_1^T, x_2^T]^T$, where $x_1$ is conditionally linear given $x_2$ and measurements $z$, i.e., $p(x_1|x_2, z)$ can be calculated analytically by using the standard Kalman filter, we can decompose the posterior density in the following way

$$p(x_1, x_2|z) = p(x_1|x_2, z)p(x_2|z)$$

Next we only need to approximate $p(x_2|z)$ with a set of particles $\{\tilde{x}_2^i, i = 1, 2, \ldots, N_s\}$. For each particle $\tilde{x}_2^i$, a Kalman filter is designed to calculate $p(x_1|\tilde{x}_2^i, z)$. Particle filters applying Rao-Blackwellization method are also referred to as *marginalized particle filter* in some literatures [38–40].

*Robust particle filter*

Up to now, we have assumed, in our particle filter design, that the system dynamics and noise statistics are precisely known to us, which might not hold in practical applications. Various methods have been put forward to robustify the particle filtering algorithm for systems with unknown statistics. The box particle filtering, which combines sequential Monte Carlo method with interval analysis, has been introduced in [41–43]. Unlike the standard particle filtering method where particles are points in the state space and likelihood functions are defined by a statistical model, the box particle filter uses multidimensional intervals in the state space as particles and a bounded error model to evaluate the likelihood functions. The key advantage of box particle filter is the reduced number of particles required for a specified accuracy in the presence of model uncertainty.

In [44], a cost-reference particle filtering (CRPF) method has been proposed, which can be seen as a generalization of the standard particle filter. The CRPF method takes advantage of a basic fact that the methodology of particle representation and propagation can be applied to any function of the state as long as it admits a recursive decomposition. In CRPF, a user-defined cost function, instead of the state posterior as is the case with standard particle filter, is defined and minimized following a procedure similar to that of standard particle filter. CRPF has been combined with APF in

[45] for target tracking in binary sensor networks without probabilistic assumptions on the model. It is shown that the CRPF and APF have a similar form when the cost functions in CRPF are considered as the generalized weights. In [46], state estimation method which combines the CRPF with $H_\infty$ method has been proposed for conditionally linear systems with unknown noise statistics.

When only hard bounds of the noises are available for filter design, set-membership theory is a powerful tool for guaranteed estimation, i.e., to find the smallest region in the state space that is guaranteed to enclose the possible states. The set-membership theory and particle filtering method have been blended together in [47] where the significance of each particle is evaluated according to the feasible set given by set-membership theory. For a class of noises with unknown time-varying parameters, the marginalized adaptive particle filtering approach has been studied in [48]. The predictive density of the noise parameters is approximated under the principle of maximum entropy so that the uncertainty is not underestimated. Since the conditional density of noise parameters admits an analytical expression given current states, the marginalization technique is employed in the joint estimation of state and noise parameters.

### Efficient implementation of particle filtering algorithms

At the end of this section, we would like to discuss briefly on the implementation issues of particle filtering. It is expected that the execution time of PFs could be minimized by exploiting the parallel structure inherent in the algorithms and allocating the computational tasks of central unit (CU) to some processing elements (PEs) which run in parallel. As mentioned previously, resampling is the main obstacle to the distributed implementation of PF algorithms since all the particles have to be involved in the resampling step, i.e., it bears no natural concurrency among iterations. Two algorithms for distributing the resampling procedure, namely resampling with proportional allocation (RPA) and resampling with non-proportional allocation (RNA), have been proposed in [49] where the sample space is divided into several groups and each PE is in charge of processing one such group. Since the numbers of particles are distributed unevenly among the PEs, a particle routing scheme is required to define the architecture for exchanging particles among PEs. It is a main focus of [49] to offer a particle routing scheme in which inter-PE communication is deterministic and independent of the CU. Another scheme for distributed implementation of PF algorithms has been proposed in [50], which is based on decomposition of the state space rather than the sample space. In the proposed approach, the original state space is decomposed into two mutually orthogonal subspaces. At each filtering period, samples are drawn sequentially from the two subspaces. Through state decomposition, the original filtering problem is transformed into two nested subproblems each of which corresponds to one of the derived subspaces. The main advantage of such decomposition lies in that part of the resampling procedure can be implemented in parallel, which facilitates more efficient calculation. Note that even though the method in [50] resembles that of marginalized particle filtering, it is applicable to any system with no requirement for a tractable linear substructure.

## Networked systems

### Introduction of networked systems and network-induced phenomena

The development of modern science and technology has given birth to a class of large-scale systems where different components are distributed spatially but work in a collaborative fashion to accomplish certain tasks such as target tracking, environment perception, process monitoring, multi-agent formation control, etc. A key feature of such systems is that all the nodes are connected by a network through which local information is shared. In view of this, such systems are referred to as networked systems to distinguish from the traditional ones. Networked systems have many attractive characteristics such as lower cost, reduced energy consumption, configuration flexibility, enhanced reliability, etc. In target tracking scenarios, a major advantage of distributed sensor nodes is that there are always a portion of nodes close to the target, thus being able to provide measurements with a high signal-to-noise ratio even when low cost sensing devices are employed [51]. In this section, we will mainly focus on state estimation problems for networked systems, i.e., we will study how some specific problems arising in networked systems are treated and how different sensor nodes operate coordinately to provide an accurate estimation of the target state.

We can identify different strategies and architectures to implement state estimation algorithms for networked systems. The simplest idea is to send all the raw measurements obtained at different sensor nodes to a fusion center where these measurements are processed together. This strategy is called a centralized filtering scheme. When Kalman filtering algorithm is adopted in the fusion center to derive the final estimation, we call this scheme *centralized Kalman filter* (CKF). Theoretically, CKF can recover the performance of standard Kalman Filter, i.e., achieve optimal filtering performance in the mean-square sense for linear systems with Gaussian additive noise. This theoretical optimality, however, is based on the ideal assumptions that the fusion center has sufficiently large computation capability and there exists a perfect communication channel between the fusion center and each sensor node, i.e., there are no limitations in data capacity, signal fidelity, transmission rate, sampling rate,

etc. Such conditions are rarely satisfied in most real-world applications. Even if they are satisfied, the filter system is still poor in robustness, i.e., it will crash in the event of fusion center failure. Since the 1970s, distributed estimation schemes, including *distributed Kalman filter* (DKF), have been developed to overcome the drawbacks of centralized filtering schemes. The fundamental idea of distributed estimation is to share the task of data processing among the whole network. In distributed estimation schemes, each node first processes its local measurements and then shares the processed data over the network to derive the global estimate. Distributed estimation has gained increasing concern since it is more robust to node failure, requires moderate communication and allows for parallel processing. Comprehensive review of the distributed estimation approaches could be found in [52,53].

Unlike general state estimation methods, a distinguishing feature of state estimation for networked systems is that the limited capability of communication and local computation has to be taken into consideration in the algorithm design. In most cases, sensor nodes employed in the network are low cost devices (with limited power supply) connected by possibly unreliable channels; so it is unrealistic to expect a perfect communication performance from them. The communication problems, also termed as *network-induced phenomena*, will cause ambiguity and reduce the informativeness of the measurements. As a result, the estimation performance will be degraded to a certain degree. In the following, we will give a brief introduction about several network-induced phenomena that frequently occur in the networked systems.

### Network-induced delay

Time delay is quite common in networked systems where there is limited access to the transmission medium. Time delay in networked systems may result from the following factors:

- Nodal processing: refers to the time required to process local data and reach a routing decision; including data collecting and processing, bit error checking and output link determination;
- Queuing: refers to the time waiting at the output link for transmission; usually depending on the congestion level of router;
- Transmission delay: refers to the time required to push all the bits in packet on the communication medium in use; also known as store and forward delay; primarily due to the limited transmission rate of links;
- Propagation delay: refers to the time required for the bit to reach the target node once it is pushed on to the communication medium; mainly due to the limited travel speed of light in a certain medium.

There are various delay models, including constant delay, random delay which is independent of previous transmission, and random delay with the probability distribution governed by a Markov chain. It is widely understood that time delay can cause performance degradation or even instability to the system. Existing research results have addressed the time delay in networked systems from two aspects: one is to determine an admissible upper bound of the time delay for which a prescribed performance can be guaranteed; the other is to design filtering algorithm for a given time delay to meet certain performance requirements.

### Packet dropout

As another network-induced phenomenon, packet dropout occurs frequently in inter-node communication. Typically the network has its own mechanism to keep the packet dropout at a low level so that the system performance will not be affected. Once the loss rate approach five per cent or higher (the threshold depends on the applications, for real-time ones, it should be lower), the user will begin to notice the presence of communication problems. There are various reasons for packet dropout, including:

- Congestion: in the Internet Standards, congestion and packet loss are treated as synonyms. The packet has to wait in a queue for its turn to be sent when it arrives at an intermediate node on its route. Once the length of the queue exceeds the maximum buffer capacity of the node, some data have to be discarded, which leads to packet loss.
- Bit errors: during the process of data transmission, it is inevitable that some bits will be modified, leading to a mismatch between the value stored in the check bit and the actual checksum. Once the mismatch is detected by the receiving router, this packet will be considered as an erroneous one and hence discarded.
- Limited processing capability: packet dropout will occur when certain local processors (router/switch) are unable to keep up with the speed of data traffic. This is a case of mismatch between communication bandwidth and processing capability.
- Deliberate discard: some routers have packet discard policies that allow them to discard certain type of packets to make room for the ones with higher privilege.

Typically, a Bernoulli distribution is used to describe the randomness of packet loss, where the probability of loss is assumed to be fixed. This assumption can effectively simplify the filter design, and the results obtained have been extended to the cases where packet loss follows a more general distribution. Similarly to time delay, packet dropout can also have adverse effects on state estimation, the severity of which

depend on the loss ratio. Existing studies on packet dropout in networked systems have been focused on the determination of maximum admissible packet loss rate and filter design in the presence of packet dropout. Note that with the maximum admissible loss rate, we can discard some redundant data artificially to save the network bandwidth without violating the performance requirement.

### Quantization

Analog signals have infinitely variable amplitude and therefore have to be quantized before they are transmitted through the network. Quantization is involved in almost all digital signal processing. Examples of quantization processes include rounding and truncation. As a many-to-one mapping, quantization is inherently a lossy process. Considerable research efforts have been devoted to the selection of information that can be discarded without significant loss in performance. The module that realizes the quantization procedure is called quantizer. Existing types of quantizer include logarithmic quantizer and uniform quantizer.

### Signal fading

Another common phenomenon in network communication is that the strength of received signals may vary over time, an effect also referred to as signal fading. It is closely related to multipath, a propagation phenomenon that results in signals reaching the receiving antenna by two or more paths. Multipath propagation can cause fading and phase shifting of the received signals. Movements of transmitter or receiver may also give rise to signal fading. Fading can occur in many forms. When all the frequency components transmitted are attenuated to the same degree, this type of fading is called flat fading. Otherwise we say there is a frequency-selective fading. Existing mathematical descriptions for the channel fading phenomena include analog erasure channel model, Rice fading channel model and Rayleigh channel model [54].

## Existing results on state estimation for networked systems

In this section, we will first review some existing results on state estimation methods designed to address the network-induced phenomena mentioned above, and then proceed to investigate some distributed estimation approaches for networked systems.

### Treatments of network-induced phenomena

*Time delay* Early results on filter design for time delay systems can be found in [55–57]. In [55], the standard Kalman–Bucy filter has been extended to include delayed mea-surements. The ordinary differential equations in Kalman filtering are replaced by partial differential equations together with boundary conditions which may not have an explicit solution. Note that this result is more closely related to the smoothing problem where the past state is estimated using current measurements. In [56], orthogonal projection has been employed to derive the optimal filter for discrete systems with multiple time delays. In [57], the same result as that in [56] has been obtained via maximum likelihood estimation for an augmented state. Despite the straightforwardness, this method is valid only when the random processes considered follow Gaussian distribution.

The $H_\infty$ filter for linear continuous systems with delayed measurements has been provided in [58]. Like $H_\infty$ filter design without delay, the filtering problem is transformed to seeking a bounded solution of a Riccati differential equation. In [59], a less conservative bounded real lemma (BRL) has been employed in the filter design for systems with known state delay to achieve a smaller overdesign. A robust $H_\infty$ filtering approach has been proposed in [60] for a type of uncertain discrete time-delay systems whose parameter matrices are assumed to belong to a convex bounded polytope. Similarly, the filtering method proposed in [61] also addresses parameter uncertainty and time delay, but in this case continuous-time systems are considered and the delay is assumed to be unknown (only the upper bound is available). A similar result has been obtained in [62] where an exponentially stable filter is designed for time-delay systems with norm-bounded parameter uncertainties. Quadratic matrix inequalities are adopted in the analysis and design of the filter. Note that robust filtering approaches in [60–62] have a common presumption that the original system is stable, which may to some extent limit the application scope of the results obtained. As an extension to [59], a robust $H_\infty$ filter has been proposed in [63] to deal with time-varying delay and polytopic type uncertainties using a more efficient BRL. The fact that the BRL is applied to the resulting error system can remove the requirement on a stable system matrix.

Robust filtering for nonlinear time-delay systems have been investigated in [64–66]. In [64], the nonlinearities are assumed to satisfy global Lipschitz conditions. A delay-dependent robust $L_2/L_\infty$ filter has been designed based on linear matrix inequalities (LMIs). In [65], full-order filter has been derived for a general class of nonlinear time-delay systems with guaranteed mean square boundedness of the error dynamics. A general class of nonlinear systems with randomly varying sensor delays have been considered in [66] where conditions for guaranteed $H_\infty$ performance are provided in terms of Hamilton–Jacobi–Isaacs (HJI) inequality.

Recently, extensive studies have been devoted to address the conservatism of $H_\infty$ filtering design due mainly to inequality scaling in the derivation. The conservatism of certain filtering method can be measured by the maximum

admissible delay or the $H_\infty$ performance. In [67], the free-weighing matrix method has been adopted in the $H_\infty$ filter design for systems with time-varying interval delays to reduce conservatism of the existing results. In [68], a novel integral inequality has been used to establish the LMI conditions for the existence of $H_\infty$ filter without resorting to model transformation or bounding technique for cross terms both of which are sources of conservatism.

Recent results on Kalman filtering for time-delay systems can be found in [69–72]. The novel idea in [69] and [70] is to reorganize the measurements from different channels as a delay free system. This reorganized innovation is combined with the orthogonal projection formula to derive the optimal filter. It is shown that for systems with $m$ delays, the obtained solution consists of $m$ standard Kalman filters with the same dimension as the original systems. To reduce the computational complexity of reorganization innovation, the equivalent delay free system has been obtained in [71] by directly solving stochastic equations. The optimal filter and error propagation formula are then derived through Itô differentials of the state expectation conditioned on observation processes. In [72], a new sub-optimal filter has been proposed in the minimum variance framework. In this method, only instantaneous terms are used, thereby avoiding the computation of distributed terms. Also, the filter derived in [72] can be applied to any bounded delay function including non-continuous delays.

*Packet dropout* Filtering problems for networked systems with packet dropout have also received considerable research attention. Many elegant results have been obtained regarding this direction. Due to space limitation, we will only introduce a small portion of them. A more comprehensive review can be found in the excellent survey paper [73]. Intuitively, the missing probability will not affect the boundedness of the error covariance until it reaches a certain critical value. This value is identified in [74] based on novel matrix decomposition techniques. The optimal $H_2$ filtering for networked system with multiple packet dropouts has been considered in [75] where stochastic packed loss is assumed to follow a Bernoulli distribution. The stochastic $H_2$ norm is defined, which generalizes the norm of systems with both deterministic and stochastic inputs. The filter which minimizes such a norm is derived based on the solution of a set of LMIs. The phenomenon of multiple packet dropouts with a more general form of missing probability has been treated in [76] where random measurement loss is allowed to follow any discrete distribution taking values over the interval [0,1] with known occurrence probability. The extended Kalman filter is derived through minimizing the upper bound for the filtering error covariance. For multi-rate sensor fusion with missing measurements, an unknown input observer has been proposed in [99] to minimize the mean square error.

It is known that packet loss can cause instability to the filtering system. Stability analysis of Kalman filter for networked systems with random packet losses has been provided in, to name just a few, [77–81]. It is shown in [77] that there exists a critical value of observation arrival probability under which the expected error covariance is likely to grow unbounded. In [78], necessary and sufficient conditions have been obtained for stability analysis of networked systems with random packet losses characterized by a binary Markov chain. The Markovian packet loss has also been studied in [79] where the notion of stability in stopping times is introduced and its equivalence with the stability in sampling times is established, which simplifies the subsequent stability analysis. The necessary and sufficient conditions for mean square stability have been derived, respectively, for second order systems with different structures and high order systems with a certain structure. In [80], convergence of the error covariance has been analyzed for a rather general class of packet dropping models and alternative performance measures are introduced when the expectation of error covariance cannot be well defined. The asymptotic behavior of the random Riccati equations (RRE) which describe the evolution of estimation error covariance in Kalman filter has been studied in [81] where sufficient conditions for the existence and uniqueness of the invariant distribution for the RRE are derived.

*Quantization* State estimation for systems with quantized measurements has been studied in [82] where the quantizer and the estimator are designed jointly. A logarithmic quantizer is employed whose resulting quantization error can be regarded as a multiplicative noise. Choices of quantization density or number of quantization levels are discussed. Furthermore, a dynamic scaling parameter for the quantizer is introduced to ensure the convergence of estimation error in unstable systems. In [83], a quantized filtering scheme using decentralized Kalman filter has been proposed for linear discrete systems with multiple sensors. The innovation process of each local sensor, instead of local estimation, is quantized to avoid saturation of the quantizer. It is proved that stability of the filter can be achieved under sufficiently high bit rate even for unstable systems. The trade-off between quantization rate and state estimation error is analyzed. The problem of rate allocation among different sensors is also considered to enhance the asymptotic behavior of estimation error. The quantized gossip-based interactive Kalman filtering approach has been studied in [84] where it is proved that the error covariance sequence at a randomly selected sensor can converge weakly to a unique invariant measure even with information loss caused by quantization. In [85], a recursive filter is designed for power systems with quantized nonlinear measurements by minimizing the upper bound of the error covariance.

*Signal fading* Kalman filter for networked systems where local measurements are sent to the fusion center via fading wireless channels has been investigated in [86]. The expected error covariance of Kalman filter is proved to be bounded and converge to a steady state value. Exact recursive formulas are provided to calculate the upper bounds of error covariance which may serve as an alternative index to be optimized when the expression of error covariance is unavailable. Kalman filter with measurements transmitted over fading channels has been considered in [87] where it is assumed that the transmission power can be adjusted to alleviate the effects of fading channels. Sufficient conditions are obtained to ensure the boundedness of error covariance. These conditions are then used for power allocation to minimize the total power consumed by the network. In [88], the envelop-constrained $H_\infty$ filter has been presented for a class of time-varying discrete systems. The finite horizon case is considered and a novel envelope-constrained performance criterion is proposed to define transition performance of error dynamics. Borrowing idea from the set-membership filtering method, an ellipsoid description of estimation error has been utilized in [88] to transform the envelop constraints into a set of matrix inequalities solvable using standard software package.

Apparently, the network-induced phenomena mentioned above will coexist in a system. A large number of papers have been published on filter design for systems with multiple network-induced phenomena. For example, the simultaneous presence of time delay and packet loss has been addressed in [89–91]; both quantization and packet dropout have been taken into consideration in [92]; a filtering scheme has been recently proposed in [93] which is robust against both channel fadings and gain variations. For more related results, the readers are referred to [94–101].

### Distributed estimation for networked systems

Another research direction of state estimation theory for networked systems is distributed estimation methods which aim to maintain an accurate estimation of certain network states at each sensor node using measurements from all sensor nodes in the network. Besides estimation accuracy, communication overhead and computational complexity also pose constraints on the filtering scheme to be designed due to the limited bandwidth and power supply of the network. Early treatments to the distributed estimation problem can be found in [102–104]. In [102], both measurements and local estimates are shared among neighboring nodes. It is shown that asymptotic agreement on the estimates can be achieved through infinitely frequent data exchange among sensors which form a communication ring. In [103], sufficient statistics are extracted from local measurements and transmitted to a fusion center where the centralized

conditional distribution is exactly reconstructed. Sufficient conditions have been presented in [104] under which global sufficient statistics can be expressed as a function of the local ones.

For distributed estimation, consensus is an important concept and a lot of research efforts have been devoted to it. Generally, consensus refers to agreement among all the members of a group. In the specific case of state estimation for networked systems, we say the network reaches consensus if all the nodes hold an identical estimate of a certain quantity of interest. As a fully distributed framework, consensus based distributed estimation allows for cooperation over the network without the participation of a fusion center, thereby avoiding over reliance on a certain node. Various forms of consensus algorithms have been developed to establish the rule following which inter-node communications are implemented to reach an agreement among nodes. Average consensus and gossip consensus are the two major consensus strategies that have been investigated extensively in recent years. Readers are referred to [105] to obtain a full understanding of consensus algorithm for networked systems.

A fully distributed Kalman filter has been proposed in [106] for sparsely connected, large-scale systems. The global dynamic model is decomposed into low-dimensional subsystems for which local filters are designed. These subsystems overlap and the common states shared by several nodes are estimated using fusion algorithm. The centralized error covariance is derived through a distributed computation algorithm for matrix inversion, called distributed iterate-collapse inversion algorithm, which assimilates local error covariances with computation complexity independent of the system dimension. In the proposed method, each sensor node only needs to deal with a portion of the system state, which has significantly reduced the communication and computation requirements.

In [107], the authors consider a two-stage distributed Kalman filter which consists of a measurement update step and a fusion step using consensus algorithm. The interaction between the filter gain, the consensus matrix and the number of communications is analyzed in depth. It is proved that the common practice of minimizing the spectral radius of consensus matrix for fastest convergence is not necessarily the optimal strategy when only a small number of communications are available between two consecutive samples. It is also shown that the joint optimization of filter gain and consensus matrix is non convex and can be analytically characterized only in some special cases.

A similar two-stage distributed filter has been investigated in [108]. Sufficient conditions have been obtained to judge the distributed detectability of the networked system, i.e., the existence of filter gains which ensure an asymptotically stable error dynamic given a specific choice of consensus weights.

A sub-optimal filtering scheme is then developed through minimizing an upper bound on a quadratic cost, and convergence analysis has been carried out for the time-invariant case.

One significant feature of the two-stage filtering schemes discussed above is that the consensus communication occurs at a much shorter time scale than the operation of local filters, i.e., it is assumed that there is sufficient time for the network to achieve consensus through intensive inter-node communications before the arrival of the next observation. This assumption, however, does not apply to the cases where there is a fast target dynamic and/or a high sampling rate. Besides, the high rate of consensus communication has blurred the line between distributed estimation and centralized estimation, as argued in [110]. To address this problem, Kar et al. proposes the gossip interactive Kalman filtering (GIKF) where consensus communication and observations take place at the same time scale. As a communication protocol inspired by social network phenomena, the gossip protocol has found broad applications, especially in networks with large scale or inconvenient structures. Readers can refer to [109] for a detailed overview of recent studies on gossip algorithms. In [110], the convergence of the GIKF scheme has been analyzed and the error covariance is shown to evolve according to a switched system of random Riccati operators where the switching is governed by a Markov chain determined by the network topology. Stochastic boundedness of estimation error and weak consensus of the error covariance have been established under weak assumptions about detectability and connectivity of the networked system. The method in [110] requires transmission of the error covariance, which may be burdensome for high-dimensional systems. An alternative approach has been given in [111] based on dynamic consensus on pseudo-observations (DCPO). The network tracking capacity (NTC) is used to characterize the influence of network topology and observation models on the stabilizability of DCPO error process. An explicit expression of the NTC is derived and asymptotic stability of DCPO error dynamics is established. The averaged pseudo-observations obtained are then used to construct local filters whose gains are designed to minimize the mean square error (MSE). It is shown that the method in [111] can achieve lower MSE while maintaining the major advantage of GIKF, i.e., the inter-node communication occurs no more frequently than sensor sampling.

Consensus-based distributed least square (LS) estimation problems have been studied in [112,113]. The authors of [112] consider the total least square (TLS) estimation for overdetermined systems where both the input data matrix and the data vector are assumed to be noisy. A semi-definite relaxation technique is used to transform the nonconvex TLS problem into an equivalent convex semidefinite program (SDP). At this point, the dual based subgradient algorithm (DBSA) can be used to solve the distributed TLS problem

without reliance on the computationally expensive SDP optimization procedure. In [113], underdetermined least square estimation problem has been considered. The requirement for consensus is expressed as constraints where an auxiliary variable is introduced to facilitate parallel processing, and the resulting constrained optimization problem is solved under the Augmented Lagrangian framework.

Distributed estimation schemes based on consensus algorithms aim to reach an agreement among all the nodes in the network. However, the ultimate purpose of state estimation problems is to achieve at each node an estimate that minimizes a predefined cost function, which does not necessarily require that all nodes provide the same result. Moreover, it is shown in [115] that the consensus network can become unstable even if all the local filters are stable, i.e., cooperation by means of consensus algorithms may lead to disastrous consequences. Motivated by such observations, the estimation schemes based on diffusion strategies have been proposed. As another class of fully distributed estimation methods, diffusion algorithms make several key improvements upon the consensus ones, among which the fundamental one is that agreement is no longer the goal. In the diffusion Kalman filtering proposed in [114], each node first adapt its local estimate using measurements from neighboring sensors, obtaining an intermediate estimate, which is refer to the incremental update step. Then a diffusion update step is performed by combining (through calculating a weighted average) the intermediate estimates received from neighboring nodes. Diffusion filter belongs to the single time-scale estimation scheme, i.e., the communication requirement of distributed filter is comparable to that of the gossip filter, but diffusion networks can achieve faster convergence rate and lower MSE than consensus networks. In addition, it is proved in [115] that the stability of local filters is sufficient to guarantee global stability of network under the diffusion framework, regardless of the choice of combination weights.

## Particle filter in networked systems

In the previous sections, we have investigated the particle filtering methods and state estimation for networked systems, respectively. In the cases where nonlinear systems or non-Gaussian PDFs are addressed, particle filtering becomes a more suitable choice. The applications of PF to networked systems, however, have given rise to some new challenges. This is mainly because in particle filtering methods, a large amount of particles are required to represent the posterior PDF, which implies a huge communication burden if local information is exchanged among nodes. One major concern of particle filtering for networked systems is, therefore, how to achieve an affordable communication cost while

maintaining an acceptable accuracy. On the other hand, the network-induced phenomena should also be addressed when PF is applied to networked systems. In this section, we will discuss the applications of PF algorithms to the filtering problems of networked systems, highlighting how the network-induced phenomena are treated by particle filters and how, in networked systems, local particle filters work coordinately to accomplish the state estimation task with an affordable communication cost.

## Particle filter with network-induced phenomena

### Particle filtering for systems with missing data

Generally speaking, there are two categories of methods for missing data problems: guaranteed cost ones and data imputation based ones. The methods introduced in Sect. 2 obviously belong to the former category. Next we will introduce some data imputation based approaches which are more commonly used in Monte Carlo methods. The basic idea of data imputation is to generate some data which obey the same distribution as the missing ones. These artificially generated data will then be regarded as real measurements in the subsequent processing. It is typically difficult to calculate the distribution of missing data conditioned on the observed ones. This being the case, one has to resort to sampling-based approaches such as the MCMC method introduced in Sect. 2 of this paper.

As a special form of the MCMC approach, Gibbs Sampler has attracted extensive attention among researchers since its introduction in [116]. Essentially, Gibbs sampler is an iterative algorithm to draw samples from an unknown distribution without direct calculation of the density. The pseudo code of Gibbs Sampler is given in Algorithm 3 where the case of three random variables, $x$, $y$ and $z$, is considered and the marginal distribution $p(x)$ is of interest to us. The readers can also refer to [117] for an explanation of the underlying principle of Gibbs sampler.

---

**Algorithm 3** Gibbs Sampler

---

**for** $n = 1, ..., N$ **do**
  Initialize $\{x_0^{(n)}, y_0^{(n)}\}$.
  **for** $k = 0, ..., M - 1$ **do**
    Sample $z_k^{(n)} \sim p(z|x_k^{(n)}, y_k^{(n)})$.
    Sample $x_{k+1}^{(n)} \sim p(x|y_k^{(n)}, z_k^{(n)})$.
    Sample $y_{k+1}^{(n)} \sim p(y|z_k^{(n)}, x_{k+1}^{(n)})$.
  **end for**
**end for**
Then $\{x_M^{(i)}, i = 1, ..., N\}$ can be viewed as $N$ samples drawn from $p(x)$.

---

Kong et al. extended the Gibbs sampler to sequential imputation which does not require iterations, thus reducing the computational burden [18]. The sequential imputation uses samples and associated weights to approximate the unknown distribution in the presence of missing data, and thus can be seen as a combination of Gibbs sampler and sequential importance sampling. As new data arrive, a new sample is drawn and the augmented data set is updated to include this sample. The corresponding weight is then determined according to the quality of the augmented data set. Some related problems, such as effective sample size, the order of imputation, the behavior of weights, and sensitivity to the choice of prior distribution, are also analyzed in detail. An interesting finding, which is also illustrated by simulation results, is that the order of imputation can have a huge impact on the approximation accuracy. To be brief, a "good" data set cannot play its due role if it is processed after the "bad" ones because the trail distribution has already been corrupted by early imputations.

Expectation Maximization (EM) algorithm is another powerful tool to address missing data problems. An excellent introduction of it can be found in [118]. The EM algorithm consists of an expectation (E) step and a maximization (M) step. These two steps are executed alternately: in the E step, expected values of the missing data are computed, which will then be used in the M step; while in the M step, the maximum likelihood estimate of unknown parameters are calculated, to be used in the next E step. It is guaranteed that the likelihood function will not decrease after every such iteration. This implies that only local maximum can be reached, which may be seen as a deficiency of the EM algorithm. The basic EM procedure for estimating unknown parameter $\theta$ in the presence of missing data $z_k$ is illustrated in Fig. 1 below.

In [119], three EM algorithms have been adopted to treat the maximum a posteriori (MAP) state estimation for JMLS. Both the Markov chain and continuous states are unknown and to be determined. The first algorithm addresses MAP estimation of the Markov chain. The unknown continuous states are regarded as missing data and estimated with a fixed-interval Kalman smoother in the E step. Assuming the continuous states to be known, the MAP estimation of Markov chain is then obtained through dynamic programming in the M step. The second algorithm aims to estimate the continuous states with the unknown Markov chain viewed as missing data. A forward and backward recursion is applied in the E step to calculate the probability of Markov chain and a Kalman smoother is used in the M step to compute MAP estimation of the continuous states. The last algorithm deals with joint estimation of Markov chain and continuous states, which is realized through an alternate execution of fixed-interval Kalman smoother and dynamic programming. To overcome local convergence of the EM method, stochastic sampling based algorithms have been proposed in [120] for state estimation of JMLS. Three data augmentation (DA) algorithms (DA, stochastic annealing DA, and Metropolis-
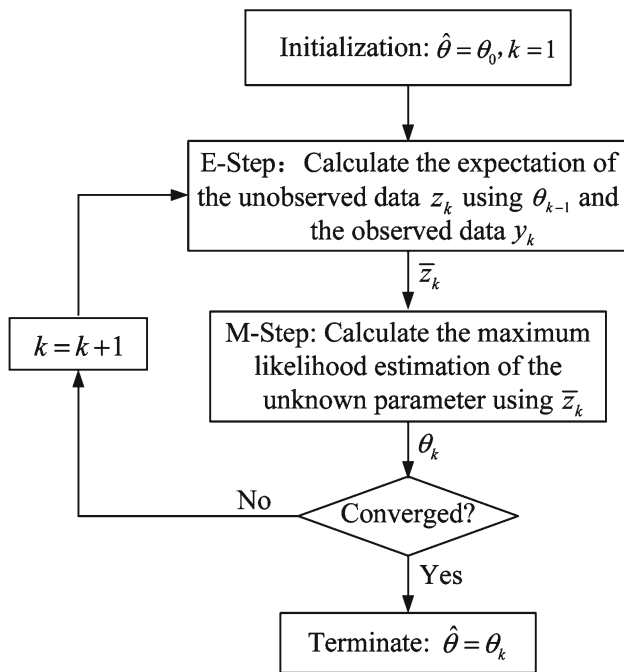
**Fig. 1** Basic EM procedure for parameter estimation

Hasting DA) are employed and an acceptable computational cost is achieved. As a special form of MCMC method, the DA algorithm can ensure convergence to the globally optimal solution. In [121], the EM algorithm is applied to estimate missing data in the Moderate Resolution Imaging Spectroradiometer (MODIS) time series for forest growth prediction.

Multiple imputation particle filter (MIPF) has been introduced in [122] where particle approximation is utilized to perform multiple imputation. This method resembles both the Gibbs sampler and EM algorithm. First, multiple imputations are drawn from a proposal distribution where the true states are replaced with particle representation which is calculated regardless of the missing observations. Next, for each imputation, a particle filter is performed to obtain an approximation of the state posterior. We then combine the approximations derived from different particle filters to give the final particle representation of the target density. Almost sure convergence of the MIPF method is established in a later work [123].

*Particle filtering for time-delay systems*

As mentioned in Sect. 2, transmission delay occurs frequently in a networked environment, which gives rise to the so-called out-of-sequence measurements (OOSMs). A great deal of research has been done to address this phenomenon, but mostly within the framework of Kalman filtering. The particle filtering algorithm in the presence of OOSMs has been studied in [124], where the basic idea is to rerun the particle filter to incorporate the delayed measurements. A major

drawback of this method is that we have to store the particles and corresponding weights at each sampling period, which poses severe challenges to the storage capability of the processor, especially when the required number of particles is large. The proposed method in [124] also suffers from the problem particle degeneracy, but this can be mitigated via an MCMC step [125]. In [126,127], a backward information filter has been adopted to retrodict particles corresponding to the delayed measurements. These particles are then used to recalculate the current weights. The implementation of backward information filter, however, involves intensive computation, which may be formidable in some practical applications.

The above-mentioned methods suffer from either excessive memory requirement or huge computational burden. A storage efficient particle filter has been proposed in [128] where only mean and covariance of the particle set need to be stored. The memory requirement of this method is dramatically decreased compared with that of [124], but the problem of particle degeneracy remains, especially when the OOSMs contain such a great amount of information that the original support is not enough to describe the filtering distribution. A selective procedure has been proposed in [129] to distinguish the OOSMs according to their utility. At every sampling period, a threshold for measurement selection is firstly calculated through a constrained optimization problem. The measurements whose utility exceeds the threshold are identified as informative and processed in the subsequent filtering step, while those with utility below the threshold are simply discarded. To reduce computational cost associated with solving the optimization problem, Gaussian approximation and linearization are employed for a rapid prediction of the mean square error (MSE) reduction brought by each delayed measurement. In addition, another threshold test is conducted to detect particle degeneracy. Once the effective sample size is found to drop dramatically, which implies that current support can no longer give an accurate description of the filtering distribution, the OOSMs are reprocessed through another filtering procedure which allows for simultaneous adjustment of the weights and locations of particles, thus avoiding particle degeneracy.

The exact Bayesian solution to the filtering problem for OOSMs has been derived in [130]. Different from the storage efficient particle filter [128] whose performance degrades when the target states do not follow a unimodal distribution, the exact Bayesian algorithm uses all the past particles and weights to achieve optimal performance. The cost of optimality, however, is a huge computational overhead, which limits the application scope of the exact Bayesian algorithm to low-dimensional problems or offline processing. For nonlinear model containing a linear substructure, two Rao-Blackwellized particle filtering algorithms have been

presented in [131] to yield efficient execution and high accuracy, respectively.

Particle filtering problem for target tracking in the presence of signal propagation delay has been considered in [132]. Due to the interaction between target dynamics and propagation delay model, neither kinematic state of the target nor the propagation delay can be determined independently, which substantially complicates the problem. To tackle this difficulty, an augmented state vector is defined which includes both the time delay and the target state with a stochastic time stamp. The key of this method is to solve the unknown delay from an implicit equation. It is shown that iterative techniques can be used to obtain an approximate solution to the implicit equation as long as a fairly weak convergence condition is satisfied. The bootstrap particle filter is employed where iterations are incorporated in the time-update step to predict the time delay of current measurements. The resultant particles have different time stamps with each other, therefore a time synchronization is performed before the final estimate is derived.

Particle filtering algorithm with one-step delayed measurements has been studied in [133]. The standard particle filtering algorithm is modified to take the probable delay into account. When the latency probability is unknown to the designer, a maximum likelihood algorithm is proposed to identify it. The result has been extended to handle multiple-step randomly delayed measurements in [134], but only the case of known latency probability is considered.

*Particle filtering for systems with signal quantization*

For estimation problems with quantized measurements, the Cramer-Rao lower bound (CRLB) has been derived in [39], giving an indication on the information loss caused by signal quantization. Both Kalman filtering and particle filtering algorithms are developed to handle measurement quantization, and the superiority of particle filtering over Kalman filtering is demonstrated through numerical experiments. Measurement quantization will induce big error to the filter system when the values of observed data are large. The filtering problem with innovation quantization has been studied in [135]. A counterexample is constructed to show that the Kalman filtering may perform below expectation or even diverge in the presence of quantized innovation. The particle filter, instead, seems capable of approximating the optimal filter in the same situation.

It is revealed in [136] that the state conditional on quantized observations can be decomposed into the sum of two independent random variables, one of whom follows Gaussian distribution while the other is a truncated Gaussian random vector. The authors of [136] point out that we only need to propagate the truncated Gaussian variable, rather than the sum, since the truncated Gaussian variable has a probabil-
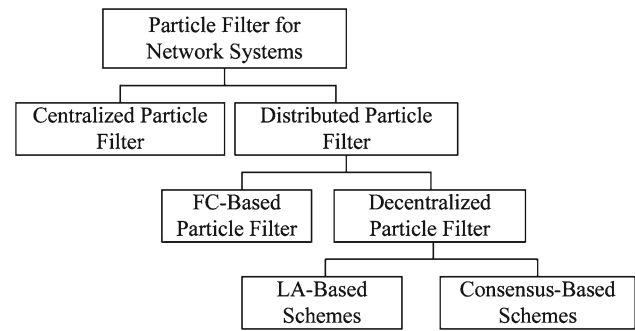


**Fig. 2** Taxonomy of PF for networked systems

ity density whose covariance is much smaller than that of the conditional state density. Taking advantage of the Gaussian properties, the authors design a Kalman-like particle filter (KLPF) where a group of Kalman filters are processing in parallel to obtain minimum mean square estimate of the state conditioned on perfect observations. One major advantage of KLPF is that the required number of particles is dramatically reduced compared with directly using particle filter as in [135].

### Particle filter for cooperative estimation in networked systems

As state estimation problem for networked systems has gained increased research attention, a great variety of particle filtering schemes for networked systems have been published in the literature [137]. We can, similarly to that of Sect. 2, define centralized particle filtering (CPF) and distributed particle filtering (DPF) according to whether local processing is performed at each agent. It should be noted that the CPF method is nothing different from the general form of particle filtering introduced in Sect. 1, and has the similar shortcomings as the CKF discussed in the previous section. Therefore, we will present a brief review of the existing CPF methods in the next paragraph and reserve most of our attention for the discussion of DPF methods. For clarity, we first present a taxonomy of different particle filtering approaches for networked systems in Fig. 2.

*Centralized particle filtering*

Theoretically, the CPF can give the optimal state estimate if we ignore the error caused by particle representation of the continuous posterior distribution. The optimality, however, comes with a high cost, both in communication burden and computation complexity. The requirement on communication could be formidable in sensor networks where each node has limited power supply and thus limited communication capability. In [138,139], a CPF method based on state partition and parallel EKF has been proposed for tar-

get tracking using collaborative acoustic sensors. The major computation task is efficiently done in the fusion center, thus freeing sensor nodes from local data processing. To obtain a proposal distribution which is closer to the state posterior, a bank of EKFs are used to process data from all activated sensors concurrently, and a weight sum of these EKF estimations is calculated and taken as the proposal distribution. An efficient way to store particles has been introduced in [140] where a compression step is taken before storing particle states. Simulation results suggest that this scheme can significantly reduce memory requirement with minimal performance loss.

In a sensor network where the communication capability is limited, data are always quantized at each sensor node before transmission. This, plus the imperfect nature of the communication channels, should be taken into account by the fusion center to gain better performance. A channel-aware particle filtering scheme is put forward in [141] to address the quantized measurements and fading channels simultaneously. The likelihood function, in which both data quantization and channel imperfection are considered, is calculated in three different scenarios. The posterior CRLBs for the proposed method are also derived. When there is a constraint on the total number of bits that can be transmitted, bit allocation becomes necessary. In [142], a dynamic bit allocation algorithm based on approximate dynamic programming has been presented. It is shown that the proposed algorithm can save much of the computational cost while achieving comparable accuracy with other existing methods. The amount of transmitted data can be significantly reduced if sensor nodes are able to distinguish informative measurements from uninformative ones. The data censoring performed at each sensor node does exactly this. The particle filtering with censored data has been studied in [143] where it is pointed out that even though uninformative measurements are not to be transmitted to the fusion center, the fact that they are uninformative also delivers some useful information for data processing. This information is exploited in the proposed filtering method through a full likelihood function, which contributes to an enhanced performance. Strictly speaking, the method in [143] does not belong to CPF since KF update is run at each node to obtain the variance of local innovation based on which the censoring threshold is identified. However, we introduce it here because, like other CPF methods, it requires transmission of raw measurements to eliminate the dependence between local data.

### Distributed particle filtering

In the remainder of this section, we will focus on DPF for state estimation in agent networks. Hlinka et al. presented a detailed classification of the existing DPF methods (see [144]). A fundamental distinction between different DPF methods is whether a fusion center (FC) is present. In the FC-based DPF scheme, each agent processes its own measurements with a local PF and reports the obtained posterior to the FC according to a predefined communication protocol [145,146]. This scheme is suitable for those applications where global knowledge is required only in the FC [147]. However, it suffers from two major drawbacks: (1) the filtering performance relies highly on the FC, which implies a poor robustness against the FC faults; (2) the communication path is highly dependent on the network topology, once the topology changes, which is very common in mobile networks, the total route table has to be re-established; (3) excessive communication burden is imposed on the nodes which are closer to the FC. To reduce long-distance communication, a two-tiered network structure has been proposed in [148] where some selected nodes, referred to as cluster heads (CHs), are responsible for processing raw measurements of the nearby sensors and sending the obtained local estimate to the FC for a further fusion. In this way, only the CHs are required to be capable of directly communicating with the FC.

DPF schemes without an FC is also referred to as *decentralized particle filtering*. We can further classify various decentralized particle filtering methods according to whether all the agents run the particle filter simultaneously or only a portion of them are in charge of data processing. We refer to the schemes where a portion of activated agents take the charge of global estimation as *leader agent (LA)-based DPF* (see, for example, [149–151]), and term those with all agents performing particle filtering algorithm *consensus-based DPF* (see, for example, [156,158,164]). In the LA-based schemes, a sequence of adjacent nodes form a LA path along which the local estimation is accumulated. Typically, this LA path is constructed dynamically and adaptively, i.e., current LA is in charge of selecting the next LA among its neighbors based on the assessment of their informativeness. The selection scheme can affect the estimation accuracy and energy usage to a large degree [152]. Compared with the LA-based schemes, consensus-based DPFs can achieve enhanced scalability and robustness against changes in network topology or node failures. The price for these advantages is a heavier demand on communication and the likely delay due to consensus iterations. In view of the fact that a detailed introduction of both LA-based and consensus-based schemes has already been provided in [144], we will, in the following, investigate from a different perspective, i.e., we will focus on how different decentralized particle filtering methods make a reasonable trade-off between multiple performance index including accuracy, communication burden and computation complexity.

Ideally, we hope the performance of decentralized particle filtering methods can reach that of the centralized ones which is theoretically optimal. Decentralized particle filtering for blind equalization has been studied in [153] where

each node evaluates the likelihood function of its local observations and then broadcasts it to the entire network. The local PF performed at each node thereby has access to the global likelihood function, and is guaranteed to converge to the optimal filter asymptotically. Synchronization is required to ensure that an identical set of particles are generated at different nodes. It is also shown that the filtering performance could be enhanced via the optimal importance function, which, however, asks for an extra amount of broadcast. A modified method has been proposed in [154] to reduce inter-node communication by employing parametric approximations of the remote likelihood function. This method achieves significant communication reduction with only moderate performance degradation. The communication requirement could be further cut down through a protocol where inter-node connection for message exchange is established randomly and each node transmits its local data to only one remote sensor at each time step [155].

The methods mentioned in the previous paragraph rely on broadcasting and are thus suitable only for fully connected networks. Two consensus-based methods have been provided in [155] where inter-node communications are limited within adjacent nodes. Both methods involve evaluating the global likelihood function. The first one uses average consensus to calculate the global likelihood at each node and a quantization step to eliminate the discrepancy between particle sets at different nodes. To avoid performance degradation caused by quantization, the second method adopts a modified minimum consensus algorithm, which borrows the idea of flooding scheme, to obtain an ordered list of local likelihood functions shared by all the sensor nodes. The merit of this method lies in that it, unlike the first one, does not require an infinite number of consensus iterations for a guaranteed performance.

The consensus-based schemes reduce inter-node communication at the cost of heavier local computation. This is desirable in most applications since communication between nodes typically consumes more energy than local computation. Similar ideas have been discussed in [156] where a likelihood consensus (LC)-based DPF method is developed. The basic idea of LC-based DPF is that the underlying sufficient statistics, rather than raw measurements, should be exploited in the DPF to avoid redundant communication. A key step in the proposed algorithm is to obtain a parametric representation of the local likelihood functions. Specifically, local likelihood functions are approximated by the sum of a series of basis functions multiplied by respective coefficients. In this way, the useful information contained in local measurements is compressed into certain coefficients. Since the basis functions are known to all the nodes, the knowledge of corresponding coefficients is sufficient to reconstruct the likelihood functions. With this approximation, a consensus procedure is carried out to calculate the sum of the exponential term of global likelihood function which can be expressed as the product of all the local likelihood functions. The pseudo code of LC-based DPF is presented in Algorithm 4. An attractive feature of the LC-based DPF method is that its communication cost does not depend on measurement dimensions, which makes it particularly suitable in applications which involve high-dimensional measurements. Also note that synchronization is no longer a requirement in the LC-based method since the likelihood functions exchanged between nodes are in a parametric form rather than represented by discrete values.

---

**Algorithm 4** LC-based DPF

At time $k (k = 1, 2, ...)$, sensor node $j$ performs the following steps:

- Resample: for $i = 1, ..., N_s$, sample $\tilde{x}_{k-1,j}^i \sim \sum_{i=1}^{N_s} w_{k-1,j}^i \delta(x_{k-1,j} - x_{k-1,j}^i)$ and associate each $\tilde{x}_{n-1,k}^i$ with weight $1/N_s$.
- Prediction: for $i = 1, ..., N_s$, sample $x_{k,j}^i \sim p(x_{k,j}|\tilde{x}_{k-1,j}^i)$.
- Approximation: for $i = 1, ..., N_s$, use basis expansion to obtain a parametric representation of the local likelihood function $p(z_{k,j}|x_{k,j})$, see [156] for more details.
- Consensus: for $i = 1, ..., N_s$, perform average consensus to obtain the approximation of the global likelihood function $p(z_k|x_{k,j})$.
- Update: for $i = 1, ..., N_s$, associate $x_{k,j}^i$ with normalized weight $w_{k,j}^i \propto p(z_k|x_{k,j}^i)$.
- Estimation: approximate the global minimum MSE estimation of $x_k$ by $\hat{x}_k = \sum_{i=1}^{N_s} w_{k,j}^i \delta(x_k - x_{k,j}^i)$.

---

One shortcoming of the method proposed in [156] is that current measurements have not been incorporated in the proposal density. This problem has been treated in [157] via proposal adaptation which is implemented in a distributed manner. Gaussian distribution is used to approximate both local and global posteriors, and EKF/UKF is employed to incorporate local measurements in the local posterior. Fusing local information to obtain a global proposal density now amounts to running consensus algorithms to calculate two sums, for global mean and global covariance, respectively, over all sensor nodes. The benefit gained from the adapted proposal density is twofold: on the one hand, particles are used with higher efficiency since they are located in a region of higher likelihood; on the other hand, the least square approximation in the likelihood consensus will also have an improved accuracy since the local likelihood functions are approximated in a smaller region.

Another efficient way to implement proposal adaptation has been proposed in [158] where a set-membership constrained particle filtering scheme is developed. It is argued that the importance sampling step can be implemented in a distributed fashion if particles are drawn from the prior dis-

tribution. The prior distribution, however, is a non-adapted one which has not taken current measurements into account. This implies that we have to use a large number of particles to ensure that the posterior density is well characterized by these particles because it is possible that the likelihood function is a very peaked one compared with the prior PDF. A large number of particles will in turn reduce the efficiency of DPF algorithm because the computational complexity is positively correlated with the number of particles in a consensus-based scheme. To overcome this dilemma, we seek to develop a proposal adaptation scheme with affordable distributed implementation. The set-membership based adaptation proposed in [158] does exactly this. In this method, each sensor first determines a local set which approximates the local posterior density. Local sets at different nodes are then combined using consensus algorithms to construct the global set which will serve as the global proposal density in the subsequent importance sampling. The consensus algorithms are guaranteed to converge in finite iterations, which further reduces the overall communication cost. The pseudo code of set-membership constrained DPF approach is presented in Algorithm 5.

---

**Algorithm 5** Set-membership Constrained DPF

---

At time $k(k = 1, 2, ...)$, sensor node $j$ performs the following steps:

- Local set calculation: run local particle filter to obtain a particle representation of the local posterior $\hat{p}_{k,j}(x_k|z_{1:k-1}, z_{k,j}) = \frac{1}{N_s} \sum_{i=1}^{N_s} \delta(x_k - \tilde{x}_{k,j}^i)$, and construct local set as $\varepsilon_{k,j} = \{x : \min_i \tilde{x}_{k,j}^i \preceq x \preceq \max_i \tilde{x}_{k,j}^i\}$ where the inequalities ($\preceq$) are component-wise.
- Global set determination: perform max-consensus to construct the global set $\varepsilon_k$ as the intersection of local sets.
- Prediction: for $i = 1, ..., N_s$, sample $x_{k,j}^i \sim p_{\varepsilon_k}(x_{k,j}|x_{k-1,j}^i) = \frac{\alpha_j \mathbf{1}_{\{x_{k,j} \in \varepsilon_k\}} + \beta_j \mathbf{1}_{\{x_{k,j} \in \varepsilon_k\}}}{c_j} p(x_{k,j}|x_{k-1,j}^i)$ where $\alpha_j \gg \beta_j$ and $c_j$ is a normalization constant. Also, note that an identical set of particles are generated at different nodes, i.e., $x_{k,j}^i = x_{k,m}^i$ for $j \neq m$.
- Consensus: for $i = 1, ..., N_s$, perform average consensus to share the global likelihood function $p(z_k|x_{k,j}^i)$ across the whole network.
- Update: for $i = 1, ..., N_s$, associate $x_{k,j}^i$ with normalized weight $w_{k,j}^i \propto p(z_k|x_{k,j}^i)$.
- Estimation: approximate the global minimum MSE estimation of $x_k$ by $\hat{x}_k = \sum_{i=1}^{N_s} w_{k,j}^i \delta(x_k - x_{k,j}^i)$.

---

In [159], a Gaussian mixture model (GMM) has been employed to represent the posterior PDF to circumvent the transmission of a huge number of particles. In this method, each node obtains the global statistics through an average consensus which diffuses local statistics over the network. Based on the global statistics, an EM algorithm is performed to estimate the global GMM (see also [160]). The transmis-

sion of GMM representations, however, can be inefficient for high-dimensional systems since the amount of transmitted data grows with the volume of covariance matrix which is proportional to the square of state dimension. To achieve better scalability, a Markov chain distributed particle filter (MCDPF) has been proposed in [161,162] based on random walks on the graph. In the MCDPF method, particles traverse the network along a random path and update the associated weights according to the local measurements at each node they pass by. The communication overhead of MCDPF approach depends linearly on the state dimension, which is particularly suitable for high-dimensional systems. Note that although data exchange is limited within neighboring nodes, the MCDPF does not belong to the consensus-based approaches since no consensus iterations are required. However, convergence to the optimal filter can only be established when the number of particles and the length of the random path both goes to infinity. It is also pointed out in [162] that, for low dimensional systems, the MCDPF algorithm is inefficient and GMM representation may be a better choice. Therefore, one needs to select the most suitable scheme on a case-by-case basis.

Gaussian mixture model has also been employed in [163] to develop a soft-data-constrained DPF. In this method, the global GMM is calculated from local ones using the consensus propagation algorithm. Instead of representing the global posterior from which the new sets of particles are drawn, the global GMM is used to pose soft-data-constraints according to which the local particles are reweighed. The resultant local particles at each node represent a local posterior closer to the global one, which implies an enhanced robust against noise and failures.

Up to now, the consensus-based DPF methods mentioned have a basic requirement that consensus be achieved before the arrival of next measurement. In a network with intermittent communication connectivity, however, convergence of the consensus algorithm between every two consecutive measurements cannot be guaranteed, which may lead to severe performance degradation. A consensus/fusion based DPF method has been presented in [164] to address this problem. In this method, an extra filter, referred to as the fusion filter, is employed at each node in addition to the local particle filter to diffuse local estimate and reach consensus across the entire network. Note that the fusion filter is allowed to run at a different rate from the local one, thereby removing the requirement on convergence between successive measurements. Another function of the fusion filter is to compensate for the common information contained in different local estimates which is a general problem for DPF schemes where local estimates rather than raw measurements are diffused [165].

A constrained sufficient statistic-(CSS) based DPF method has been provided in [166]. Similarly to the LC-based approach [156], this method also seeks to fuse local sufficient

statistics (LSS) to the global one. However, no approximation of the global sufficient statistics (GSS) is involved, which implies an enhanced accuracy. Communication overhead per iteration is no longer dependent on the state dimensions (as is the case with [159, 167]) or the number of particles (as is the case with [158]). It is, instead, proportional to the number of GSS parameters which is much lower compared with either scenario mentioned above. To adapt the proposed method to error prone networks with intermittent connectivity, the authors of [166] further combine the CSS based DPF with distributed unscented particle filtering (DUPF) to achieve a guaranteed performance with fewer number of iterations per consensus run.

## Conclusion and outlook

In this survey, we have reviewed existing results on particle filter and its applications in networked systems. As a simulation-based method, particle filter has particular advantages in complex systems where nonlinearities and non-Gaussian noises are ubiquitous. It can be seen that the application of particle filter is still limited by the hardware resources, therefore, existing results on particle filter design have mainly focused on the trade-off between estimation accuracy and computational complexity. It is believed, however, that with the development of hardware technology and improvement of computational power, particle filter will find more extensive applications in various fields.

At last, we point out the following research directions in the area of particle filter which are worthy of further studies:

- **How to incorporate prior knowledge into the design of particle filter:** the efficiency of particle filter is highly dependent on the number of particles employed to represent the posterior PDF that is of interest. Without any prior knowledge, one can only use a large number of particles for an exhaustive exploration of the state space, which will result in an excessive computational burden especially in real-time applications. In the standard SIR algorithm, prior knowledge can be incorporated in either the sampling step or the importance step. In the sampling step, knowledge about the model information or the high likelihood region can be reflected in the construction of proposal distribution. Some schemes for proposal distribution adaptation, such as APF, EPF and UPF, have already been proposed in the existing literatures. This idea can be further extended to tackling other types of prior knowledge such as state constraints and time delay. In the importance step, one can address, say, signal fading or quantization, by evaluating a full likelihood function in which the corresponding occurrence probability has been incorporated.

- **How to deal with model uncertainty and norm-bounded noise:** most particle filtering methods proposed in the literature have relied on perfect knowledge about the model information and noise statistics. This is largely due to the fact that one is unable to simulate a random signal without its statistic information. In practical applications, however, one has to deal with model uncertainty and random noise without accurate statistics. This is especially true for networked systems where the accurate occurrence probability of network-induced phenomena is generally unavailable and only an upper bound is known. In the existing literatures, the CRPF method has been proposed as an attempt to incorporate the prescribed cost function into particle filter design. This approach can be further developed so that the existing results and computational tools in the field of guaranteed cost filtering can be applied in particle filter design.

- **How to achieve further variance reduction:** for particle filtering, variance reduction can be achieved in several ways. First, the linear substructure of the dynamic model should be fully exploited. Linear filtering methods can be combined with particle filter to derive an analytical solution to the estimation of conditionally linear states. This is justified by the Rao-Blackwell theorem which reveals that any redundant random variable present in the estimator will cause extra variance. Second, the resampling step should be performed with more flexibility. On one hand, in view of the extra variance that resampling has inevitably introduced, further studies should be focused on how to circumvent resampling while maintaining an acceptable ESS; On the other hand, novel resampling schemes should be developed to address the trade-off between particle diversity and variance reduction.

- **How to apply particle filtering approach in controller design:** the controller design for nonlinear non-Gaussian systems is a quite challenging problem. Recently, it has been suggested that the controller design for nonlinear non-Gaussian systems should be based on the posterior PDF of system states, i.e., the aim of control is to shape the PDF which is represented by the particles and the corresponding weights. Therefore, it is of interest to investigate the interaction between control input and importance weights, and how the error of particle representation affects the performance of control system.

# References

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

2. Djuric PM, Kotecha JH, Zhang J, Huang Y, Ghirmai T, Bugallo MF, Miguez J (2003) Particle filtering. IEEE Signal Process Mag 20(5):19–38

3. Cappe O, Godsill S, Moulines E (2007) An overview of existing methods and recent advances in sequential Monte Carlo. Proc IEEE 95(5):899–924

4. Gordon NJ, Salmond DJ, Smith AFM (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. Proc Inst Elecr Eng F 140:107–113

5. Metropolis NC, Ulam SM (1949) The Monte Carlo method. J Am Stat Assoc 44(247):335–341

6. Metropolis NC, Rosenbluth AW, Rosenbluth MN, Teller AH (1953) Equations of state calculations by fast computing machines. J Chem Phys 21:1087–1091

7. Hastings WK (1970) Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57:97–109

8. Brooks S (1998) Markov chain Monte Carlo method and its application. J R Stat Soc Ser D (Statist.) 47:69–100

9. Andrieu C, Freitas N, Doucet A, Jordan MI (2003) An introduction to MCMC for machine learning. Mach Learn 50:5–43

10. Smith AF, Roberts GO (1993) Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. J R Stat Soc Ser B Stat Methodol 55:3–23

11. Liu J, Chen R (1998) Sequential Monte Carlo methods for dynamic systems. J Am Stat Assoc 93:1032–1044

12. Crisan D, Doucet A (2002) A survey of convergence results on particle filtering methods for practitioners. IEEE Trans Signal Process 50(3):736–746

13. Chopin N (2004) Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. Ann Stat 32(6):2385–2411

14. Hu X-L, Schon TB, Ljung L (2008) A basic convergence result for particle filtering. IEEE Trans Signal Process 56(4):1337–1348

15. Hu X-L, Schon TB (2011) A general convergence result for particle filtering. ieee trans signal process 59(7):3424–3429

16. Mbalawata IS, Sarkka S (2006) Moment conditions for convergence of particle filters with unbounded importance weights. Signal Process 118:133–138

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

18. Kong A, Liu JS, Wong WH (1994) Sequential imputations and Bayesian missing data problems. J Am Stat Assoc 89:278–288

19. de Freitas JFG, Niranjan M, Gee AH, Doucet A (2000) Sequential Monte Carlo methods to train neural network models. Neural Comput 12(4):955–993

20. Van der Merwe R, De Freitas N, Doucet A, Wan E (2001) The unscented particle filter. Adv Neural Inf Process Syst 13:584–590

21. Pitt MK, Shephard N (1999) Filtering via simulation: auxiliary particle filters. J Am Stat Assoc 94(446):590–591

22. Johansen AM, Doucet A (2008) A note on auxiliary particle filters. Stat Probab Lett 78(12):1498–1504

23. Maskell S, Briers M, Wright R, Horridge P (2005) Tracking using a radar and a problem specific proposal distribution in a particle filter. IEE Proc Radar Sonar Navig 152(5):315–322

24. Havangi R, Taghirad HD, Nekoui MA, Teshnehlab M (2014) A square root unscented fastSLAM with improved proposal distribution and resampling. IEEE Trans Ind Electron 61(5):2334–2345

25. Bi H, Ma J, Wang F (2015) An improved particle filter algorithm based on ensemble Kalman filter and Markov chain Monte Carlo method. IEEE J Sel Top Appl Earth Observ Remote Sens 8(2):447–459

26. Carpenter J, Clifford P, Fearnhead P (1999) Improved particle filter for nonlinear problems. IEE Proc Radar Sonar Navig 146(1):2–7

27. Higuchi T (1997) Monte Carlo filtering using genetic algorithm operators. J Stat Comput Simul 59(1):1–23

28. Kitagawa G (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state-space. J Comput Graph Stat 5(1):1–25

29. Fu X, Jia Y (2010) An improvement on resampling algorithm of particle filters. IEEE Trans Signal Process 58(10):5414–5420

30. Li T, Bolic M, Djuric P (2015) Resampling methods for particle filtering: classification, implementation, and strategies. IEEE Signal Process Mag 32(3):70–86

31. Musso C, Oudjane N, Le Gland F (2001) Improving regularised particle filters. In: Doucet A, de Freitas JFG, Gordon NJ (eds) Sequential Monte Carlo methods in practice. Springer, New York

32. Liu J, Wang W, Ma F (2011) A regularized auxiliary particle filtering approach for system state estimation and battery life prediction. Smart Mater Struct 20(7):075021

33. Doucet A, Gordon NJ, Krishnamurthy V (2001) Particle filters for state estimation of jump Markov linear systems. IEEE Trans Signal Process 49(3):613–624

34. Bunch P, Godsill S (2013) Improved particle approximations to the joint smoothing distribution using Markov chain Monte Carlo. IEEE Trans Signal Process 61(4):956–963

35. Kotecha JH, Djuric P (2003) Gaussian particle filtering. IEEE Trans Signal Process 51(10):2592–2601

36. Petetin Y, Desbouvries F (2013) Optimal SIR algorithm vs. fully adapted auxiliary particle filter: a non asymptotic analysis. Stat Comput 23(6):759–775

37. Casella G, Robert CP (1996) Rao-Blackwellisation of sampling schemes. Biometrika 83(1):81–94

38. Schon T, Gustafsson F, Nordlund P (2005) Marginalized particle filters for mixed linear/nonlinear state-space models. IEEE Trans Signal Process 53(7):2279–2289

39. Karlsson R, Schon T, Gustafsson F (2005) Complexity analysis of the marginalized particle filter. IEEE Trans Signal Process 53(11):4408–4411

40. Smidl V, Hofman R (2011) Marginalized particle filtering framework for tuning of ensemble filters. Mon Weather Rev 139(11):3589–3599

41. Abdallah F, Gning A, Bonnifait P (2008) Box particle filtering for nonlinear state estimation using interval analysis. Automatica 44(3):807–815

42. Gning A, Ristic B, Mihaylova L (2012) Bernouli/box-particle filters for detection and tracking in the presence of triple measurement uncertainty. IEEE Trans Signal Process 60(5):2138–2151

43. Gning A, Ristic B, Mihaylova L, Abdallah F (2013) An introduction to box particle filtering. IEEE Signal Process Mag 30(4):166–171

44. Miguez J, Bugallo MF, Djuric PM (2004) A new class of particle filters for random dynamic systems with unknown statistics. EURASIP J Appl Signal Process 15:2278–2294

45. Djuric PM, Vemula M, Bugallo MF (2008) Target tracking by particle filtering in binary sensor networks. IEEE Trans Signal Process 56(6):2229–2238

46. Yu Y (2013) Combining $H_\infty$ filter and cost-reference particle filter for conditionally linear dynamic systems in unknown non-Gaussiannoises. Signal Process 93:1871–1878

47. Balestrino A, Caiti A, Crisostomi E (2006) Particle filtering within a set-membership approach to state estimation. In: Proceedings of 2006 Mediterranean conference on control and automation, vol 1 and 2, pp 44–49

48. Emre O, Vaclav S, Saikat S, Christian L, Fredrik G (2013) Marginalized adaptive particle filtering for nonlinear models with

unknown time-varying noise parameters. Automatica 49:1566–1575

49. Bolic M, Djuric PM, Hong S (2005) Resampling algorithms and architectures for distributed particle filters. IEEE Trans Signal Process 53(7):2442–2450

50. Chen T, Schon TB, Ohlsson H, Ljung L (2011) Decentralized particle filter with arbitrary state decomposition. IEEE Trans Signal Process 59(2):465–478

51. Ribeiro A, Schizas ID, Roumeliotis S, Giannakis GB (2010) Kalman filtering in wireless sensor networks. IEEE Control Syst Mag 30(2):66–86

52. Mahmoud MS, Khalid HM (2013) Distributed Kalman filtering: a bibliographic review. IET Control Theory Appl 7(4):483–501

53. Dong H, Wang Z, Ding SX, Gao H (2014) A survey on distributed filtering and fault detection for sensor networks. Math Probl Eng 2014:858624-1–858624-7. doi:10.1155/2014/858624

54. Tsai JSH, Lu FC, Provence RS, Shieh LS, Han Z (2009) A new approach for adaptive blind equalization of chaotic communication: the optimal linearization technique. Comput Math Appl 58:1687–1698

55. Kwakernaak (1967) Optimal filtering in linear systems with time delays. IEEE Trans Autom Control AC–12(2):169–173

56. Priemer R, Vacroux AG (1969) Estimation in linear discrete systems with multiple time delays. IEEE Trans Autom Control AC–14:384–387

57. Farooq M, Mahalanabis AK (1971) A note on the maximum likelihood state estimation of linear discrete systems with multiple time delays. IEEE Trans Autom Control 16(1):104–105

58. Pila AW, Shaked U, de Souza CE (1999) $H_\infty$ filtering for continuous-time linear systems with delay. IEEE Trans Autom Control 44(7):1412–1417

59. Fridman E, Shaked U (2001) A new $H_\infty$ filter design for linear time delay systems. IEEE Trans Signal Process 49(11):2839–2843

60. Palhares RM, de Souza CE, Peres PLD (2001) Robust filtering for uncertain discrete-time state-delayed systems. IEEE Trans Signal Process 49(8):1696–1703

61. de Souza CE, Palhares RM, Peres PLD (2001) Robust $H_\infty$ filter design for uncertain linear systems with multiple time-varying state delays. IEEE Trans Signal Process 49(3):569–576

62. Wang Z, Burnham KJ (2001) Robust filtering for a class of stochastic uncertain nonlinear time-delay systems via exponential state estimation. IEEE Trans Signal Process 49(4):794–804

63. Fridman E, Shaked U, Xie L (2003) Robust $H_\infty$ filtering of linear systems with time-varying delay. IEEE Trans Autom Control 48(1):159–165

64. Gao H, Wang C (2003) Delay-dependent robust $H_\infty$ and $L_2 - L_\infty$ filtering for a class of uncertain nonlinear time-delay systems. IEEE Trans Autom Control 48(9):1661–1666

65. Wang Z, Ho DWC (2003) Filtering on nonlinear time-delay stochastic systems. Automatica 39:101–109

66. Shen B, Wang Z, Shu H, Wei G (2009) $H_\infty$ filtering for nonlinear discrete-time stochastic systems with randomly varying sensor delays. Automatica 45:1032–1037

67. He Y, Wang Q, Lin C (2006) An improved $H_\infty$ filter design for systems with time-varying interval delay. IEEE Trans Circ Syst II Express Brief 53(11):1235–1239

68. Zhang X, Han Q (2008) Robust $H_\infty$ filtering for a class of uncertain linear systems with time-varying delay. Automatica 44:157–166

69. Lu X, Zhang H, Wang W, Teo K (2005) Kalman filtering for multiple time-delay systems. Automatica 41:1455–1461

70. Zhang H, Lu X, Cheng D (2006) Optimal estimation for continuous-time systems with delayed measurements. IEEE Trans Autom Control 51(5):823–827

71. Kong S, Saif M, Zhang H (2013) Optimal filtering for Itô-stochastic continuous-time systems with multiple delayed measurements. IEEE Trans Autom Control 58(7):1872–1877

72. Cacace F, Conte F, Germani A (2015) Filtering continuous-time linear systems with time-varying measurement delay. IEEE Trans Autom Control 60(5):1368–1373

73. Schenato L, Sinopoli B, Franceschetti M, Poolla K, Sastry SS (2007) Foundations of control and estimation over lossy networks. Proc IEEE 95(1):163–187

74. Plarre K, Bullo F (2009) On Kalman filtering for detectable systems with intermittent observations. IEEE Trans Autom Control 54(2):386–390

75. Sahebsara M, Chen T, Shah SL (2007) Optimal $H_2$ filtering in networked control systems with multiple packet dropout. IEEE Trans Autom Control 52(8):1508–1513

76. Hu J, Wang Z, Gao H, Stergioulas LK (2012) Extended Kalman filtering with stochastic nonlinearities and multiple missing measurements. Automatica 48:2007–2015

77. Sinopoli B, Schenato L, Franceschetti M, Poolla K, Jordan MI, Sastry SS (2004) Kalman filtering with intermittent observations. IEEE Trans Autom Control 49(9):1453–1464

78. Xie L, Xie L (2009) Stability analysis of networked sampled-data linear systems with Markovian packet losses. IEEE Trans Autom Control 54(6):1368–1374

79. You K, Fu M, Xie L (2011) Mean square stability for Kalman filtering with Markovian packet losses. Automatica 47:2647–2657

80. Censi A (2011) Kalman filtering with intermittent observations: convergence for semi-Markov chains and an intrinsic performance measure. IEEE Trans Autom Control 56(2):376–381

81. Kar S, Sinopoli B, Moura JMF (2012) Kalman filtering with intermittent observations: weak convergence to a stationary distribution. IEEE Trans Autom Control 57(2):405–420

82. Fu M, de Souza CE (2009) State estimation for linear discrete-time systems using quantized measurements. Automatica 45:2937–2945

83. Leong AS, Dey S, Nair GN (2013) Quantized filtering schemes for multi-sensor linear state estimation: stability and performance under high rate quantization. IEEE Trans Signal Process 61(15):3852–3865

84. Li D, Kar S, Alsaadi FE, Dobaie AM, Cui S (2015) Distributed Kalman filtering with quantized sensing state. IEEE Trans Signal Process 63(19):5180–5193

85. Hu L, Wang Z, Liu X (2016) Dynamic state estimation of power systems with quantization effects: a recursive filter approach. IEEE Trans Neural Netw Learn Syst 27(8):1604–1614

86. Dey S, Leong AS, Evans JS (2009) Kalman filtering with faded measurements. Automatica 45:2223–2233

87. Quevedo DE, Ahlen A, Leong AS, Dey S (2012) On Kalman filtering over fading wireless channels with controlled transmission powers. Automatica 48:1306–1316

88. Ding D, Wang Z, Shen B, Dong H (2015) Envelope-constrained $H_\infty$ filtering with fading measurements and randomly occurring nonlinearities: The finite horizon case. Automatica 55:37–45

89. Wang Z, Yang F, Ho DWC, Liu X (2006) Robust $H_\infty$ filtering for stochastic time-delay systems with missing measurements. IEEE Trans Signal Process 54(7):2579–2587

90. Dong H, Wang Z, Gao H (2010) Robust $H_\infty$ filtering for a class of nonlinear networked systems with multiple stochastic communication delays and packet dropouts. IEEE Trans Signal Process 58(4):1957–1966

91. Dong H, Wang Z, Gao H (2013) Distributed $H_\infty$ filtering for a class of Markovian jump nonlinear time-delay systems over lossy sensor networks. IEEE Trans Ind Electron 60(10):4665–4672

92. Dong H, Wang Z, Gao H (2012) Distributed filtering for a class of time-varying systems over sensor networks with quantization

errors and successive packet dropouts. IEEE Trans Signal Process 60(6):3164–3173

93. Zhang S, Wang Z, Ding D, Dong H, Alsaadi F, Hayat T (2016) Nonfragile $H_\infty$ fuzzy filtering with randomly occurring gain variations and channel fadings. IEEE Trans Fuzzy Syst 24(3):505–518

94. Moayedi M, Foo YK, Soh YC (2010) Adaptive Kalman filtering in networked systems with random sensor delays, multiple packet dropouts and missing measurements. IEEE Trans Signal Process 58(3):1577–1588

95. Yang R, Shi P, Liu G (2011) Filtering for discrete-time networked nonlinear systems with mixed random delays and packet dropouts. IEEE Trans Autom Control 56(11):2655–2660

96. Shi P, Luan X, Liu F (2012) $H_\infty$ filtering for discrete-time systems with stochastic incomplete measurement and mixed delays. IEEE Trans Ind Electron 59(6):2732–2739

97. Zhang D, Wang Q, Yu L, Shao Q (2013) $H_\infty$ filtering for networked systems with multiple time-varying transmissions and random packet dropouts. IEEE Trans Ind Inform 9(3):1705–1716

98. Sun S (2013) Optimal linear filters for discrete-time systems with randomly delayed and lost measurements with/without time stamps. IEEE Trans Autom Control 58(6):1551–1556

99. Geng H, Liang Y, Zhang X (2014) Linear-minimum-mean-square-error observer for multi-rate sensor fusion with missing measurements. IET Control Theory Appl 8(14):1375–1383

100. Geng H, Liang Y, Pan Q (2016) The joint optimal filtering and fault detection for multi-rate sensor fusion under unknown inputs. Inf Fusion 29:57–67

101. Geng H, Liang Y, Pan Q (2017) Model-reduced fault detection for multi-rate sensor fusion with unknown inputs. Inf Fusion 33:1–14

102. Borkar V, Varaiya P (1982) Asymptotic agreement in distributed estimation. IEEE Trans Autom Control AC–27(3):650–655

103. Castanon DA, Teneketzis D (1985) Distributed estimation algorithms for nonlinear systems. IEEE Trans Autom Control AC–30(5):418–425

104. Viswanathan R (1993) A note on distributed estimation and sufficiency. IEEE Trans Inf Theory 39(5):1765–1767

105. Saber RO, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. Proc IEEE 95(1):215–233

106. Khan UA, Moura JMF (2008) Distributing the Kalman filter for large-scale systems. IEEE Trans Signal Process 56(10):4919–4935

107. Carli R, Chiuso A, Schenato L, Zampieri S (2008) Distributed Kalman filtering based on consensus strategies. IEEE J Sel Areas Commun 26(4):622–633

108. Matei I, Baras JS (2012) Consensus-based linear distributed filtering. Automatica 48:1776–1782

109. Dimakis AG, Kar S, Moura JMF, Rabbat MG, Scaglione A (2010) Gossip algorithms for distributed signal processing. Proc IEEE 98(11):1847–1864

110. Kar S, Moura JMF (2011) Gossip and distributed Kalman filtering: weak consensus under weak detectability. IEEE Trans Signal Process 59(4):1766–1784

111. Das S, Moura JMF (2015) Distributed Kalman filtering with dynamic observations consensus. IEEE Trans Signal Process 63(17):4458–4473

112. Bertrand A, Moonen M (2011) Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks. IEEE Trans Signal Process 59(5):2320–2330

113. Paul H, Fliege J, Dekorsy A (2013) In-network-processing: distributed consensus-based linear estimation. IEEE Commun Lett 17(1):59–62

114. Cattivelli FS, Sayed AH (2010) Diffusion strategies for distributed Kalman filtering and smoothing. IEEE Trans Autom Control 55(9):2069–2084

115. Tu S, Sayed AH (2012) Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. IEEE Trans Signal Process 60(12):6217–6234

116. Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Trans Pattern Anal Mach Intell 6:721–741

117. Casella G, George EI (1992) Explaining the Gibbs sampler. Am Stat 46(3):167–174

118. Moon T (1996) The expectation-maximization algorithm. IEEE Signal Process Mag 13(6):47–60

119. Logothetics A, Krishnamurthy V (1999) Expectation maximizationalgorithms for MAP estimation of jump Markov linear systems. IEEE Trans Signal Process 47(8):2139–2156

120. Doucet A, Logothetis A, Krishnamurthy V (2000) Stochastic sampling algorithms for state estimation of jump Markov linear systems. IEEE Trans Autom Control 45(2):188–202

121. Mustafa Y, Tolpekin V, Stein A (2012) Application of the expectation maximization algorithm to estimate missing values in Gaussian Bayesian network modeling for forest growth. IEEE Trans Geosci Remote Sens 50(5):1821–1831

122. Housfater A, Zhang X, Zhou Y (2006) Nonlinear fusion of multiple sensors with missing data. Proc ICASSP:961–964

123. Zhang X, Khwaja AS, Luo J, Housfater AS, Anpalagan A (2015) Multiple imputation particle filters: convergence and performance analyses for nonlinear state estimation with missing data. IEEE Trans Signal Process 9(8):1536–1547

124. Orton M, Marrs A (2001) Storage efficient particle filters for the out of sequence measurement problem. In: Proceedings of the IEE colloquium on target tracking: algorithms and applications, Enschede, The Netherlands

125. Orton M, Marrs A (2005) Particle filters for tracking with out-of-sequence measurements. IEEE Trans Aerosp Electron Syst 41(2):693–702

126. Mallick M, Kirubarajan T, Arulampalam S (2002) Out-of-sequence measurement processing for tracking ground target using particle filters. Proc IEEE Aerosp Conf 4:1809–1818

127. Zhang W, Huang X, Wang M (2010) Out-of-sequence measurement algorithm based on Gaussian particle filter. Inf Technol J 9(5):942–948

128. Orguner U, Gustafsson F (2008) Storage efficient particle filters for the out of sequence measurement problem. In: Proceedings of ISIF international conference on information fusion, Cologne, Germany

129. Oreshkin BN, Liu X, Coates MJ (2011) Efficient delay-tolerant particle filtering. IEEE Trans Signal Process 59(7):3369–3381

130. Zhang S, Bar-Shalom Y (2012) Out-of-sequence measurement processing for particle filter: exact Bayesian solution. IEEE Trans Aerosp Electron Syst 48(4):2818–2831

131. Berntorp K, Robertsson A, Arzen K-E (2014) Rao-Blackwellized particle filters with out-of-sequence measurement processing. IEEE Trans Signal Process 62(24):6454–6467

132. Orguner U, Gustafsson F (2011) Target tracking with particle filters under signal propagation delays. IEEE Trans Signal Process 59(6):2485–2495

133. Zhang Y, Huang Y, Li N, Zhao L (2015) Particle filter with one-step randomly delayed measurements and unknown latency probability. Int J Syst Sci 47(1):209–221

134. Huang Y, Zhang Y, Li N, Zhao L (2015) Particle filter for nonlinear systems with multiple step randomly delayed measurements. Electron Lett 51(23):1859–1861

135. Sukhavasi R, Hassibi B (2009) Particle filtering for quantized innovations. In: Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP 2009), pp 2229–2232

136. Sukhavasi R, Hassibi B (2013) The Kalman-like particle filter: optimal estimation with quantized innovations/measurements. IEEE Trans Signal Process 61(1):131–136

137. Orton M, Fitzgerald W (2002) A Bayesian approach to tracking multiple targets using sensor arrays and particle filters. IEEE Trans Signal Process 50(2):216–223

138. Zhai Y, Yeary M, Noyer J-C (April 2006) Target tracking in a sensor network based on particle filtering and power-aware design, IMTC 2006—instrumentation and measurement technology conference. Sorrento, Italy, pp 24–27

139. Zhai Y, Yeary MB, Havlicek JP, Fan G (2008) A new centralized sensor fusion-tracking methodology based on particle filtering for power-aware systems. IEEE Trans Instrum Meas 57(10):2377–2387

140. Tian Q, Pan Y, Yan X, Zheng N, Huan R (2013) Particle state compression scheme for centralized memory-efficient particle filters. IEEE international conference on acoustics, speech, and signal processing (ICASSP), Vancouver, Canada, pp 2577–2581

141. Ozdemir O, Niu R, Varshney PK (2009) Tracking in wireless sensor networks using particle filtering: physical layer considerations. IEEE Trans Signal Process 57(5):1987–1999

142. Masazade E, Niu R, Varshney PK (2012) Dynamic bit allocation for object tracking in wireless sensor networks. IEEE Trans Signal Process 60(10):5048–5063

143. Zheng Y, Niu R, Varshney PK (2014) Sequential Bayesian estimation with censored data for multi-sensor systems. IEEE Trans Signal Process 62(10):2626–2641

144. Hlinka O, Hlawatsch F, Djuric P (2013) Distributed particle filtering in agent networks. IEEE Signal Process Mag 30(1):61–81

145. Sheng X, Hu YH (2005) Distributed particle filters for wireless sensor network target tracking. In: Proceedings of IEEE ICASSP, Philadelphia, PA, pp 845–848

146. Zuo L, Mehrotra K, Varshney PK, Mohan CK (2006) Bandwidth-efficient target tracking in distributed sensor networks using particle filters. In: Proceedings of FUSION, Florence, Italy

147. Cheng Q, Varshney PK (2007) Joint state monitoring and fault detection using distributed particle filtering. In: Proceedings of 41st Asilomar conference on conference on signals, systems, and computers, Pacific Grove, CA, pp 715–719

148. Vemula M, Bugallo MF, Djuric PM (2006) Target tracking in a twotiered hierarchical sensor network. In: Proceedings of IEEE ICASSP, Toulouse, France

149. Ihler AT, Fisher JW III, Willsky AS (2005) Particle filtering under communications constraints. In: Proceedings of IEEE SSP, Bordeaux, France

150. Guo D, Wang X (2004) Dynamic sensor collaboration via sequential Monte Carlo. IEEE J Sel Areas Commun 22:1037–1047

151. Sheng X, Hu YH, Ramanathan P (2005) Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network. In: Proceedings of IPSN, Los Angeles, CA

152. Williams JL, Fisher JW III, Willsky AS (2007) Approximate dynamic programming for communication-constrained sensor network management. IEEE Trans Signal Process 55:4300–4311

153. Bordin CJ Jr, Bruno MGS (2008) Cooperative blind equalization of frequency-selective channels in sensor networks using decentralized particle filtering. In: Proceedings of 42nd Asilomar conference on signals, systems, and computers, Pacific Grove, CA, USA, pp 1198–1201

154. Bordin CJ Jr, Bruno MGS (2009) Nonlinear distributed blind equalization using network particle filtering. 15th IEEE workshop statistics signal process, Cardiff, Wales

155. Dias SS, Bruno MGS (2013) Cooperative target tracking using decentralized particle filtering and RSS sensors. IEEE Trans Signal Process 61(14):3632–3646

156. Hlinka O, Sluciak O, Hlawatsch F, Djuric PM, Rupp M (2012) Likelihood consensus and its application to distributed particle filtering. IEEE Trans Signal Process 60(8):4334–4349

157. Hlinka O, Hlawatsch F, Djuric PM (2014) Consensus-based distributed particle filtering with distributed proposal adaptation. IEEE Trans Signal Process 62(12):3029–3041

158. Farahmand S, Roumeliotis SI, Giannakis GB (2011) Set-membership constrained particle filter: Distributed adaptation for sensor networks. IEEE Trans Signal Process 59(9):4122–4138

159. Gu D (2007) Distributed particle filter for target tracking. In: Proceedings of IEEE ICRA, Rome, Italy

160. Gu D (2008) Distributed EM algorithm for Gaussian mixtures in sensor networks. IEEE Trans Neural Netw 19(7):1154–1166

161. Lee SH, West M (2009) Markov chain distributed particle filters (MCDPF). In: Proceedings of 48th IEEE conference on decision and control, pp 5496–5501

162. Lee SH, West M (2013) Convergence of the Markov chain distributed particle filter (MCDPF). IEEE Trans Signal Process 61(4):801–812

163. Seifzadeh S, Khaleghi B, Karray F (2015) Distributed soft-data-constrained multi-model particle filter. IEEE Trans Cybern 45(3):384–394

164. Mohammadi A, Asif A (2013) Distributed particle filter implementation with intermittent/irregular consensus convergence. IEEE Trans Signal Process 61(10):2572–2587

165. Gu D, Junxi S, Zhen H, Hongzuo L (2008) Consensus based distributed particle filter in sensor networks. In: Proceedings of international conference on information and automation, pp 302–307

166. Mohammadi A, Asif A (2015) Distributed consensus + innovation particle filtering for bearing/range tracking with communication constraints. IEEE Trans Signal Process 63(3):620–635

167. Simonetto A, Keviczky T, Babuska R (2010) Distributed nonlinear estimation for robot localization using weighted consensus. In: Proceedings of IEEE international conference on robotics and automation, pp 3026–3031