Computers, Materials & Continua CMC, vol.61, no.3, pp.1323-1344, 2019

Research on Time Synchronization Method Under Arbitrary Network Delay in Wireless Sensor Networks

Bing Hu[1], Feng Xiang[2], Fan Wu[3], Jian Liu[4], Zhe Sun[1] and Zhixin Sun[1, *]

Abstract: To cope with the arbitrariness of the network delays, a novel method, referred to as the composite particle filter approach based on variational Bayesian (VB-CPF), is proposed herein to estimate the clock skew and clock offset in wireless sensor networks. VB-CPF is an improvement of the Gaussian mixture kalman particle filter (GMKPF) algorithm. In GMKPF, Expectation-Maximization (EM) algorithm needs to determine the number of mixture components in advance, and it is easy to generate overfitting and underfitting. Variational Bayesian EM (VB-EM) algorithm is introduced in this paper to determine the number of mixture components adaptively according to the observations. Moreover, to solve the problem of data packet loss caused by unreliable links, we propose a robust time synchronization (RTS) method in this paper. RTS establishes an autoregressive model for clock skew, and calculates the clock parameters based on the established autoregressive model in case of packet loss. The final simulation results illustrate that VB-CPF yields much more accurate results relative to GMKPF when the network delays are modeled in terms of an asymmetric Gaussian distribution. Moreover, RTS shows good robustness to the continuous and random dropout of time messages.

Keywords: Time synchronization, particle filter, expectation maximization, wireless sensor networks (WSNs).

1 Introduction

Wireless sensor networks (WSNs) consist of many low-cost sensor nodes capable of onboard sensing, computing and communications. WSNs are gaining importance since its wide applications, such as environment monitoring, object tracking and industrial machines controlling, etc. Most of these applications require the time of nodes to be synchronized to each other. Furthermore, some fundamental operations, such as power management, data fusion and transmission scheduling, etc. Require all the nodes running on a common timescale. However, in WSNs, every individual sensor works independently and maintains

[1] Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China.

[2] YuanTong Express Co. LTD, Shanghai, 201705, China.

[3] Department of Computer Science, Tuskegee University, Tuskegee, AL, 36088, USA.

[4] College of Information Engineering, Nanjing University of Finance and Economics, Nanjing, 210023, China.

[*] Corresponding Author: Zhixin Sun. Email: sunzx@njupt.edu.cn.

a local time measured by its own clock. This makes time synchronization between different nodes a critical piece of infrastructure [Wang, Jiang, Zhou et al. (2017); Qiu, Zhang, Qiao et al. (2018); Liu (2018); Chen, Liu and Han (2018)].

Usually, time synchronization between any two nodes is accomplished through the exchange of time messages. Since deterministic and nondeterministic delay exists in the process of message transmission, the time messages may be arbitrarily delayed. At present, the common solution is to build a distribution model for the nondeterministic network delay, such as, Gaussian, Exponent, Gamma and Weber etc. [Wu, Chaudhari and Serpedin (2011); Wang, Jeske and Serpedin (2015); Noh, Chaudhari and Serpedin (2007)]. However, for sensor networks with complex real environment, various reasons will affect the distribution of network delay in varying degrees. It is difficult to find a network delay distribution model that is in line with the actual environment. Although the rationality and applicability of Gaussian network delay distribution model and Exponential network delay distribution model are verified in reference [Etzlinger, Wymeersch and Springer (2013); Abdel-Ghaffar (2002)], the simulation results in Noh et al. [Noh, Chaudhari and Serpedin (2007)] show that the estimation accuracy of clock parameters is very sensitive to the network delay distribution model. Therefore, it is necessary to research the clock parameter estimation method under arbitrary delay [Kim, Lee, Serpedin et al. (2009); Kim, Lee, Serpedin et al. (2011); Guo, Shen, Sun et al. (2015)].

Since Gaussian Mixture Model (GMM) can approximate arbitrary probability density [Anderson and Moore (1979)], Kim et al. [Kim, Lee, Serpedin et al. (2009)] estimated nondeterministic delay distribution using GMM, and proposed two estimation algorithms of clock offset, Gaussian Mixture Kalman Particle Filter (GMKPF), and Iterative Gaussian Mixture Kalman Particle Filter (IGMKPF), respectively. GMKPF combines measurement update steps based on Important Sampling (IS) with Gaussian Sum Filter (GSF) based on Kalman Filter (KF) for time update and proposed distribution generation. Then, Expectation Maximization (EM) algorithm is used to approximate the posterior distribution function of clock parameters by GMM. The introduction of EM algorithm not only alleviates the particle degradation problem caused by the particle filer algorithm, but also avoids the phenomenon that the number of GMM components increases exponentially with the iterations number increases. The simulation results of GMKPF show that when the network delay follows a single non-Gaussian (non-Exponential) distribution or a mixture of arbitrary distributions, it can maintain high synchronization accuracy with fewer message exchanges. However, GMKPF only tracks the clock offset and dose not estimate the clock skew, which will greatly reduce the synchronization period and increase communication overhead. Moreover, when GMKPF uses EM algorithm to approximate posterior distribution function with GMM, it needs to determine the components number of GMM beforehand, which is prone to under-fitting or over-fitting, and the estimation accuracy of parameters depends on the initial values setting. If set improperly, it is likely to converge to the local maximum. On the other hand, the unreliability of sensor network links may lead to the loss of time messages during transmission, while GMKPF does not discuss the algorithm performance in this case.

Therefore, we propose a Composite Particle Filter Approach based on Variational Bayesian (VB-CPF) to realize the joint estimation of clock offset and skew in this paper.

VB-CPF replaces the EM algorithm with Variational Bayesian EM (VB-EM) algorithm, which is used to estimate the GMM parameters of posterior distribution function. VB-EM algorithm is a deterministic approximate reasoning algorithm. It can determine the number of mixture components while determining the estimated values of mixture model parameters. Moreover, in order to solve the problem of time message loss caused by unreliable links, a robust time synchronization method (RTS) is proposed. RTS establishes an autoregressive model for clock skew, and utilizes the estimated clock skew obtained from each iteration of VB-CPF to estimate the autoregressive model parameters by recursive least squares method. If the time message is not received, the node can estimate the current clock parameters through the established clock skew autoregressive model, thus improving the robustness of the time synchronization method.

2 System model

2.1 Discrete clock model

The clock model of sensor node A is shown in Formula (1).

$$c_A(t) = \beta_A(t)t + \theta_A \tag{1}$$

where β_A and θ_A represents the clock rate and initial clock phase, respectively. Since β_A is time-varying, the above model can be expressed as an integral form, as shown in formula (2).

$$c_A(t) = \int_0^t \beta_A(\tau)d\tau + \theta_A \tag{2}$$

Time synchronization between nodes is usually achieved by the exchange of time stamps, which can be regarded as discrete samples of continuous time. Assuming that the sampling period is τ_0, the discrete clock model of node A is

$$\begin{aligned} c_A(n) &= \sum_{k=1}^{n} \beta_A(k)\tau_0 + \theta_A \\ &= n\tau_0 + \underbrace{\sum_{k=1}^{n-1}(\beta_A(k)-1)\tau_0 + \theta_A}_{v_A(n-1)} + (\beta_A(n)-1)\tau_0 \\ &= n\tau_0 + v_A(n-1) + (\beta_A(n)-1)\tau_0 \end{aligned} \tag{3}$$

where $v_A(n)$ denotes the cumulative clock offset, $\beta_A(k)$ denotes the instantaneous clock skew at the k-th sampling. According to the analysis of Luo [Luo (2014)], the time-varying clock skew $\beta_A(n)$ can be described by the Gauss-Markov model, that is

$$\beta_A(n) = \beta_A(n-1) + u_A(n) \tag{4}$$

where $u_A(n) \sim \mathcal{N}(0,\ \sigma_{u_A}^2)$.

On the other hand, according to the definition of cumulative clock offset in formula (3), the recursive form of $v_A(n)$ can be written as

$$v_A(n) = v_A(n-1) + (\beta_A(n) - 1)\tau_0 \tag{5}$$

Substituting formula (4) into formula (5), and obtain

$$v_A(n) = v_A(n-1) + \tau_0 \cdot \beta_A(n-1) + \tau_0 \cdot u_A(n) - \tau_0 \tag{6}$$

Let $\mathbf{x}_A(n) = [\beta_A(n) \;\; v_A(n)]^T$, according to formulas (4) and (6), the clock parameter evolution model of node A can be written in the following matrix form.

$$\mathbf{x}_A(n) = \underbrace{\begin{bmatrix} 1 & 0 \\ \tau_0 & 1 \end{bmatrix}}_{\mathbf{F}} \mathbf{x}_A(n-1) + \underbrace{\begin{bmatrix} u_A(n) \\ \tau_0 u_A(n) \end{bmatrix}}_{\mathbf{w_n}} + \underbrace{\begin{bmatrix} 0 \\ -\tau_0 \end{bmatrix}}_{\mathbf{Y}} \tag{7}$$

2.2 Local timestamp measurement model

In order to establish the clock relationship between two neighbor nodes, a two-way time-stamps exchange mechanism is adopted in this paper. As shown in Fig. 1, in the i^{th} round of message exchange, R sends a synchronization message to A at t_i^1, embedding its clock reading $c_R(t_i^1)$. Upon reception of this message at t_i^2, A records its time $c_A(t_i^2)$, and replies R at t_i^3. The replied message contains both time-stamps $c_A(t_i^2)$ and $c_A(t_i^3)$. Then R records the reception time of A's replay as $c_R(t_i^4)$. Note that $c_R(t_i^1)$ and $c_R(t_i^4)$ are clock readings recorded by R, while $c_A(t_i^2)$ and $c_A(t_i^3)$ are recorded by A. After N rounds of message exchange, the node R obtains a set of time stamps $\{c_R(t_i^1), c_A(t_i^2), c_A(t_i^3), c_R(t_i^4)\}_{i=1}^N$. The above procedure can be modeled as

$$c_A(t_i^2) = \beta_A(c_R(t_i^1) + d + X_i) + \theta_A \tag{8}$$

$$c_A(t_i^3) = \beta_A(c_R(t_i^4) - d - Y_i) + \theta_A \tag{9}$$

where d represents the deterministic delay in message transmission between two nodes, X_i and Y_i represent the nondeterministic delays.

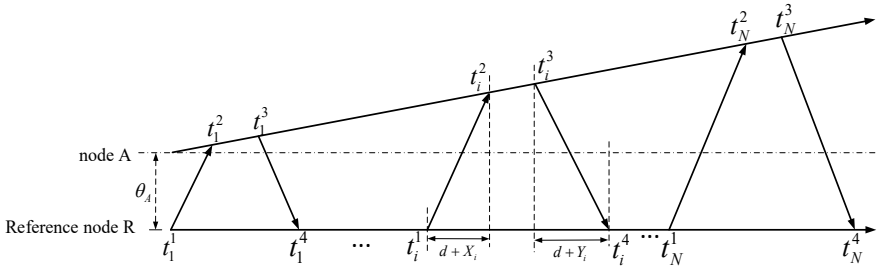

Figure 1: Two-way time-stamps exchange between two nodes R and A

For the time to complete a round of message exchange is very short, we assume that the clock parameters remain unchanged during a round of message exchange. Firstly, the clock model (2) is represented by reference clock and accumulative clock, as follows:

$$\begin{aligned} c_A(t) &= \int_0^t \beta_A(\tau)d\tau + \theta_A \\ &= t + \int_0^t [\beta_A(\tau) - 1]d\tau + \theta_A \\ &= t + v_A(t) \end{aligned} \tag{10}$$

To simplify symbolic representation , let $T_{1,t} \triangleq c_R(t_i^1)$, $T_{2,t} \triangleq c_A(t_i^2)$, $T_{3,t} \triangleq c_A(t_i^3)$, $T_{4,t} \triangleq c_R(t_i^4)$, then formulas (8) and (9) can be simplified as

$$T_{2,t} - v_A(t) = T_{1,t} + d + X_t \tag{11}$$

$$T_{3,t} - v_A(t) = T_{4,t} - d - Y_t \tag{12}$$

Add formula (11) and formula (12), and let $z_A(t) \triangleq T_{2,t} + T_{3,t} - T_{1,t} - T_{4,t}$, $V_t \triangleq X_t - Y_t$, then, a discrete local time measurement model is obtained by sampling, as shown in formula (13).

$$z_A(n) = \mathbf{H}^T \mathbf{x}_A(n) + V_n \tag{13}$$

where $z_A(n)$ is the observed value, $\mathbf{H} = [0\ 2]^T$ is the observation vector, V_n is the observation noise. It is easy to observe that formulas (7) and (13) transform the estimation of clock parameters into Gauss-Markov estimation with unknown states.

3 VB-CPF

3.1 KF based time update and proposed distribution generation

Firstly, it should be noted that all probability density function can be approximated by the GMM shown in formula (14) [Anderson and Moore (1979)].

$$p(x) \approx \sum_{g=1}^{G} \varphi^{(g)} \mathcal{N}(x; \mu^{(g)}, \Sigma^{(g)}) \tag{14}$$

where G is the number of mixture components in GMM, $\varphi^{(g)}$ represents the mixture weights and satisfies $\varphi^{(g)} \geq 0, \sum_{g=1}^{G} \varphi^{(g)} = 1$, $\mathcal{N}(x; \mu^{(g)}, \Sigma^{(g)})$ denotes a normal distribution with $\mu^{(g)}$ mean and variance $\Sigma^{(g)}$.

To simplify symbolic representation， let $\mathbf{x}_A(n) \triangleq \mathbf{x}_n$, $z_A(n) \triangleq z_n$, $u_A(n) \triangleq u_n$. Suppose that at time $n-1$, the posterior probability density function $p(\mathbf{x}_{n-1} \mid z_{1:n-1})$ and the density function of observed noise V_{n-1} are approximated by the following GMMs.

$$p(\mathbf{x}_{n-1} \mid z_{1:n-1}) = \sum_{g=1}^{G} \varphi_{n-1}^{(g)} \mathcal{N}(\mathbf{x}_{n-1}; \mu_{n-1}^{(g)}, \Sigma_{n-1}^{(g)}) \tag{15}$$

$$p(V_{n-1}) = \sum_{j=1}^{J} \gamma_{n-1}^{(j)} \mathcal{N}(V_{n-1}; \mu_{V_{n-1}}^{(j)}, R_{n-1}^{(j)}) \tag{16}$$

It can be known from formula (4), u_n is a Gaussian noise with zero mean and variance σ_u^2, so the state noise in formula (7) obey $\mathbf{w}_n = \begin{bmatrix} u_n \\ \tau_0 u_n \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_n)$, then

$$p(\mathbf{w}_n) = \mathcal{N}(\mathbf{w}_n; \mathbf{0}, \mathbf{Q}_n) \tag{17}$$

At time n, the distribution of the current state can be predicted based on previous observations and state information, that is

$$\begin{aligned} &\hat{p}(\mathbf{x}_n \mid z_{1:n-1}) \\ &= \int p(\mathbf{x}_n \mid \mathbf{x}_{n-1}) p(\mathbf{x}_{n-1} \mid z_{1:n-1}) d\mathbf{x}_{n-1} \\ &= \int p(\mathbf{x}_n \mid \mathbf{x}_{n-1}) \sum_{g=1}^{G} \varphi_{n-1}^{(g)} \mathcal{N}(\mathbf{x}_{n-1}; \mu_{n-1}^{(g)}, \Sigma_{n-1}^{(g)}) d\mathbf{x}_{n-1} \\ &= \int \mathcal{N}(\mathbf{x}_n; \mathbf{F}\mathbf{x}_{n-1} + \mathbf{Y}, \mathbf{Q}_n) \sum_{g=1}^{G} \varphi_{n-1}^{(g)} \mathcal{N}(\mathbf{x}_{n-1}; \mu_{n-1}^{(g)}, \Sigma_{n-1}^{(g)}) d\mathbf{x}_{n-1} \end{aligned} \tag{18}$$

Theorem 1 [Anderson and Moore (1979)] If the state equation of the system is the form of $x_n = f_n(x_{n-1}) + g_n(x_{n-1}) w_{n-1}$, and $p(\mathbf{x}_{n-1} \mid z_{1:n-1})$ can be expressed in form of Gauss sum as shown in formula (15), then the one-step predictive density function is uniformly close to Gauss sum

$$\sum_{g=1}^{G} \tilde{\varphi}_n^{(g)} \mathcal{N}(\mathbf{x}_n; \tilde{\mu}_n^{(g)}, \tilde{\Sigma}_n^{(g)})$$

According to Theorem 1, $\hat{p}(\mathbf{x}_n \mid z_{1:n-1})$ in formula (18) can be written as follows:

$$\hat{p}(\mathbf{x}_n \mid z_{1:n-1}) = \sum_{g=1}^{G} \tilde{\varphi}_n^{(g)} \mathcal{N}(\mathbf{x}_n; \tilde{\mu}_n^{(g)}, \tilde{\Sigma}_n^{(g)}) \tag{19}$$

The mixture model parameters can be obtained from a set of parallel KF.

$$\begin{aligned} \tilde{\mu}_n^{(g)} &= \mathbf{F} \mu_{n-1}^{(g)} \\ \tilde{\Sigma}_n^{(g)} &= \mathbf{F} \Sigma_{n-1}^{(g)} \mathbf{F}^T + \mathbf{Q}_n \\ \tilde{\varphi}_n^{(g)} &= \varphi_{n-1}^{(g)} \end{aligned} \tag{20}$$

Then, according to the latest observation value z_n obtained from the system observation Eq. (13), combined with the density functions $p(\mathbf{x}_{n-1} \mid z_{1:n-1})$ and $p(V_n)$, the measurement update of the posterior density function $\hat{p}(\mathbf{x}_n \mid z_{1:n})$ is completed by KF. The measurement update process of $\hat{p}(\mathbf{x}_n \mid z_{1:n})$ is shown in formula (21).

$$\begin{aligned}
&\hat{p}(\mathbf{x}_n \mid z_{1:n}) \\
&= \frac{p(z_n \mid \mathbf{x}_n)\hat{p}(\mathbf{x}_n \mid z_{1:n-1})}{p(z_n \mid z_{1:n-1})} \\
&= B_n p(z_n \mid \mathbf{x}_n)\hat{p}(\mathbf{x}_n \mid z_{1:n-1}) \\
&= B_n[\sum_{g=1}^{G}\tilde{\varphi}_n^{(g)}\mathcal{N}(\mathbf{x}_n;\tilde{\mu}_n^{(g)},\tilde{\Sigma}_n^{(g)})\sum_{j=1}^{J}\gamma_n^{(j)}\mathcal{N}(z_n;H\mathbf{x}_n+\mu_{V_n}^{(j)},R_n^{(j)})]
\end{aligned} \tag{21}$$

where $B_n = (\int p(z_n \mid \mathbf{x}_n)\hat{p}(\mathbf{x}_n \mid z_{1:n-1})d\mathbf{x}_n)^{-1}$.

Theorem 2 [Anderson and Moore (1979)] If the observation equation of the system is the form of $z_n = h_n(x_n) + V_n$, and $p(\mathbf{x}_n \mid z_{1:n-1})$ can be expressed in form of Gauss sum as shown in formula (19), then the updated posterior density function $\hat{p}(\mathbf{x}_n \mid z_{1:n})$ is uniformly close to Gauss sum $\sum_{g'=1}^{G'}\varphi_n^{(g')}\mathcal{N}(\mathbf{x}_n;\mu_n^{(g')},\Sigma_n^{(g')})$ at $\tilde{\Sigma}_{n-1}^{(g)} \to 0, g = 1,\cdots G$.

So have

$$\hat{p}(\mathbf{x}_n \mid z_{1:n}) = \sum_{g'=1}^{G'}\varphi_n^{(g')}\mathcal{N}(\mathbf{x}_n;\mu_n^{(g')},\Sigma_n^{(g')}) \tag{22}$$

where

$$\begin{aligned}
&G' = GJ \\
&K_n = \tilde{\Sigma}_n^{(g)}H^T(H\tilde{\Sigma}_n^{(g)}H^T + R_n^{(j)})^{-1} \\
&\mu_n^{(g')} = \tilde{\mu}_n^{(g)} + K_n(z_n - H\tilde{\mu}_n^{(g)}) \\
&\Sigma_n^{(g')} = \tilde{\Sigma}_n^{(g)} - K_n H\tilde{\Sigma}_n^{(g)}
\end{aligned} \tag{23}$$

Then update its weight

$$\varphi_n^{(g')} = \frac{\tilde{\varphi}_n^{(g)}\gamma_n^{(j)}\lambda_n^{(j)}}{\sum_{g=1}^{G}\sum_{j=1}^{J}\tilde{\varphi}_n^{(g)}\gamma_n^{(j)}\lambda_n^{(j)}} \tag{24}$$

where $\lambda_n^{(j)} \sim \mathcal{N}(z_n - H\tilde{\mu}_n^{(g)}, H\tilde{\Sigma}_n^{(g)}H^T + R_n^{(j)})$, $g' = g + (j-1)G$. The posterior probability density $\hat{p}(\mathbf{x}_n \mid z_{1:n})$ will be used as the proposed distribution function of the measurement update step, which is based on Importance Sampling (IS).

It is easy to find that the number of mixture components in GMM of posterior probability density $\hat{p}(\mathbf{x}_n \mid z_{1:n})$ increases from G to G', and with the increase of iterations, the number of mixture components increases exponentially, which will greatly increase the computational complexity. So, the number of mixture components must be reduced by adopting corresponding schemes.

3.2 Measurement update based on VB-EM algorithm

IS is a Monte Carlo method, which represents distribution $p(x)$ by empirical approximation based on weighted particle (sample) set, that is, $p(x) \approx \hat{p}(x) = \sum_{l=1}^{N} \zeta^{(l)} \delta(x - \chi^{(l)})$, where $\delta(\bullet)$ is a Dirac delta function and the weighted particle set $\{\zeta^{(l)}, \chi^{(l)}; l = 1, \cdots, M\}$ is obtained from the proposed distribution $q(x)$. The first step in implementing IS is to sample particles from the proposed distribution function $q(\mathbf{x}_n) = \hat{p}(\mathbf{x}_n \mid z_{1:n})$ (formula (22)), and then calculate their corresponding importance weights.

$$\tilde{\zeta}_n^{(l)} = \frac{p(z_n \mid \chi_n^{(l)}) \hat{p}(\chi_n^{(l)} \mid z_{1:n-1})}{\hat{p}(\chi_n^{(l)} \mid z_{1:n})} \tag{25}$$

Normalized importance weights

$$\zeta_n^{(l)} = \frac{\tilde{\zeta}_n^{(l)}}{\sum_{m=1}^{M} \tilde{\zeta}_n^{(m)}} \tag{26}$$

Then the posterior probability density $p(\mathbf{x}_n \mid z_{1:n})$ can be approximated as

$$p(\mathbf{x}_n \mid z_{1:n}) \approx \sum_{l=1}^{M} \zeta_n^{(l)} \delta(\mathbf{x}_n - \chi_n^{(l)}) \tag{27}$$

However, after many iterations, the weights of most particles are negligible, and only a few particles have large weights, thus resulting in the phenomenon of particle weight degradation. Although the introduction of resampling step alleviates the problem of particle degradation to a certain extent [Gordon, Salmond and Smith (1993)], excessive resampling will lead to particle depletion. Therefore, VB-EM algorithm is used to replace the resampling step to avoid particle depletion. Here, the GMM expression for posterior probability density $p(\mathbf{x}_n \mid z_{1:n})$ is

$$p(\mathbf{x}_n \mid z_{1:n}) = \sum_{g=1}^{G} \varphi_n^{(g)} \mathcal{N}(\mathbf{x}_n; \mu_n^{(g)}, \Sigma_n^{(g)}) \tag{28}$$

where G denotes the number of mixture components in GMMs, $\varphi_n^{(g)}$ denotes the weight of each mixture component, $\mathcal{N}(\mathbf{x}_n; \mu_n^{(g)}, \Sigma_n^{(g)})$ is a normal distribution with $\mu_n^{(g)}$ mean and covariance matrix $\Sigma_n^{(g)}$. The Gaussian mixture is specified by the parameter set $\Phi = \{\varphi_n^{(1)}, \cdots, \varphi_n^{(G-1)}, \mu_n^{(1)}, \cdots, \mu_n^{(G)}, \Sigma_n^{(1)}, \cdots, \Sigma_n^{(G)}\}$. In order to facilitate the calculation of VB-EM algorithm, the precision matrix $\Lambda_n^{(g)}$ is used to replace the covariance matrix $\Sigma_n^{(g)}$, in which the precision matrix is the inverse of the covariance matrix.

According to the above description, particle set $\mathcal{X}_n = \{\chi_n^{(1)}, \chi_n^{(2)}, \ldots, \chi_n^{(M)}\}$ are sampled from the proposed distribution. Hidden variables $\mathcal{L}_n = \{\mathcal{L}_n^{(1)}, \mathcal{L}_n^{(2)}, \ldots, \mathcal{L}_n^{(M)}\}$ are introduced, where $\mathcal{L}_n^{(l)}$ indicates which Gaussian distribution is the particle $\chi_n^{(l)}$ coming from and

satisfies $\mathcal{L}_n^{(l)} \in \{0,1\}$, $\sum_l \mathcal{L}_n^{(l)} = 1$. Under given mixture weight $\varphi_n = \{\varphi_n^{(1)}, \varphi_n^{(2)}, \ldots, \varphi_n^{(G)}\}$, the conditional probability density of the hidden variables is [Ishikawa, Takeuchi and Nakano (2010)]

$$p(\mathcal{L}_n \mid \varphi_n) = \prod_{l=1}^{M} \prod_{g=1}^{G} (\varphi_n^{(g)})^{\mathcal{L}_n^{lg}} \tag{29}$$

Under the condition of given hidden variables and model parameters, the conditional probability density of particle set $\mathcal{X}_n$ is [Bishop (2016)]

$$p(\mathcal{X}_n \mid \mathcal{L}_n, \mu_n, \Lambda_n) = \prod_{l=1}^{M} \prod_{g=1}^{G} \mathcal{N}(\mathbf{x}_n; \mu_n^{(g)}, (\Lambda_n^{(g)})^{-1})^{\mathcal{L}_n^{lg}} \tag{30}$$

where $\mu_n = \{\mu_n^{(1)}, \mu_n^{(2)}, \ldots, \mu_n^{(G)}\}$, $\Lambda_n = \{\Lambda_n^{(1)}, \Lambda_n^{(2)}, \ldots, \Lambda_n^{(G)}\}$. $\mathcal{L}_n^{lg}$ indicates which of the G Gaussian distributions the particle $\chi_n^{(l)}$ is from, and $\sum_{g=1}^{G} \mathcal{L}_n^{lg} = 1,\ l = 1, 2, \ldots, M$.

Since the Bayesian estimation is used to solve the GMM parameters, all parameters in the model are considered as random variables. Assume that the prior distribution of the mixture weight φ_n obeys the Dirichlet distribution [Gorur and Rasmussen (2010)].

$$p(\varphi_n) = Dir(\varphi_n \mid \vartheta_0) = C(\vartheta_0) \prod_{g=1}^{G} (\varphi_n^{(g)})^{\vartheta_0 - 1} \tag{31}$$

The prior distribution of mean μ_n and precision matrix Λ_n obeys the joint Gaussian-Wishart distribution [Bishop (2016)].

$$p(\mu_n, \Lambda_n) = p(\mu_n \mid \Lambda_n) p(\Lambda_n) = \prod_{g=1}^{G} \mathcal{N}(\mu_n^{(g)}; m_0, (\ell_0 \Lambda_n^{(g)})^{-1}) \mathcal{W}(\Lambda_n^{(g)}; \Re_0, \wp_0) \tag{32}$$

where ϑ_0, m_0, ℓ_0, $\Re_0$, $\wp_0$ are all parameters to be sought, and subscript 0 indicates the initialization value. The goal of solving the GMM parameters using the Bayesian method is to calculate the posterior distribution $p(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n \mid \mathcal{X}_n)$ based on the a priori information described above, but it is difficult to directly calculate the posterior distribution with high dimension. Therefore, VB-EM introduces approximate distribution $q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)$, and gradually approximates to the real posterior distribution through continuous iterative updating. First, the marginal likelihood function $\ln p(\mathcal{X}_n)$ is analyzed.

$$\begin{aligned}\ln p(\mathcal{X}_n) &= \ln \int\int\int\int p(\mathcal{X}_n, \mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n) d\mathcal{L}_n d\varphi_n d\mu_n d\Lambda_n \\ &= \ln \int\int\int\int q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n) \frac{p(\mathcal{X}_n, \mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)}{q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)} d\mathcal{L}_n d\varphi_n d\mu_n d\Lambda_n \\ &= \int\int\int\int q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n) \ln \frac{p(\mathcal{X}_n, \mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)}{q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)} d\mathcal{L}_n d\varphi_n d\mu_n d\Lambda_n \\ &\quad + [-\int\int\int\int q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n) \ln \frac{p(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n \mid \mathcal{X}_n)}{q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)} d\mathcal{L}_n d\varphi_n d\mu_n d\Lambda_n] \\ &= L(q) + KL(q \,\|\, p) \geq L(q)\end{aligned} \tag{33}$$

where

$$p(\mathcal{X}_n, \mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n) = p(\mathcal{X}_n \mid \mathcal{L}_n, \mu_n, \Lambda_n) p(\mathcal{L}_n \mid \varphi_n) p(\varphi_n) p(\mu_n \mid \Lambda_n) p(\Lambda_n) \tag{34}$$

$$L(q) = \int\int\int\int q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n) \ln \frac{p(\mathcal{X}_n, \mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)}{q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)} d\mathcal{L}_n d\varphi_n d\mu_n d\Lambda_n \tag{35}$$

$$KL(q \,\|\, p) = -\int\int\int\int q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n) \ln \frac{p(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n \mid \mathcal{X}_n)}{q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)} d\mathcal{L}_n d\varphi_n d\mu_n d\Lambda_n \tag{36}$$

$KL(q \,\|\, p)$ is the KL (Kullback-Leibler) divergence of the approximate distribution and the true posterior distribution, and $KL(q \,\|\, p) \geq 0$, so $L(q)$ is a lower bound of $\ln p(\mathcal{X}_n)$, also called the lower bound of variation. When the approximate distribution $q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)$ reaches to the real posterior distribution $p(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n \mid \mathcal{X}_n)$, that is $q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n) = p(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n \mid \mathcal{X}_n)$, then $KL(q \,\|\, p) = 0$. Since $\ln p(\mathcal{X}_n)$ is fixed by the distribution $q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)$, if you want to minimize $KL(q \,\|\, p)$, you only need to maximize $L(q)$.

The distribution $q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n)$ can be decomposed into the following form [Bishop (2016)]

$$q(\mathcal{L}_n, \varphi_n, \mu_n, \Lambda_n) = q(\mathcal{L}_n) q(\varphi_n, \mu_n, \Lambda_n) \tag{37}$$

VB-EM algorithm is a generalized EM algorithm, which is also an iterative algorithm. There are also two iterative steps, namely VB-E step and VB-M step.

VB-E: Bring the decomposition form of formula (37) into formula (35), and obtain

$$L(q) = \int\int\int q(\mathcal{L}_n) q(\varphi_n, \mu_n, \Lambda_n) \ln \frac{p(\mathcal{X}_n, \mathcal{L}_n, \mu_n, \Lambda_n \mid \varphi_n)}{q(\mathcal{L}_n) q(\varphi_n, \mu_n, \Lambda_n)} d\mathcal{L}_n d\varphi_n d\mu_n d\Lambda_n \tag{38}$$

Then find the partial derivative of $L(p)$ in respect to $q(\mathcal{L}_n)$, and let it equal to 0, after calculation

$$\ln q^{t+1}(\mathcal{L}_n) = \sum_{l=1}^{M} \sum_{g=1}^{G} \mathcal{L}_n^{lg} \ln \varphi_n^{lg,t} + const \tag{39}$$

where $const$ is a constant independent of variables,

$$\varphi_n^{lg,t} = \frac{1}{2}\mathrm{E}(\ln|\Lambda_n^{(g),t}|) - \frac{D}{2}\ln 2\pi - \frac{1}{2}\mathrm{E}_{\mu_n^{(g)},\Lambda_n^{(g)}}(\mathcal{X}_n^{(l)} - \mu_n^{(g),t})^T \Lambda_n^{(g),t}(\mathcal{X}_n^{(l)} - \mu_n^{(g),t})$$

$$\mathrm{E}(\ln|\Lambda_n^{(g),t}|) = \sum_{i=1}^{D}\psi(\frac{\wp^{(g),t}+1-i}{2}) + D\ln 2 + \ln|\Re^{(g),t}| \tag{40}$$

$$\mathrm{E}_{\mu_n^{(g)},\Lambda_n^{(g)}}(\mathcal{X}_n^{(l)} - \mu_n^{(g),t})^T \Lambda_n^{(g),t}(\mathcal{X}_n^{(l)} - \mu_n^{(g),t})$$
$$= D\ell^{(g),t} + \wp^{(g),t}(\mathcal{X}_n^{(l)} - m^{(g),t})^T \Re^{(g),t}(\mathcal{X}_n^{(l)} - m^{(g),t})$$

D denotes the dimension of the unknown state, $\psi(\bullet)$ is the digamma function. Take exponent on both sides of formula (39), then normalized

$$q^{t+1}(\mathcal{L}_n) = \prod_{l=1}^{M}\prod_{g=1}^{G}(r_n^{lg,t})^{\mathcal{L}_n^{lg}} \tag{41}$$

where $r_n^{lg,t} = \dfrac{\varphi_n^{lg,t}}{\sum_g \varphi_n^{lg,t}}$. Observing formulas (29) and (41), we find that they have similar structures, the only difference is that the parameters are different. Therefore, if $\varphi_n^{(g)}$ is a prior mixing coefficient, then $r_n^{lg,t}$ is a posterior mixing coefficient. Then define three statistics separately.

$$N_n^{(g),t+1} = \sum_{l=1}^{M}\zeta_n^{(l)} r_n^{lg,t} \tag{42}$$

$$\bar{\mathcal{X}}_n^{(g),t+1} = \frac{1}{N_n^{(g),t+1}}\sum_{l=1}^{M}\zeta_n^{(l)} r_n^{lg,t}\mathcal{X}_n^{(l)} \tag{43}$$

$$S_n^{(g),t+1} = \frac{1}{N_n^{(g),t+1}}\sum_{l=1}^{M}\zeta_n^{(l)} r_n^{lg,t}(\mathcal{X}_n^{(l)} - \bar{\mathcal{X}}_n^{(g),t+1})(\mathcal{X}_n^{(l)} - \bar{\mathcal{X}}_n^{(g),t+1})^T \tag{44}$$

VB-M: Calculate $\partial L(q) \big/ \partial q(\varphi_n, \mu_n, \Lambda_n)$ according to formula (38), and let it equal to 0, after calculation

$$\begin{aligned}\ln q^{t+1}(\varphi_n, \mu_n, \Lambda_n) = {} & \ln p(\varphi_n) + \sum_{g=1}^{G}\ln p(\mu_n^{(g)}, \Lambda_n^{(g)}) + E[\ln p(\mathcal{L}_n \mid \varphi_n)] \\ & + \sum_{g=1}^{G}\sum_{l=1}^{M} E[\mathcal{L}_n^{lg}]\ln \mathcal{N}(\mathbf{x}_n \mid \mu_n^{(g)}, (\Lambda_n^{(g)})^{-1}) + const\end{aligned} \tag{45}$$

where $E[\mathcal{L}_n^{lg}] = r_n^{lg}$. It is easy to find that in formula (45), items related to φ_n and items related to μ_n and Λ_n are independent. First, deal with items related to φ_n.

$$\ln q^{t+1}(\varphi_n) = (\vartheta_0 - 1)\sum_{g=1}^{G}\ln \varphi_n^{(g),t} + \sum_{g=1}^{G}\sum_{l=1}^{M} r_n^{lg,t}\ln \varphi_n^{(g),t} \tag{46}$$

Similarly, take exponent on both sides of formula (46), and obtain

$$q^{t+1}(\varphi_n)= Dir(\varphi_n \mid \vartheta^{t+1}) \tag{47}$$

where $\vartheta^{t+1} = \{\vartheta^{(1),t+1}, \vartheta^{(2),t+1}, \ldots, \vartheta^{(G),t+1}\}$ are posterior parameters, and have

$$\vartheta^{(g),t+1} = \vartheta_0 + N_n^{(g),t+1} \tag{48}$$

For $q^{t+1}(\mu_n, \Lambda_n)$, we know that it obeys the Gaussian-Wishart distribution, that is

$$q^{t+1}(\mu_n, \Lambda_n) = \prod_{g=1}^{G} \mathcal{N}(\mu_n^{(g)}; m^{(g),t+1}, (\ell^{(g),t+1}\Lambda_n^{(g)})^{-1}) W(\Lambda_n^{(g)}; \Re^{(g),t+1}, \wp^{(g),t+1}) \tag{49}$$

where

$$\begin{aligned}
&\ell^{(g),t+1} = \ell_0 + N_n^{(g),t+1} \\
&m^{(g),t+1} = \frac{1}{\ell^{(g),t+1}}(\ell_0 m_0 + N_n^{(g),t+1} \bar{\mathcal{V}}_n^{(g),t+1}) \\
&\wp^{(g),t+1} = \wp_0 + N_n^{(g),t+1} \\
&(\Re^{(g),t+1})^{-1} = (\Re_0)^{-1} + N_n^{(g),t+1} S_n^{(g),t+1} + \frac{\ell_0 N_n^{(g),t+1}}{\ell_0 + N_n^{(g),t+1}} (\bar{\mathcal{V}}_n^{(g),t+1} - m_0)(\bar{\mathcal{V}}_n^{(g),t+1} - m_0)^T
\end{aligned} \tag{50}$$

Therefore, the weight, mean and variances of the g^{th} Gaussian mixture component are $N_n^{(g),t+1}$, $m^{(g),t+1}$ and $E(\Lambda_n^{(g),t+1})$ respectively, where $E(\Lambda_n^{(g),t+1})$ can be obtained from the properties of Wishart distribution.

VB-E and VB-M are executed alternately and iteratively. As the number of iterations increases, $L(q)$ gradually increases. The iteration is performed until $|L^{t+1}(q) - L^t(q)| < \varepsilon$, where ε is the predetermined error limit.

Finally, the conditional mean $\hat{\mathbf{x}}_n = E[\mathbf{x}_n \mid z_{1:n}]$ and the corresponding error covariance $\hat{\Sigma}_n = E[(\mathbf{x}_n - \hat{\mathbf{x}}_n)(\mathbf{x}_n - \hat{\mathbf{x}}_n)^T]$ can be obtained in two ways. The first one is obtained directly from the weighted sum of the particle set of formula (27) before the VB-EM algorithm is executed.

$$\begin{aligned}
&\hat{\mathbf{x}}_n = \sum_{l=1}^{M} \zeta_n^{(l)} \chi_n^{(l)} \\
&\hat{\Sigma}_n = \sum_{l=1}^{M} \zeta_n^{(l)} (\chi_n^{(l)} - \hat{\mathbf{x}}_n)(\chi_n^{(l)} - \hat{\mathbf{x}}_n)^T
\end{aligned} \tag{51}$$

The second is calculated according to the formula (28) by using the weighted particle set fitted by GMM.

$$\hat{\mathbf{x}}_n = \sum_{g=1}^{G} \varphi_n^{(g)} \mu_n^{(g)}$$

$$\hat{\Sigma}_n = \sum_{g=1}^{G} \varphi_n^{(g)} [\Sigma_n^{(g)} + (\mu_n^{(g)} - \hat{\mathbf{x}}_n)(\mu_n^{(g)} - \hat{\mathbf{x}}_n)^T] \tag{52}$$

The estimation performance of formula (51) is better than that of formula (52). However, because of $M \gg G$, the first kind of computation complexity is higher than the second one. The pseudo code of VB-CPF is as follows.

Composite Particle Filter Approach based on Variational Bayesian (VB-CPF)

Initialization

1) The probability density function expression of the initial state $\mathbf{x}_0$ is

$$p(\mathbf{x}_0) = \sum_{g=1}^{G} \varphi_0^{(g)} \mathcal{N}(\mathbf{x}_0; \mu_0^{(g)}, \Sigma_0^{(g)})$$

2) Assume that at $n-1$ $(n>1)$, the forms of the posterior density function, process noise and observed noise density function are as follows

- posterior density function

$$p(\mathbf{x}_{n-1} \mid z_{1:n-1}) = \sum_{g=1}^{G} \varphi_{n-1}^{(g)} \mathcal{N}(\mathbf{x}_{n-1}; \mu_{n-1}^{(g)}, \Sigma_{n-1}^{(g)})$$

- process noise density function

$$p(\mathbf{w}_n) = \mathcal{N}(\mathbf{w}_n; \mathbf{0}, \mathbf{Q}_n)$$

- observed noise density function

$$p(V_n) = \sum_{j=1}^{J} \gamma_n^{(j)} \mathcal{N}(V_n; \mu_{V_n}^{(j)}, R_n^{(j)})$$

Prediction process

1) time update

A set of parallel KF is used to calculate the one-step predictive density function $\hat{p}(\mathbf{x}_n \mid z_{1:n-1})$, and the update formula of the parameter is shown in (20).

2) measurement update (the generation of proposed distribution)

A set of parallel KF is used to calculate the posterior density function $\hat{p}(\mathbf{x}_n \mid z_{1:n})$, and the update formulas of the parameter are shown in (23) and (24).

Measurement update process

1) Sampling M particles $\{\chi_n^{(l)}; l = 1, \cdots, M\}$ from the proposed distribution function $q(\mathbf{x}_n) = \hat{p}(\mathbf{x}_n \mid z_{1:n})$

2) Initialization, assign initial values to ϑ_0, m_0, ℓ_0, $\Re_0$, $\wp_0$ and error limits ε

3) Calculate $r_n^{lg,t}$ according to formula (40), then calculate the formulas (42)-(44)

4) Update the parameters in formula (50) with the calculation result in step 3), and then calculate $L^{t+1}(q)$

5) If $|L^{t+1}(q)-L^t(q)|\geq\varepsilon$, go back to step 3). Otherwise, stop iterating and get the optimal parameter estimation $\vartheta^{(g),t+1},\ell^{(g),t+1},m^{(g),t+1},\wp^{(g),t+1},\Re^{(g),t+1}$, where $g=1,2,\ldots,G$

State estimation

Estimate system state and variance using formulas (51) and (52).

4 Robust time synchronization method (RTS)

In practical sensor networks, the links between nodes are unreliable and susceptible to external interference, so the packets containing timestamps may be lost or collided during transmission. In order to solve the problem of data packet loss, the traditional wireless network can solve it by simply retransmitting. However, data packet retransmitting is infeasible for WSNs, because the energy cost of data retransmitting is too large, and the uncertainty delay in the process of data packet retransmitting will also affect the accuracy of time synchronization. So how to ensure the estimation accuracy of clock parameters in the case of data packet loss is the problem to be solved in this section.

According to Section 2.1, $\beta_A(n)$ varies with time in the actual environment, and for each sampling, it is not completely independent [Kim (2014); Kim, Ma and Hamilton (2012)]. Moreover, due to the insufficient energy of nodes and the change of temperature, it may change greatly. In this section, the time-varying clock skew is modeled as an Auto-Regressive (AR) process [Tibshirani (2011)], and then the parameters of AR model are estimated by recursive least square method based on the estimated clock skew obtained by VB-CPF. When encountering data packet loss, the clock parameters of the next time can be estimated according to the AR model, which ensures the robustness of VB-CPF method in unreliable link environment. The AR model of clock skew is as follows.

$$\beta_A(n)=\sum_{l=1}^{P}\pi_l\beta_A(n-l)+\eta(n) \tag{53}$$

where P is the order of AR model, $\pi_l(l=1,2,...,P)$ is AR coefficient, $\eta(n)$ is Gauss noise with zero mean and variance σ_η^2. This section chooses $P=4$, i.e., $\beta_A(n)=\sum_{l=1}^{4}\pi_l\beta_A(n-l)+\eta(n)$, because the fourth-order autoregressive model can simulate the time-varying characteristics of clock skew very well, and its computational complexity is not very high [Kim (2014)]. Next, the process of solving AR coefficients is described.

Assuming that the first five transmitted time messages are received correctly, according to the above estimation method, we can obtain five estimation values of clock skew, i.e., $(\beta_1,\beta_2,\beta_3,\beta_4,\beta_5)$, where $\beta_n=\mathbf{x}_n[1]$, $\mathbf{x}_n[1]$ represents the first element of column vector

$\mathbf{x}_n$. Then

$$\beta_5 = \pi_1\beta_4 + \pi_2\beta_3 + \pi_3\beta_2 + \pi_4\beta_1 + \eta(5) = \underbrace{\left[\beta_4, \beta_3, \beta_2, \beta_1\right]}_{\mathbf{F}_1} \underbrace{\begin{bmatrix}\pi_1\\ \pi_2\\ \pi_3\\ \pi_4\end{bmatrix}}_{\pi_1} + \eta(5) \tag{54}$$

According to the recursive least squares estimation method, the mean square error matrix of π_1 is calculated first.

$$\mathbf{M}_1 = \frac{1}{\sigma_\eta^2}(\mathbf{F}_1^{\mathrm{T}}\mathbf{F}_1)^{-1} \tag{55}$$

Then we can obtain the estimated value of π_1

$$\hat{\pi}_1 = \beta_5\sigma_\eta^2\mathbf{M}_1\mathbf{F}_1^{\mathrm{T}} \tag{56}$$

When β_6 is obtained, the AR coefficients are updated in the following recursive form according to the $\mathbf{M}_1$ and $\hat{\pi}_1$ calculated for the first time.

Step 1: calculate the mean square error matrix of π_2, $\mathbf{M}_2 = (\mathbf{M}_1^{-1} + \sigma_\eta^2\mathbf{F}_2^{\mathrm{T}}\mathbf{F}_2)^{-1}$.

Step 2: update AR coefficient, $\hat{\pi}_2 = \hat{\pi}_1 + \mathbf{K}_2(\beta_6 - \mathbf{F}_2\hat{\pi}_1)$, where $\mathbf{K}_2 = \sigma_\eta^2\mathbf{M}_2\mathbf{F}_2^{\mathrm{T}}$.

Each time a new clock skew estimated value is obtained, AR coefficients are updated according to the above process. With the number of updates increase, AR model fits the estimated value of clock skew better. When the time message is lost or destroyed during transmission, the node can estimate the current clock skew according to the AR model. The overall method block diagram of RTS is shown in Fig. 2.

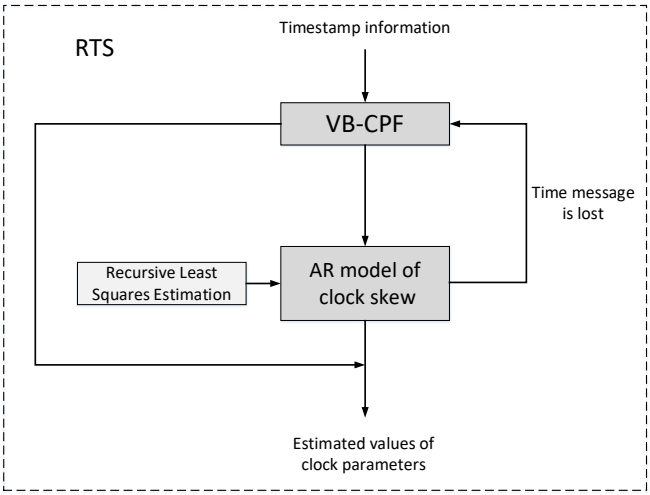

Figure 2: The overall method block diagram of RTS

5 Simulation experiment

To validate the performances of the proposed algorithm, simulation results are presented and compared to GMKPF. Since the VB-CPF method proposed in this paper is an improvement to GMKPF, in order to simplify the description of the simulation results, this section only evaluates the performance of the VB-CPF method under the asymmetric Gaussian delay model, in which the variance of the uplink nondeterministic Gaussian delay is $\sigma_1 = 0.5$, and the downlink is $\sigma_2 = 1$. And the other parameters used in simulations are as follows. Initial clock offsets, clock offsets and deterministic delays are uniformly drawn from ranges $[-5\tau_0 \ 5\tau_0]$, $[0.9 \ 1.1]$ and $[0.01\tau_0 \ 0.02\tau_0]$, respectively, where $\tau_0 = 0.1s$. The variance of u_A in formula (4) is $\sigma_{u_A}^2 = 10^{-4}$. The number of the mixture components in GMM is 3. The performance of the algorithm is evaluated using the mean square error of the clock parameter estimated value.

$$\mathrm{MSE}(\mathbf{x}_n) = \frac{1}{S}\sum_{s=1}^{S}(\hat{\mathbf{x}}_{n,s} - \mathbf{x}_{n,s})^2 \tag{57}$$

where S is the number of times the simulation experiment is executed, let $S = 100$. The smaller the value of MSE, the higher the estimation accuracy of the clock parameters.

Firstly, the effects of time message exchange times N (number of observations) and particle number M on the performance of VB-CPF are analyzed. As shown in Fig. 3 and Fig. 4, as the number of time message exchanges increases, the MSE of the clock parameter estimate gradually decreases, and when $N > 20$, the magnitude of the MSE decrease become smaller. It can also be seen that when the number of particles is large ($M \geq 400$), the value of MSE is smaller. Obviously, through 5 times message exchanges, the estimated value of clock parameters obtained by sampling 500 particles per time is less than that obtained by 25 times message exchanges and sampling 100 particles per time. Therefore, the number of message exchanges can be reduced by increasing the number of particles sampled, thus reducing the communication overhead.

As shown in Fig. 5 and Fig. 6, when the number of time message exchanges is fixed, the MSE of the estimated clock parameters decreases with the increase of the number of particles. When the number of particles is greater than 600, the MSE of the cumulative clock offset estimates decreases slightly. When the number of particles is greater than 500, the MSE of the clock skew estimates tends to be stable. Therefore, considering the synchronization accuracy and computational complexity, in the following comparative analysis of the performance of VB-CPF and GMKPF, we select the number of particles $M = 500$.

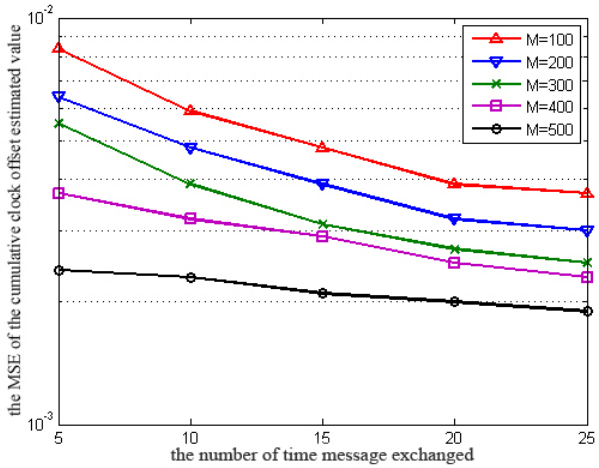

Figure 3: The relationship the MSE of the cumulative clock offset estimated value and the number of time message exchanged

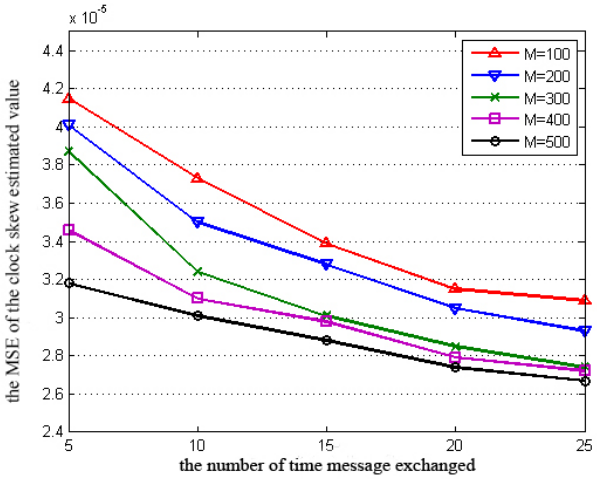

Figure 4: The relationship the MSE of the clock skew estimated value and the number of time message exchanged

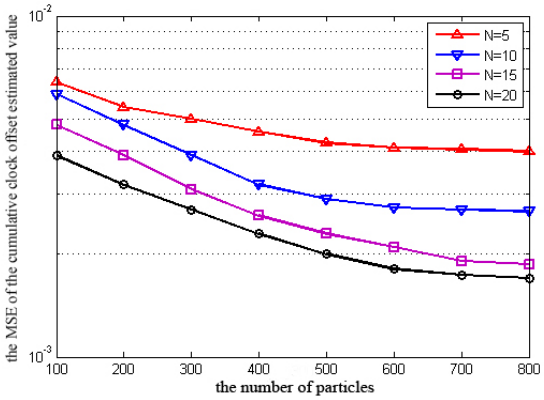

Figure 5: The relationship the MSE of the cumulative clock offset estimated value and the number of particles

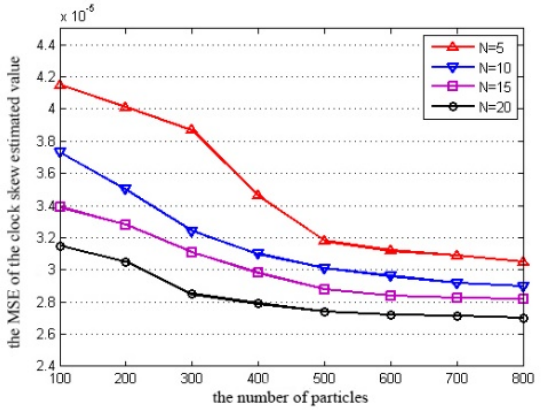

Figure 6: The relationship the MSE of the clock skew estimated value and the number of particles

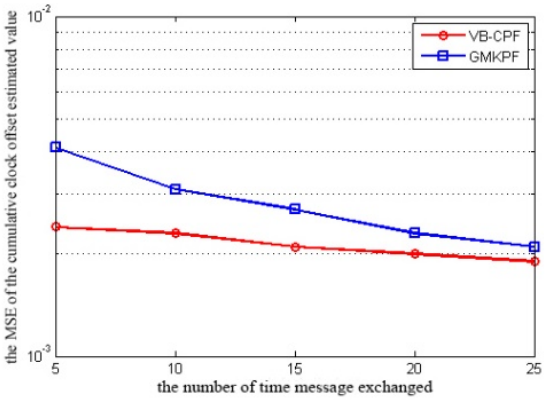

Figure 7: The MSE of the cumulative clock offset estimated value

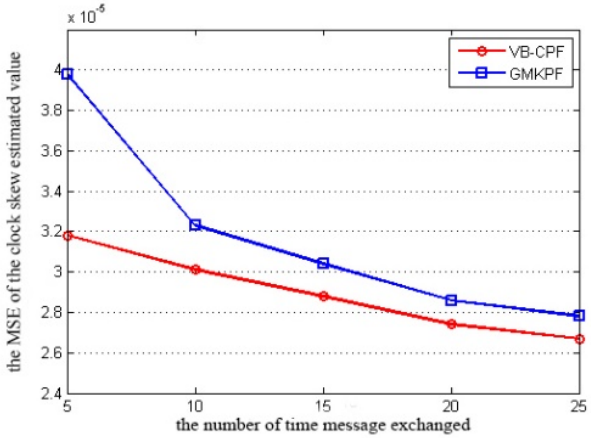

Figure 8: The MSE of the clock skew estimated value

The time synchronization method proposed in Serpedin et al. [Serpedin, Kim and Lee (2009); Kim, Lee, Serpedin et al. (2011)] uses GMKPF algorithm to estimate the clock offset, while the simulation results shown in Fig. 7 and Fig. 8 are the joint estimated value of cumulative clock offset and clock skew, which is obtained using the GMKPF algorithm. As can be seen from the figure, the MSE of the clock parameters estimated value decreases gradually with the increase of the number of time message exchanges. Comparing the two hybrid filtering methods: VB-CPF and GMKPF, the performance of VB-CPF is better than that of GMKPF. This is because VB-CPF uses VB-EM algorithm for Gaussian mixture fitting of sampled particles. VB-EM algorithm can adaptively determine the number of mixture components, avoiding the problem of over-fitting and under-fitting caused by the EM algorithm to determine the number of mixture components in advance.

Next, we evaluate the performance of VB-CPF and RTS in the case of time message lost. The simulation experiments consider two types of time message dropout: (1) continuous dropout, (2) random dropout. Fig. 9 shows the performance comparison of VB-CPF and RTS when time messages (observations) are lost five times in a row. Obviously, the performance of RTS is only 10^{-6} difference from the estimation result without message dropout, which is better than that of VB-CPF. This is because, in each time of message loss, RTS estimates the cumulative clock offset and clock skew based on the pre-established clock skew model, while VB-CPF cannot update the estimated clock parameters. However, as the number of message loss increases, the MSE of the clock parameter estimate by RTS gradually increases. This is because the autoregressive model established for the clock skew is a statistical prediction model that predicts future values based on past estimated clock skew values. Therefore, when the time message is lost for a long time, the performance of the RTS will not be guaranteed.

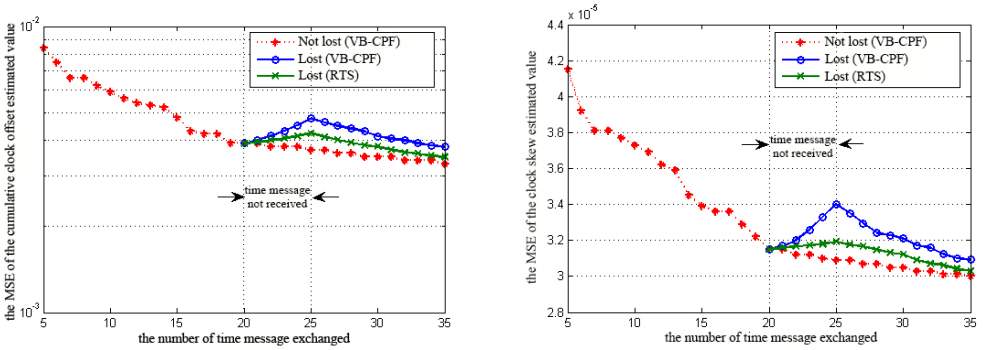

Figure 9: When time messages are lost continuously, the MSE of (a) the cumulative clock offset estimated value and (b) the clock skew estimated value

When time messages are lost randomly, Fig. 10 shows the performance comparison of VB-CPF and RTS. When encountering message loss, the estimation performance of cumulative clock offset that used VB-CPF decrease 5.26×10^{-4} on average when compared with the case of no messages loss, while the estimation performance of RTS decrease 2.54×10^{-4} on average. Similarly, the estimation performance of clock skew that used VB-CPF decrease 1.62×10^{-6} on average when compared with the case of no messages loss, while the estimation performance of RTS only decrease 6.39×10^{-7} on average. Obviously, the MSE of the clock parameter estimation obtained by RTS is smaller than that of VB-CPF, so the robustness of RTS in dealing with time message loss is better than that of VB-CPF. According to the analysis of Fig. 9 and Fig. 10, RTS can show better performance regardless of whether the time messages are continuous dropout or random dropout. This shows that RTS can well solve the problem of synchronization accuracy decline caused by the time message dropout.

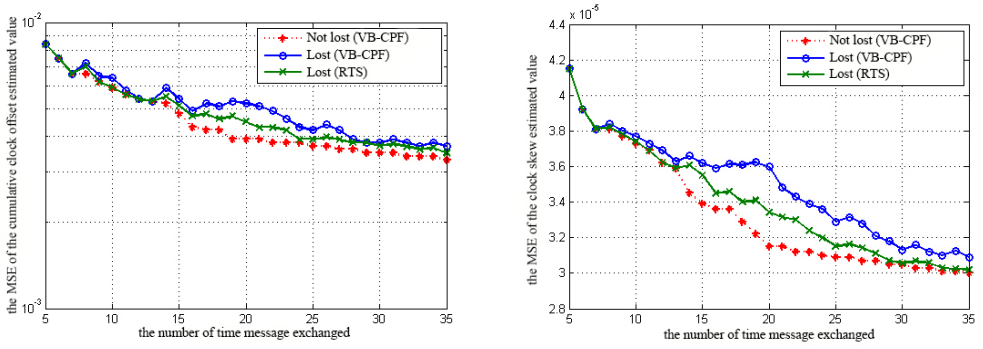

Figure 10: When time messages are lost randomly, the MSE of (a) the cumulative clock offset estimated value and (b) the clock skew estimated value

6 Conclusions

A composite particle filter approach based on variational Bayesian (VB-CPF) is proposed in this paper. VB-CPF replaces the EM algorithm used by GMKPF in estimating the parameters of Gaussian mixture model of the posterior distribution function of clock parameters with VB-EM algorithm, which makes it possible to determine the number of mixture components adaptively, thereby improving the estimation accuracy of clock parameters. At the same time, in order to solve the problem of time message dropout caused by unreliable link, a robust time synchronization method (RTS) is designed. RTS improves the robustness of VB-CPF in unstable network environment by establishing an autoregressive model for clock skew. The simulation results show that the estimation accuracy of clock parameters obtained by VB-CPF is better than that obtained by GMKPF, and RTS is robust to the continuous and random dropout of time messages.

Acknowledgement: This work was supported by the National Natural Science Foundation of China (No. 61672299), the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province of China (No. 18KJB520035), the Youth Foundation of Nanjing University of Finance and Economics (No. L-JXL18002), the Youth Foundation of Nanjing University of Posts and Telecommunications (No. NY218142) and the Natural Science Foundation of Jiangsu Province (No. BK20160913). The authors would like to thank anonymous reviewers for their valuable suggestions.

References

Abdel-Ghaffar, H. S. (2002): Analysis of synchronization algorithms with time-out control over networks with exponentially symmetric delays. *IEEE Transactions on Communications*, vol. 50, no. 10, pp. 1652-1661.

Anderson, B. D. O.; Moore, J. B. (1979): *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall.

Bishop, C. M. (2016): *Pattern Recognition and Machine Learning.* Springer.

Chen, H. P.; Liu, K. X.; Han, Y. (2018): A novel time-aware frame adjustment strategy for RFID anti-collision. *Computers, Materials & Continua*, vol. 57, no. 2, pp. 195-204.

Etzlinger, B.; Wymeersch, H.; Springer, A. (2013): Cooperative synchronization in wireless networks. *IEEE Transactions on Signal Processing*, vol. 62, no. 11, pp. 2837-2849.

Gordon, N. J.; Salmond, D. J.; Smith, A. F. M. (1993): Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings-F Radar and Signal Processing*, vol. 140, no. 2, pp. 107-113.

Gorur, D.; Rasmussen, C. E. (2010): Dirichlet process gaussian mixture models: choice of the base distribution. *Journal of Computer Science and Technology*, vol. 25, no. 4, pp. 653-664.

Guo, C. S.; Shen, J.; Sun, Y.; Ying, N. (2015): RB particle filter time synchronization algorithm based on the DPM model. *Sensors*, vol. 15, no. 9, pp. 22249-22265.

Ishikawa, Y.; Takeuchi, I.; Nakano, R. (2010): Multi-directional search from the primitive initial point for Gaussian mixture estimation using variational Bayes method.

Neural Networks, vol. 23, no. 3, pp. 356-364.

Kim, H. Y. (2014): *Modeling and Tracking Time-Varying Clock Drifts in Wireless Networks (Ph.D. Thesis).* Georgia Institute of Technology.

Kim, H.; Ma, X. L.; Hamilton, B. R. (2012): Tracking low-precision clocks with time-varying drifts using kalman filtering. *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 257-270.

Kim, J. S.; Lee, J.; Serpedin, E.; Qaraqe, K. (2009): A robust estimation scheme for clock phase offsets in wireless sensor networks in the presence of non-Gaussian random delays. *Signal Processing*, vol. 89, no. 6, pp.1155-1161.

Kim, J. S.; Lee, J.; Serpedin, E.; Qaraqe, K. (2011): Robust clock synchronization in wireless sensor networks through noise density estimation. *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3035-3047.

Liu, X. D.; Liu, Q. (2018): A dual-spline approach to load error repair in a HEMS sensor network. *Computers, Materials & Continua*, vol. 57, no. 2, pp. 179-194.

Luo, B. (2014): *Distributed Clock Synchronization for Wireless Sensor Networks (Ph.D. Thesis).* HongKong University.

Noh, K. L.; Chaudhari, Q. M.; Serpedin, E.; Suter, B. W. (2007): Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks. *IEEE Transactions on Communications,* vol. 55, no. 4, pp.766-777.

Qiu, T.; Zhang, Y. S.; Qiao, D. J.; Zhang, X. Y.; Wymore, M. L. et al. (2018): A robust time synchronization scheme for industrial internet of things. *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3570-3580.

Tibshirani, R. (2011): Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society B-Statistical Methodology*, vol. 73, no. 3, pp. 273-282.

Wang, W. D.; Jiang, S. F.; Zhou, H. F.; Yang, M.; Ni, Y. Q. et al. (2018): Time synchronization for acceleration measurement data of Jiangyin bridge subjected to a ship collision. *Structural Control & Health Monitoring*, vol. 25, no. 1, e2039.

Wang, X.; Jeske, D.; Serpedin, E. (2015): An overview of a class of clock synchronization algorithms for wireless sensor networks: a statistical signal processing perspective. *Algorithms*, vol. 8, no. 3, pp. 590-620.

Wu, Y. C.; Chaudhari, Q.; Serpedin, E. (2011): Clock synchronization of wireless sensor networks. *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124-138.