Uncertainty-Constrained Belief Propagation for Cooperative Target Tracking

Cheng Xu, *Member, IEEE*, Yuchen Shi, Jiawang Wan, and Shihong Duan, *Member, IEEE*

***Abstract*—Cooperative localization is essential for many Internet of Things (IoT)-related applications in harsh environments. Generally, the inertial navigation system is self-contained and adopted as the basis of a cooperative tracking system, but it still faces the problem of accumulated errors and cannot provide long-term, high-precision positioning. The particle filter (PF) is widely used to fuse multiple information to inhibit accumulative errors. However, particle degradation and impoverishment remain unsolved. This article proposed an IMU/time-of-arrival (TOA) fusion-based tracking method, namely, uncertainty-constrained belief propagation (UCBP). We address particle degradation and impoverishment by introducing uncertainty-constrained optimization into belief propagation (BP). An uncertainty-constrained resampling (UCR) method is applied to quantify the uncertainty in cooperative systems. Hierarchical resampling is realized to solve the particle impoverishment issue. Meanwhile, particle degradation is resolved through constrained resampling while ensuring the diversity of particles. Furthermore, we illustrated the factor graph (FG) structure of UCBP to mitigate the accumulation of errors through message fusion over the graph. Compared with the state-of-the-art methods, our proposed UCBP algorithm has better precision and robustness without introducing much time overhead.**

***Index Terms*—Accumulative error, belief propagation (BP), cooperative localization, factor graph (FG), multisource fusion, uncertainty constraint.**

I. Introduction

In last decades, cooperative target tracking has been drawing the attention of various location-based applications, such as autopilot [1], [2], emergency search and rescue [3], [4], and pedestrian navigation [5]–[7]. In cooperative scenes, mobile nodes should infer their location based on intersensor measurements and wireless information exchanged with others [8]. Specifically, the localization process consists of two stages [9]. The first phase is the *measurement*, during which the mobile node measures internal state information (e.g., using an inertial measurement unit) and estimates its position based on direct communication with neighboring nodes or anchors. The second phase is the *update*, in which the mobile node infers its state based on internal measurements, estimated location information, and state information from neighboring nodes.

The *measurement* phase is affected by the uncertainty due to multiple sources, such as noise, multipath, blockages, interference, clock drifts, etc [10], [11]. The underlying transmission technology is critical in how these sources affect the measurements. The global navigation satellite system (GNSS) can provide essential support for most applications. But in harsh environments, such as building ruins, urban canyons, and underground seabeds, the probability of signal degradation and blockage is high, which limits its practical usage [12]. Ultrawide bandwidth (UWB) [13] is a promising alternative for localization in communication-constrained environments. Time-of-arrival (TOA)-based UWB does not require unique timestamps or specific devices to achieve accurate positioning in some conditions [14]. However, in harsh environments, TOA-based techniques are often affected by nonline-of-sight (NLOS) [15], resulting in decreased accuracy. IMU-based inertial navigation can realize target tracking in unknown areas without deploying additional base stations. However, its positioning errors will accumulate over time due to the inherent characteristics of dead reckoning. The particle filter (PF) can restrain the accumulative errors, but it faces the problem of particle degradation and impoverishment in the resampling process [14]. In short, TOA and IMU-based methods have significant limitations during the measurement process. The development of multisource cooperative tracking [17]–[19] provides a new direction for solving these issues in require for high precision and efficiency. IMU/TOA fusion [14] is more reliable, but challenges remain in how to quantify the uncertainty during the measurement phase to achieve higher precision target tracking.

For the *update* phase, the factor graph (FG) [22] is a widely used probability graph in Bayesian inference. It provides an abstract graphical representation scheme for multisource information fusion in cooperative tracking problems. Typical use cases include pedestrian collaborative navigation [23] and cooperative aircraft localization [24]. The target node can then cooperatively infer its location by message passing. When the decomposed FG matches the network topology, efficient message passing for distributed inference can be used, such as belief propagation (BP) and mean-field methods [25]. The

Manuscript received 10 July 2021; revised 14 December 2021, 15 February 2022, and 17 March 2022; accepted 3 April 2022. Date of publication 8 April 2022; date of current version 23 September 2022. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62101029; in part by the China National Postdoctoral Program for Innovative Talents under Grant BX20190033; in part by the Postdoctoral Research Foundation of Shunde Graduate School of University of Science and Technology Beijing under Grant 2020BH001; and in part by the Fundamental Research Funds for the Central Universities under Grant 06500127. *(Cheng Xu and Yuchen Shi are co-first authors.) (Corresponding author: Cheng Xu.)*

The authors are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, and also with the Shunde Graduate School, University of Science and Technology Beijing, Foshan 528223, China (e-mail: xucheng@ustb.edu.cn; shiyuchen@xs.ustb.edu.cn; wjiawang@xs.ustb.edu.cn; duansh@ustb.edu.cn).

Digital Object Identifier 10.1109/JIOT.2022.3165818

multisource cooperative tracking based on IMU/TOA fusion can also be viewed as an instance of the BP to realize the information fusion in the FG [23]. However, making full use of the dynamic constraints in collaborative systems is still an open problem in studying BP methods.

Considering the mentioned issues in the *measurement* and *update* phases, this article proposes an IMU/TOA fusion-based tracking method using uncertainty-constrained BP (UCBP). An uncertainty-constrained resampling (UCR) method is proposed to quantify the uncertainty in cooperative systems. UCR establishes error ellipses with different confidence probabilities and realizes hierarchical resampling based on the internal information to solve the particle impoverishment problem. Based on UCR, we extend the formal nonparametric BP (NBP) [26] using IMU/TOA fusion. Finally, the FG of UCBP for cooperative target tracking is illustrated in detail. UCBP can be expressed as probability information spreading on the FG and then reused the information to improve the overall efficiency [28], [29]. In summary, the main contributions of this article are twofold as follows.

1) We proposed the UCBP algorithm by introducing uncertainty-constrained optimization into BP. A UCR method is applied to quantify the uncertainty in cooperative systems. Hierarchical resampling is realized to solve the particle impoverishment issue. Meanwhile, particle degradation is resolved through constrained resampling while ensuring the diversity of particles.
2) We demonstrated the proposed uncertainty-constrained BP framework using FG. The fused information is interacted in a collaborative network to maintain computational efficiency. UCBP achieves significantly improved accuracy without introducing a high computational cost.

The remainder is organized as follows. Section II defines the collaborative network. Section III elaborates on the fusion BP algorithm based on uncertainty-constraint resampling. Section IV details UCBP and its FG representation. Section V conducts numerical simulations to verify the performance of the UCBP. Finally, conclusions are drawn in Section VI.

II. Problem Definition

In Section II-A, we first present a formal description of the problem discussed in this article. Then, some premises and assumptions are given in Section II-B. The measured distance between moving targets is used as the mutual information in the collaborative network. The moving target can also use the inertial sensor to obtain internal information, such as heading angle and step length in the moving process. Target nodes exchange data in real time and realize cooperative tracking.

A. Formal Description

A total of N nodes are defined in the collaboration network, which are N_T targets and N_A fixed nodes, i.e., anchors. Each target posseses a distance-ranging sensor measuring external information and an inertial sensor measuring internal information. The set of all nodes is defined as $\Omega_N = \{1, 2, 3, \ldots, N\}$, where the set of targets and anchors are Ω_T and Ω_A, respectively. Then, we denote $\Omega_N = \Omega_T + \Omega_A$. The motion process is represented as a sequence of discrete measurements at sufficiently small intervals, denoted as t_k, $k = 0, 1, 2, \ldots, K$. The mutual information between the nth target and others is $Z^{\text{mut}}_{1:K,n}$ $(n \in \Omega_T)$, and the internal is $Z^{\text{self}}_{1:K,n}$. Define the position and velocity of the target n at time t_k as $X_{k,n} = [x_{k,n}, y_{k,n}]^T$ and $V_{k,n} = [vx_{k,n}, vy_{k,n}]^T$, respectively.

Assume that the target conforms to the random walk model [31]. The position of target n at time t_k is defined as $X_{k,n}$, which can be transferred from the state $X_{k-1,n}$ at time t_{k-1}. The update process could be expressed as follows:

$$X_{k,n} = X_{k-1,n} + \hat{V}_{k,n} \cdot t_s \tag{1}$$

where t_s is the sampling time interval, and $\hat{V}_{k,n}$ indicates the velocity at time t_k, which could be measured by the accelerator, where

$$\hat{V}_{k,n} = V_{k,n} + \varepsilon_{1,k}, \varepsilon_{1,k} \sim N\left(\mu_{1,k}, {\sigma_{1,k}}^2\right) \tag{2}$$

where $V_{k,n}$ represents the actual velocity at time t_k, and $\varepsilon_{1,k}$ is a Gaussian distribution [30] with mean $\mu_{1,k}$ and variance $\sigma_{1,k}$. Next, the definitions of $\varepsilon_{2,k}$, $\varepsilon_{3,k}$, and $\varepsilon_{4,k}$ are consistent with $\varepsilon_{1,k}$. Therefore, the vector $\hat{V}_{1:K,n} = [\hat{V}_{1,n}, \hat{V}_{2,n}, \ldots, \hat{V}_{K,n}]^T$ indicates the velocity of target n over time series. Assume that t_s equals 1 s, and all targets are moving at a constant speed.

Denote $\hat{\theta}_{k,n}$ as the heading angle of target n measured by the gyroscope at time t_k, which is as follows:

$$\hat{\theta}_{k,n} = \theta_{k,n} + \varepsilon_{2,k}, \varepsilon_{2,k} \sim N\left(\mu_{2,k}, {\sigma_{2,k}}^2\right) \tag{3}$$

where $\theta_{k,n}$ represents the true horizontal angle at time t_k. Therefore, the vector $\hat{\theta}_{1:K,n} = [\hat{\theta}_{1,n}, \hat{\theta}_{2,n}, \ldots, \hat{\theta}_{K,n}]^T$ indicates the angle conditions of target n over time series.

Let $\hat{S}_{k,n}$ be the step length of target n at time t_k, which could be obtained by quadratic integration of acceleration, then

$$\hat{S}_{k,n} = S_{k,n} + \varepsilon_{3,k}, \varepsilon_{3,k} \sim N\left(\mu_{3,k}, {\sigma_{3,k}}^2\right) \tag{4}$$

where $S_{k,n}$ is the actual displacement of target n between t_{k-1} and t_k, i.e., $S_{k,n} = \|X_{k,n} - X_{k-1,n}\|$. Therefore, the vector $\hat{S}_{1:K,n} = [\hat{S}_{1,n}, \hat{S}_{2,n}, \ldots, \hat{S}_{K,n}]^T$ indicates the step length conditions of target n over time series.

In summary, the update process could be formulated as follows:

$$\begin{aligned} X_{k,n} &= X_{k-1,n} + f\left(\hat{\theta}_{k,n}, \hat{S}_{k,n}\right) \\ &= \begin{bmatrix} x_{k-1,n} \\ y_{k-1,n} \end{bmatrix} + \begin{bmatrix} \hat{S}_{k,n} \cdot \cos\left(\hat{\theta}_{k,n}\right) \\ \hat{S}_{k,n} \cdot \sin\left(\hat{\theta}_{k,n}\right) \end{bmatrix}. \end{aligned} \tag{5}$$

Finally, the inertial measurements could be denoted as $Z^{\text{self}}_{1:K,n} = [\hat{\theta}_{1:K,n}, \hat{S}_{1:K,n}]$.

At time t_k, the distance information between target n and another $s \in \Omega_N \backslash \{n\}$ is defined as $\hat{d}_{k,n,s}$, namely

$$\hat{d}_{k,n,s} = d_{k,n,s} + \varepsilon_{4,k}, \varepsilon_{4,k} \sim N\left(\mu_{4,k}, {\sigma_{4,k}}^2\right) \tag{6}$$

where $d_{k,n,s}$ is the actual distance between target n and another s at time t_k, i.e., $d_{k,n,s} = \|X_{k,s} - X_{k,n}\|$. Thus, $\hat{d}_{1:K,n,s} = [\hat{d}_{1,n,s}, \hat{d}_{2,n,s}, \ldots, \hat{d}_{K,n,s}]^T$ indicates the distance between nodes n and s over time series. Finally, the distance measurement outside the node, i.e., the mutual information between nodes is $Z^{\text{mut}}_{1:K,n} = \sum_{s\in\Omega_N\backslash\{n\}} \hat{d}_{1:K,n,s}$.

B. Premises and Assumptions

This article is to realize real-time target tracking with internal and mutual information fusion. A global function can be expressed as $f(\hat{X}_{1:K}, X_{0:K}, Z_{1:K})$, where $\hat{X}_{1:K}$ represents the posterior position information of all nodes from time t_1 to time t_k, $X_{0:K}$ is the prior position information, and $Z_{1:K}$ indicates all the measured information. The final estimate is the posterior position of the target node. Therefore, the global function can be expressed as follows:

$$\begin{aligned} f\left(\hat{X}_{1:K}, X_{0:K}, Z_{1:K}\right) &\triangleq p\left(\hat{X}_{1:K}|X_{0:K}, Z_{1:K}\right) \\ &= p\left(\hat{X}_{1:K}|X_{0:K}, Z^{\text{self}}_{1:k}, Z^{\text{mut}}_{1:k}\right). \end{aligned} \tag{7}$$

Based on the general definition in sensor networks [27], for better reasoning below, we present some reasonable premises and assumptions in the form of probability functions.

Premises:

1) The priori probability of target n's location: $p(X_{0,n})$.
2) Location of anchor m: $X_{m=1:N_A}$.
3) The motion model of target n at time t_k: $p(X_{k,n}|\hat{X}_{k-1,n})$.
4) The internal information of target n at time t_k: $p(Z^{\text{self}}_{1:k,n}|X_{k,n}, \hat{X}_{k-1,n})$.
5) The mutual information of target n at time t_k: $p(Z^{\text{mut}}_{1:k,n}|X_{k,n}, \hat{X}_{k-1,n})$.
6) The resampling of target n at time t_k: $p(\hat{X}_{k,n}|X_{k,n})$.

Assumptions:

1) Assume that the location of one anchor is fixed, and its corresponding probability should be set as 1. Therefore, all the following assumptions (except Assumption 6) are also valid for anchors. In the subsequent chapters, we treat the anchors and targets equally.
2) The movement of all target nodes conforms to the memoryless random walk model, and the resampling process conforms to the Markov-like property

$$p\left(\hat{X}_{1:k}, X_{0:k}\right) = p(X_0) \cdot \prod_{k=1}^{K}\left(p\left(X_k|\hat{X}_{k-1}\right) \cdot p\left(\hat{X}_k|X_k\right)\right). \tag{8}$$

3) The movement of the target is conditionally independent of each other

$$p\left(X_k|\hat{X}_{k-1}\right) = \prod_{n\in\Omega_T} p\left(X_{k,n}|\hat{X}_{k-1,n}\right). \tag{9}$$

4) The target nodes are conditionally independent of each other during the resampling process

$$p\left(\hat{X}_k|X_k\right) = \prod_{n\in\Omega_T} p\left(\hat{X}_{k,n}|X_{k,n}\right). \tag{10}$$

5) The mutual information between the targets during the movement only depends on their positions and has nothing to do with other information

$$p\left(Z^{\text{mut}}_{1:K}|\hat{X}_{1:K}, X_{0:K}, Z^{\text{self}}_{1:K}\right) = p\left(Z^{\text{mut}}_{1:K}|X_{0:K}\right). \tag{11}$$

6) The internal information of all targets is conditionally independent of each other and only depends on their locations at current time and the previous time

$$p\left(Z^{\text{self}}_{1:K}|\hat{X}_{1:K}, X_{0:K}\right) = \prod_{k=1}^{K} p\left(Z^{\text{self}}_{1:k}|X_k, \hat{X}_{k-1}\right). \tag{12}$$

7) The internal information of different targets at time t_k is conditionally independent of each other

$$p\left(Z^{\text{self}}_{1:k}|X_k, \hat{X}_{k-1}\right) = \prod_{n\in\Omega_T} p\left(Z^{\text{self}}_{1:k,n}|X_{k,n}, \hat{X}_{k-1,n}\right). \tag{13}$$

8) The mutual information of all target nodes is conditionally independent of each other at different times and only depends on their current positions

$$p\left(Z^{\text{mut}}_{1:K}|X_{0:K}\right) = \prod_{k=1}^{K} p\left(Z^{\text{mut}}_k|X_k\right). \tag{14}$$

9) The mutual information between any target node and the others at time t_k is conditionally independent of each other and only depends on their locations

$$p\left(Z^{\text{mut}}_k|X_k\right) = \prod_{n\in\Omega_T}\prod_{s\in\Omega_N\backslash\{n\}} p\left(\hat{d}_{k,n,s}|X_{k,n}, X_{k,s}\right). \tag{15}$$

III. Uncertainty-Constrained NBP Resampling

In this section, we first present the extended NBP using IMU/TOA fusion in Section III-A. Then, in Section III-B, a UCR method is demonstrated to solve the particle degradation and impoverishment.

A. IMU/TOA-Based Nonparametric Belief Propagation

NBP [26] uses a set of weighted particles $\{w^i_t, x^i_t\}$ to represent the posterior probability distribution of the target state. It uses spatial information, i.e., the distance between nodes, to update the particles' weights. Assuming that a collaborative network has N_T targets and N_A anchors. At time t_k, the positions of target $n \in \Omega_T$ and the other target $s \in \Omega_N\backslash\{n\}$ are $X_{k,n}$ and $X_{k,s}$, respectively. Then, the distance between them is denoted as follows:

$$\begin{aligned} \hat{d}_{k,n,s} &= \left\|X_{k,n} - X_{k,s}\right\| + \varepsilon_{4,k} \\ \varepsilon_{4,k} &\sim N\left(\mu_{4,k}, {\sigma_{4,k}}^2\right). \end{aligned} \tag{16}$$

The specific goal of target tracking is to estimate the maximum posterior position for a given set of observations. If the noise is Gaussian distribution and the probability of the measured distance is constant, the joint estimation becomes a nonlinear least squares optimization [15]. In this case, we assume a model where the probability of detecting nearby sensors decreases with the squared exponent [26]

$$P_o(X_{k,n}, X_{k,s}) = \exp\left(-\frac{\left\|X_{k,n} - X_{k,s}\right\|^2}{2R^2}\right). \tag{17}$$

First, at the initial time t_k, there is *a priori* state function $f(X_{0,n})$ for target n, and R weighted sample particles are obtained from the prior function. These R particles will be copied and filtered under the NBP and passed from time t_0 to t_k. Based on the above premise, at time t_k, there are R particles $\{w^i_{k,n}, X^i_{k,n}\}^{i=1:R}$ upon $X_{k,n}$ of target n. Next, we take target n as an instance to illustrate the interaction process between it and target $l \in \Omega_T \backslash \{n\}$, as well as anchors.

1) Interaction With Targets: First, we compute a Gaussian mixture estimate of the outgoing message $m^j_{k,n,l}$ from l to n. For any target l, it ranges distance with target n, but target l does not know in which direction target n is (under random walk model). Therefore, all sample particles $\{w^j_{k,l}, X^j_{k,l}\}^{j=1:R}$ of l move in a random direction by $\hat{d}_{k,n,l}$ plus a noise

$$m^j_{k,n,l} = x^j_{k,l} + \left(\hat{d}_{k,n,l} + v\right) \cdot \left[\cos(\theta^j); \sin(\theta^j)\right] \qquad (18)$$

where $\theta^j \sim U[0, 2\pi)$ and $v \sim N(0, \delta)$, namely, θ^j follows the uniform distribution, and v follows the Gaussian distribution. Each particle $X^i_{k,n}$ of target n is calculated on basis of $P_o(\cdot)$ [i.e., (17)]. Then, the particle weight is updated by

$$w^i_{k,n,l} = \left(\sum_{j=1}^{R} p_o\left(m^j_{k,n,l}, X^i_{k,n}\right)\right) \cdot w^i_{k,n}. \qquad (19)$$

Next, normalize the particle weights of target n

$$w^i_{k,n} = \frac{w^i_{k,n,l}}{\sum_{i=1}^{R} w^i_{k,n,l}}. \qquad (20)$$

Repeat the above process until all targets are traversed.

2) Interaction With Anchors: For any anchor $m \in \Omega_A$, it ranges distance $\hat{d}_{k,n,m}$ with target n. Since the location of the anchor is known, we directly measure the distance between the particle i of target n and the anchor $X_{k,m}$

$$d^i_{k,n,m} = \left\| X^i_{k,n} - X_{k,m} \right\|. \qquad (21)$$

Furthermore, as target n knows the location of anchor $X_{k,m}$, we could update the particle weight by

$$w^i_{k,n,m} = p_o\left(d^i_{k,n,m}, \hat{d}_{k,n,m}\right) \cdot w^i_{k,n}. \qquad (22)$$

Then, normalize the particle weights of target n. Repeat this for the next anchor until all anchors are traversed.

B. Uncertainty-Constrained Resampling

Particle degradation is one of the fundamental problems faced by PFs [32]. After a certain number of iterations, some particles' weights are too small that they should be ignored. Resampling can discard particles with small weights and copy those with large weights, effectively alleviating particle degradation. Standard resampling methods include systematic resampling, hierarchical resampling, residual resampling [32], etc. These algorithms are hierarchical filtering based on the weight threshold and are more efficient. However, during the iteration process, the reduction in the number of particles will lead the weights to be occupied by only a few particles, resulting in particle impoverishment. To this end, we proposed an uncertainty-constrained resampling to complete the redistribution of particle weights.

The particles are stratified by the confidence ellipse for resampling. The basic concept of confidence ellipse originates from the interval estimation of coordinate parameters [14]. When performing interval estimation on the scale η, for a given small probability β, if an interval (r_1, r_2) can be found such that $p(r_1 < \eta < r_2) = 1 - \beta$. Then, (r_1, r_2) is called the confidence interval of the scale η. For target n, its covariance matrix of R particles $\{w^i_{k,n}, X^i_{k,n} = (x^i_{k,n}, y^i_{k,n})\}^{i=1:R}$ at time t_k is defined as follows:

$$D = \begin{bmatrix} \mathrm{cov}\left(x_{k,n}, x_{k,n}\right) & \mathrm{cov}\left(x_{k,n}, y_{k,n}\right) \\ \mathrm{cov}\left(y_{k,n}, x_{k,n}\right) & \mathrm{cov}\left(y_{k,n}, y_{k,n}\right) \end{bmatrix} \qquad (23)$$

where $x_{k,n} = [x^1_{k,n}, x^2_{k,n}, \ldots, x^R_{k,n}]^T$, and $y_{k,n} = [y^1_{k,n}, y^2_{k,n}, \ldots, y^R_{k,n}]^T$. For each particle $X^i_{k,n}$, after introducing the error covariance matrix, the center of confidence ellipse is expressed as follows:

$$\eta = \frac{\left(x^i_{k,n} - x_p\right)^2}{\lambda_1} + \frac{\left(y^i_{k,n} - y_p\right)^2}{\lambda_2} \qquad (24)$$

where λ_1 and λ_2 are the largest and smallest eigenvectors corresponding to the covariance matrix D, respectively. (x_p, y_p) is the center of the particle set, and η is the scale of the confidence ellipse. Furthermore, when the ellipse is tilted, let the tilt angle between the ellipse's major axis and the x-axis be α, then

$$\alpha = \arctan\left(\frac{\lambda_1\left(y_{k,n}\right)}{\lambda_2\left(x_{k,n}\right)}\right). \qquad (25)$$

Suppose the rotated coordinate is (x', y'), then

$$\begin{cases} x' = x^i_{k,n} \cos\alpha - y^i_{k,n} \sin\alpha \\ y' = x^i_{k,n} \sin\alpha + y^i_{k,n} \cos\alpha. \end{cases} \qquad (26)$$

Finally, the confidence ellipse could be denoted as follows:

$$\eta = \frac{\left(\left(x' - x_p\right)\cos\alpha + \left(y' - y_p\right)\sin\alpha\right)^2}{\lambda_1} + \frac{\left(-\left(x' - x_p\right)\sin\alpha + \left(y' - y_p\right)\cos\alpha\right)^2}{\lambda_2}. \qquad (27)$$

In the resampling process, R particles are divided into three different levels according to the confidence r_1 and r_2, which are the probability that the particle will fall inside the inner and outer ellipse, respectively. The number of particles outside the outer ellipse is counted as N_l, which should be ignored. The particles in the middle of these two ellipses are defined as medium-weight particles, which should be reserved. Those N_h particles located in the inner ellipse will be replicated as follows. Copy the first $N_t = N_l - \lfloor N_l/N_h \rfloor \cdot N_h$ particles $c_1 = \lfloor N_l/N_h \rfloor + 2$ times, and the remaining $(N_h - N_t)$ particles $c_2 = \lfloor N_l/N_h \rfloor + 1$ times. After the particles to be screened is copied, the number of the particle set remains R. The selected particles will also be the initial input in next iteration. When the particle screening completed, the weights are redistributed

$$\hat{w}^i_{k,n} = \begin{cases} \hat{w}^i_{k,n}, & \eta^i_{k,n} < r_1 \\ \hat{w}^i_{k,n}, & r_1 \le \eta^i_{k,n} \le r_2 \\ 0, & \eta^i_{k,n} > r_2 \end{cases} \qquad (28)$$

where $\eta_{k,n}^{i}$ is the distance between the particle i and the predicted center of target n at t_k. Then, the number of particles remains unchanged, but the total weight has changed. Particle replication eventually leads to weight adjustment. For time t_k, the particle weight represents the *a priori* information of t_{k-1}. Here, the particle weight is reserved for filtering and copying. The subsequent weight changes are due to the normalization, reflecting the spatial information introduced during particle replication. Finally, the estimated state is updated by the weighted average

$$\begin{cases} x_{k,n} = \sum_{i=1}^{R} \hat{w}_{k,n}^{i} x_{k,n}^{i} \\ y_{k,n} = \sum_{i=1}^{R} \hat{w}_{k,n}^{i} y_{k,n}^{i}. \end{cases} \tag{29}$$

IV. Uncertainty-Constrained Belief Propagation and Its Factor Graph

In Sections IV-A and IV-B, we presented the global function decomposition, local function decomposition, and corresponding FGs of the problem described in Section II. Then, in Section IV-C, the message passing rules of UCBP are given based on BP. Finally, Section IV-D describes the particle-based local function message passing process. The UCBP combines NBP and UCR to utilize the ranging and inertial information. NBP takes advantage of distance information without cumulative errors, while UCR uses spatial–temporal optimized inertial information to estimate the target's position. Then, the UCBP realizes the propagation and fusion of distance and inertial information over the FG to solve collaborative issues.

A. Global Function Decomposition and Factor Graph

From Section II-B, we concluded that the problem sought in this article could be expressed by a global function $f(\hat{X}_{1:K}, X_{0:K}, Z_{1:K})$. It can also be denoted in the form of a posterior probability function, namely

$$\begin{aligned} p\left(\hat{X}_{1:K}|X_{0:K}, Z_{1:K}\right) &= p\left(\hat{X}_{1:K}|X_{0:K}, Z_{1:k}^{\text{self}}, Z_{1:k}^{\text{mut}}\right) \\ &= \frac{p\left(Z_{1:K}^{\text{mut}}|\hat{X}_{1:K}, X_{0:K}, Z_{1:K}^{\text{self}}\right) \cdot p\left(\hat{X}_{1:K}, X_{0:K}, Z_{1:K}^{\text{self}}\right)}{p\left(X_{0:K}, Z_{1:K}^{\text{self}}, Z_{1:K}^{\text{mut}}\right)}. \end{aligned} \tag{30}$$

Substituting (11) into (30), we get

$$\begin{aligned} &p\left(\hat{X}_{1:K}|X_{0:K}, Z_{1:K}\right) \\ &\quad \propto p\left(Z_{1:K}^{\text{mut}}|X_{0:K}\right) \cdot p\left(Z_{1:K}^{\text{self}}|\hat{X}_{1:K}, X_{0:K}\right) \cdot p\left(\hat{X}_{1:K}, X_{0:K}\right). \end{aligned} \tag{31}$$

Then, substituting (8), (12), and (14) into (31)

$$\begin{aligned} &p\left(\hat{X}_{1:K}|X_{0:K}, Z_{1:K}\right) \\ &\quad \propto p(X_0) \cdot \prod_{k=1}^{K} \left(p\left(Z_k^{\text{mut}}|X_k\right) \cdot p\left(\hat{X}_k|X_k\right)\right) \cdot \\ &\quad \prod_{k=1}^{K} \left(p\left(X_k|\hat{X}_{k-1}\right) \cdot p\left(Z_{1:k}^{\text{self}}|X_k, \hat{X}_{k-1}\right)\right). \end{aligned} \tag{32}$$

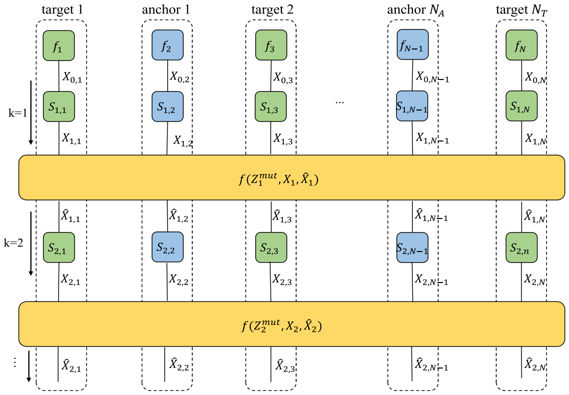


Fig. 1. FG of $p(\hat{X}_{1:\text{N}}|X_{0:N}, Z_{1:N})$ when $k = 0 : 2$. The following abbreviations are adopted for simplicity: $f_n(X_{0,n}) = p(X_{0,n})$, $S_{k,n}(X_{k,n}, \hat{X}_{k-1,n}) = p(X_{k,n}|\hat{X}_{k-1,n}) \cdot p(Z_{1:k,n}^{\text{self}}|X_{k,n}, \hat{X}_{k-1,n})$, and $f(Z_k^{\text{mut}}, X_k, \hat{X}_k) = p(Z_k^{\text{mut}}|X_k) \cdot p(\hat{X}_k|X_k)$.

Finally, substituting (9) and (13) into (32)

$$\begin{aligned} &p\left(\hat{X}_{1:K}|X_{0:K}, Z_{1:K}\right) \\ &\propto \prod_{n=1}^{N} \left(p(X_{0,n}) \prod_{k=1}^{K} \left(p\left(X_{k,n}|\hat{X}_{k-1,n}\right) \cdot p\left(Z_{1:k,n}^{\text{self}}|X_{k,n}, \hat{X}_{k-1,n}\right)\right)\right) \cdot \\ &\prod_{k=1}^{K} \left(p\left(Z_k^{\text{mut}}|X_k\right) \cdot p\left(\hat{X}_k|X_k\right)\right). \end{aligned} \tag{33}$$

Therefore, with factorization, the FG of (33) could be shown as Fig. 1.

B. Local Function Decomposition and Factor Graph

On the basis of Section IV-A, we could further factorize the local function $p(Z_k^{\text{mut}}|X_k) \cdot p(\hat{X}_k|X_k)$. Substituting (10) and (15) into $p(Z_k^{\text{mut}}|X_k) \cdot p(\hat{X}_k|X_k)$ then leads to

$$\begin{aligned} &p\left(Z_k^{\text{mut}}|X_k\right) \cdot p\left(\hat{X}_k|X_k\right) \\ &= \prod_{n\in\Omega_T} \left(p\left(\hat{X}_{k,n}|X_{k,n}\right) \cdot \prod_{s\in\Omega_N\backslash\{n\}} p\left(\hat{d}_{k,n,s}|X_{k,n}, X_{k,s}\right)\right) \end{aligned} \tag{34}$$

where $\hat{d}_{k,n,s}$ and $\hat{d}_{k,s,n}$ can be regarded as the mutual information between the target n and s. Thus, the content represented by it is equivalent. Therefore, $p(\hat{d}_{k,n,s}|X_{k,n}, X_{k,s})$ and $p(\hat{d}_{k,s,n}|X_{k,s}, X_{k,n})$ are regarded as the same function node $\varphi_{k,n,s}(X_{k,n}, X_{k,s})$ in the factorization. Based on (34), the FG of $p(Z_k^{\text{mut}}|X_k) \cdot p(\hat{X}_k|X_k)$ could be shown as Fig. 2.

C. Message Passing of the Global Function

In Sections IV-A and IV-B, we presented the FG of global function $f(\hat{X}_{1:K}, X_{0:K}, Z_{1:K})$. Next, based on the sum-product algorithm [22], how message passing over these FGs will be detailed, which mainly contains the message $\mu_{S_{k,n}\to X_{k,n}}(X_{k,n})$ passed by the function $S_{k,n}$ to edge $X_{k,n}$, and the message $\mu_{\hat{X}_{k,n}\to S_{k,n}}(\hat{X}_{k,n})$ passed by edge $\hat{X}_{k,n}$ to the function $S_{k+1,n}$.

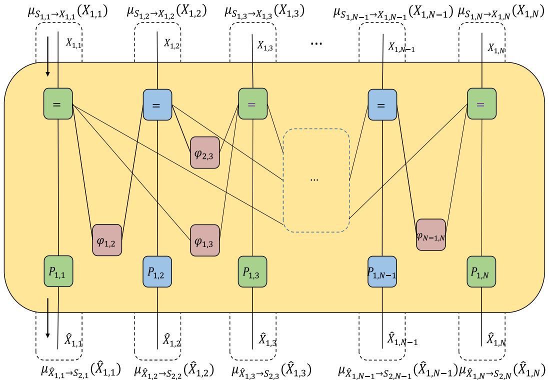

$\mu_{S_{1,1}\to X_{1,1}}(X_{1,1})$
$\mu_{S_{1,2}\to X_{1,2}}(X_{1,2})$
$\mu_{S_{1,3}\to X_{1,3}}(X_{1,3})$
$\mu_{S_{1,N-1}\to X_{1,N-1}}(X_{1,N-1})$
$\mu_{S_{1,N}\to X_{1,N}}(X_{1,N})$
$X_{1,1}$
$X_{1,2}$
$X_{1,3}$
…
$X_{1,N-1}$
$X_{1,N}$
=
=
=
=
=
$\varphi_{2,3}$
…
$\varphi_{1,2}$
$\varphi_{1,3}$
$\varphi_{N-1,N}$
$P_{1,1}$
$P_{1,2}$
$P_{1,3}$
$P_{1,N-1}$
$P_{1,N}$
$\hat{X}_{1,1}$
$\hat{X}_{1,2}$
$\hat{X}_{1,3}$
$\hat{X}_{1,N-1}$
$\hat{X}_{1,N}$
$\mu_{\hat{X}_{1,1}\to S_{2,1}}(\hat{X}_{1,1})$
$\mu_{\hat{X}_{1,2}\to S_{2,2}}(\hat{X}_{1,2})$
$\mu_{\hat{X}_{1,3}\to S_{2,3}}(\hat{X}_{1,3})$
$\mu_{\hat{X}_{1,N-1}\to S_{2,N-1}}(\hat{X}_{1,N-1})$
$\mu_{\hat{X}_{1,N}\to S_{2,N}}(\hat{X}_{1,N})$

Fig. 2. FG of $p(Z_k^{\mathrm{mut}}|X_k)\cdot p(\hat{X}_k|X_k)$, where $\varphi_{n,s}(X_{k,n},X_{k,s}) = p(\hat{d}_{k,n,s}|X_{k,n},X_{k,s})$, $P_{k,n} = p(\hat{X}_{k,n}|X_{k,n})$. $\mu_{S_{k,n}\to X_{k,n}}(\cdot)$ represents the message from function $S_{k,n}$ to edge $X_{k,n}$, and $\mu_{\hat{X}_{k,n}\to S_{k+1,n}}(\cdot)$ represents the message from edge $\hat{X}_{k,n}$ to function $S_{k+1,n}$.

Thereinto

$$\mu_{S_{k,n}\to X_{k,n}}\left(X_{k,n}\right) \propto \int S_{k,n}\left(X_{k,n}|\hat{X}_{k-1,n}\right)\times\mu_{\hat{X}_{k-1,n}\to S_{k,n}}\left(\hat{X}_{k-1,n}\right)dX_{k,n} \tag{35}$$

$$\mu_{\hat{X}_{k,n}\to S_{k,n}}\left(\hat{X}_{k,n}\right) = \mu_{P_{k,n}\to\hat{X}_{k,n}}\left(\hat{X}_{k,n}\right). \tag{36}$$

As shown in Fig. 2, for each node at time t_k, it mainly consists of the message $\mu_{X_{k,n}\to\varphi_{k,n,s}}(X_{k,n})$ by the edge $X_{k,n}$ to function $\varphi_{k,n,s}$, the message $\mu_{\varphi_{k,n,s}\to X_{k,n}}(X_{k,n})$ by the function $\varphi_{k,n,s}$ to edge $X_{k,n}$, the message $\mu_{X_{k,n}\to P_{k,n}}(X_{k,n})$ by the edge $X_{k,n}$ to function $P_{k,n}$, and the message $\mu_{P_{k,n}\to\hat{X}_{k,n}}(\hat{X}_{k,n})$ by the function $P_{k,n}$ to edge $\hat{X}_{k,n}$. As the FG in Fig. 2 is loopy, there is generally a given number of cycles or termination conditions. Thus, we assume that the number of cycles is T. Let $b^0_{X_{k,n}}(X_{k,n}) = \mu_{S_{k,n}\to X_{k,n}}(X_{k,n})$, then for $t = 1 : T$

$$\mu^t_{X_{k,n}\to\varphi_{k,n,s}}(X_{k,n}) = b^{t-1}_{X_{k,n}}(X_{k,n}) \tag{37}$$

$$\mu^t_{\varphi_{k,n,s}\to X_{k,n}}(X_{k,n}) \propto \int p\left(\hat{d}_{k,n,s}|X_{k,n},X_{k,s}\right) \times \mu^t_{X_{k,n}\to\varphi_{k,n,s}}\left(X_{k,n}\right)dX_{k,n} \tag{38}$$

$$b^t_{X_{k,n}}(X_{k,n}) = \int \mu_{S_{k,n}\to X_{k,n}}(X_{k,n}) \times \prod_{s\in\Omega_N\backslash\{n\}} \mu^t_{\varphi_{k,n,s}\to X_{k,n}}(X_{k,n})dX_{k,n}. \tag{39}$$

At the end of the loop, we can get

$$\mu_{X_{k,n}\to P_{k,n}}(X_{k,n}) = b^T_{X_{k,n}}(X_{k,n}) \tag{40}$$

$$\mu_{P_{k,n}\to\hat{X}_{k,n}}\left(\hat{X}_{k,n}\right) = \int p\left(\hat{X}_{k,n}|X_{k,n}\right)\times\mu_{X_{k,n}\to P_{k,n}}(X_{k,n})d\hat{X}_{k,n}. \tag{41}$$

Based on these rules, the overall message passing for $f(\hat{X}_{1:K},X_{0:K},Z_{1:K})$ is given in Algorithm 1, while the visualized message passing process is detailed in Fig. 3.

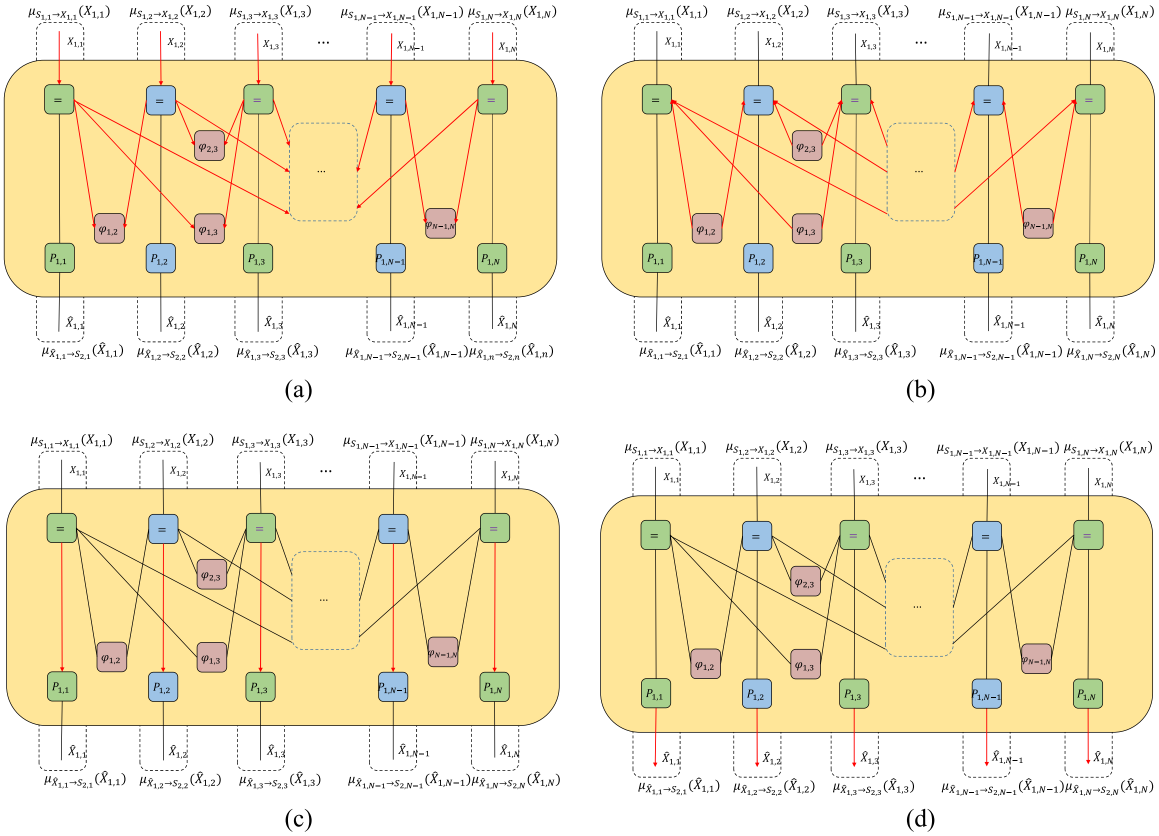
Algorithm 1 Message Passing of $f(\hat{X}_{1:K},X_{0:K},Z_{1:K})$

```
Input: $\{p(X_{0,n})\}_{n\in\Omega_T}$ ← initial position information of all nodes
Output: $\{\mu_{\hat{X}_{k,n}\to S_{k,n}}(\hat{X}_{k,n})\}^{k=1:K}_{n\in\Omega_T}$ ← posterior location information
  of all nodes
  for $k \leftarrow 1,2,\ldots,K$ do
      if $k = 0$ then
          $\mu_{\hat{X}_{0,n}\to S_{1,n}}(\hat{X}_{0,n}) = \mu_{X_{0,n}\to S_{1,n}}(X_{0,n}) = p(X_{0,n})$
      else
          for $n \in \Omega_T$ do
              $\mu_{S_{k,n}\to X_{k,n}}(X_{k,n}) \propto \int S_{k,n}(X_{k,n}|\hat{X}_{k-1,n})$
              $\times\,\mu_{\hat{X}_{k-1,n}\to S_{k,n}}(\hat{X}_{k-1,n})dX_{k,n}$
              $b^0_{X_{k,n}}(X_{k,n}) = \mu_{S_{k,n}\to X_{k,n}}(X_{k,n})$
              for $t \leftarrow 1,2,\ldots,T$ do
                  for $s \leftarrow 1,2,\ldots,N$ do
                      if $n \in \Omega_N$ and $s \neq n$ and $\{i,j\} \not\subset \Omega_A$ then
                          $\mu^t_{X_{k,n}\to\varphi_{k,n,s}}(X_{k,n}) = b^{t-1}_{X_{k,n}}(X_{k,n})$
                      end if
                      if $n \in \Omega_T$ and $s \neq n$ then
                          $\mu^t_{\varphi_{k,n,s}\to X_{k,n}}(X_{k,n})$
                          $\propto \int p(\hat{d}_{k,n,s}|X_{k,n},X_{k,s})$
                          $\times\,\mu^t_{X_{k,n}\to\varphi_{k,n,s}}(X_{k,n})dX_{k,n}$
                      end if
                      if $n \in \Omega_T$ then
                          $b^t_{X_{k,n}}(X_{k,n})$
                          $= \int \mu_{S_{k,n}\to X_{k,n}}(X_{k,n})$
                          $\times \prod_{s\in Q} \mu^t_{\varphi_{k,n,s}\to X_{k,n}}(X_{k,n})dX_{k,n}$
                          where, $Q = \Omega_N\backslash\{n\}$
                      else
                          $b^t_{X_{k,n}}(X_{k,n}) = b^0_{X_{k,n}}(X_{k,n})$
                      end if
                  end for
              end for
              $\mu_{X_{k,n}\to P_{k,n}}(X_{k,n}) = b^T_{X_{k,n}}(X_{k,n})$
              $\mu_{P_{k,n}\to\hat{X}_{k,n}}(\hat{X}_{k,n})$
              $= \int p(\hat{X}_{k,n}|X_{k,n})\times\mu_{X_{k,n}\to P_{k,n}}(X_{k,n})d\hat{X}_{k,n}$
              $\mu_{\hat{X}_{k,n}\to S_{k,n}}(\hat{X}_{k,n}) = \mu_{P_{k,n}\to\hat{X}_{k,n}}(\hat{X}_{k,n})$
          end for
      end if
  end for
```

D. Particulate Realization of the Local Function Message

Based on (34), we could decompose $p(Z_k^{\mathrm{mut}}|X_k)\cdot p(\hat{X}_k|X_k)$ into the FG shown in Fig. 2. The messages passing by variable node $X_{k,n}$ and function node $\varphi_{k,n,s}(X_{k,n},X_{k,s})$ are presented in Section IV-C. Next, based on the particle sequential sum-product framework [33], we further derived the particle-based message passing rules for variable node $X_{k,n}$ and function node $\varphi_{k,n,s}(X_{k,n},X_{k,s})$.

1) Particulation of $\varphi_{k,n,s}(X_{k,n},X_{k,s})$: We take the function node $\varphi_{1,1,s}(X_{1,s},X_{1,1})$ of $X_{1,1}$ and any node $X_{1,s}$ in Fig. 2 as an example to visualize the message passing process. To easily understand the relationship between variable and function nodes, we convert the standard FG in Fig. 2 to a Forney


$\mu_{S_{1,1}\to X_{1,1}}(X_{1,1})$
$\mu_{S_{1,2}\to X_{1,2}}(X_{1,2})$
$\mu_{S_{1,3}\to X_{1,3}}(X_{1,3})$
…
$\mu_{S_{1,N-1}\to X_{1,N-1}}(X_{1,N-1})$
$\mu_{S_{1,N}\to X_{1,N}}(X_{1,N})$
$X_{1,1}$
$X_{1,2}$
$X_{1,3}$
$X_{1,N-1}$
$X_{1,N}$
=
$\varphi_{2,3}$
$\varphi_{1,2}$
$\varphi_{1,3}$
$\varphi_{N-1,N}$
…
$P_{1,1}$
$P_{1,2}$
$P_{1,3}$
$P_{1,N-1}$
$P_{1,N}$
$\hat{X}_{1,1}$
$\hat{X}_{1,2}$
$\hat{X}_{1,3}$
$\hat{X}_{1,N-1}$
$\hat{X}_{1,N}$
$\mu_{\hat{X}_{1,1}\to S_{2,1}}(\hat{X}_{1,1})$
$\mu_{\hat{X}_{1,2}\to S_{2,2}}(\hat{X}_{1,2})$
$\mu_{\hat{X}_{1,3}\to S_{2,3}}(\hat{X}_{1,3})$
$\mu_{\hat{X}_{1,N-1}\to S_{2,N-1}}(\hat{X}_{1,N-1})$
$\mu_{\hat{X}_{1,N}\to S_{2,N}}(\hat{X}_{1,N})$

(a)

(b)

(c)

(d)

Fig. 3. Message passing process of $p(Z_1^{\text{mut}}|X_1)\boldsymbol{\cdot}p(\hat{X}_1|X_1)$ is detailed from steps (a) to (d).

one. Based on the message passing rules of the sum-product algorithm over the FG, the rules for passing function nodes to variable nodes are

$$\mu_{f\to x}(x)=\sum_{\sim\{x\}}\left(f(X)\prod_{y\in n(f)\backslash\{x\}}\mu_{y\to f}(y)\right) \tag{42}$$

where $n(f)$ is the parameter set of f, and $\sim\{x\}$ represents all the variables except x in $n(f)$. For the function node described in Fig. 4(a), $X_{1,s}$ could be either a mobile target or an anchor, while the corresponding particle-based messaging process is inconsistent. Therefore, both conditions are considered.

If $X_{1,s}$ is a *target* node, the message (42) becomes

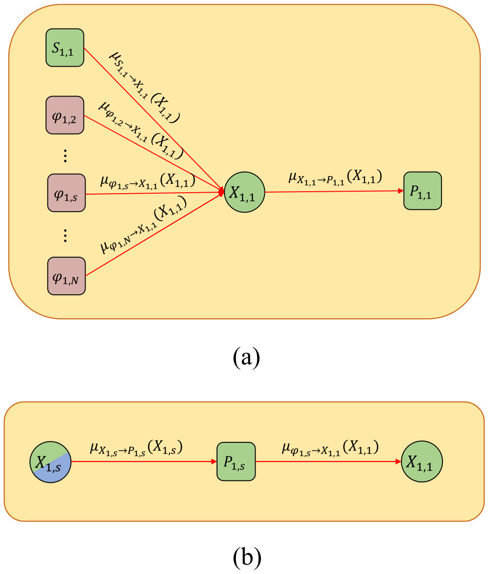
$$\mu_{\varphi_{1,1,s}\to X_{1,1}}(X_{1,1}) = \int \varphi_{1,1,s}(X_{1,s},X_{1,1}) \times \mu_{X_{1,s}\to\varphi_{1,1,s}}(X_{1,s})dX_{1,s}. \tag{43}$$

Let $H(X_{1,s})=\mu_{X_{1,s}\to\varphi_{1,1,s}}(X_{1,s})$, then (43) could be transformed into

$$\mu_{\varphi_{1,1,s}\to X_{1,1}}(X_{1,1}) = \int \varphi_{1,1,s}(X_{1,s},X_{1,1})\boldsymbol{\cdot}H(X_{1,s})dX_{1,s}. \tag{44}$$

Apparently, it just conforms the Monte Carlo integration form, thus we regard $H(X_{1,s})$ as the probability distribution function of $\varphi_{1,1,s}(X_{1,s},X_{1,1})$. Therefore, (44) could be transformed to the mean value form using the Monte Carlo integration

$$\mu_{\varphi_{1,1,s}\to X_{1,1}}(X_{1,1}) = E\left[\varphi_{1,1,s}(X_{1,s},X_{1,1})\right]$$


$S_{1,1}$
$\mu_{S_{1,1}\to X_{1,1}}(X_{1,1})$
$\varphi_{1,2}$
$\mu_{\varphi_{1,2}\to X_{1,1}}(X_{1,1})$
$\varphi_{1,s}$
$\mu_{\varphi_{1,s}\to X_{1,1}}(X_{1,1})$
$\varphi_{1,N}$
$\mu_{\varphi_{1,N}\to X_{1,1}}(X_{1,1})$
$X_{1,1}$
$\mu_{X_{1,1}\to P_{1,1}}(X_{1,1})$
$P_{1,1}$
$X_{1,s}$
$\mu_{X_{1,s}\to P_{1,s}}(X_{1,s})$
$P_{1,s}$
$\mu_{\varphi_{1,s}\to X_{1,1}}(X_{1,1})$
$X_{1,1}$

(a)

(b)

Fig. 4. Message passing process of function node $\varphi_{1,1,s}(X_{1,s},X_{1,1})$ and variable node $X_{1,1}$ is detailed from steps (a) to (b). The FG is converted from the standard form to the Forney [29], where $\varphi_{1,1,n}$ is denoted as $\varphi_{1,n}$ for simplicity.

$$= \frac{1}{R}\sum_{i=1}^{R}\varphi_{1,1,s}(X_{1,s}^{i},X_{1,1}) \tag{45}$$

where $X_{1,1}^{i}$ is the sample extracted from the probability distribution function $H(X_{1,1})$.

With the use of importance sampling, the message $\mu_{\varphi_{1,1,s}\rightarrow X_{1,1}}(X_{1,1})$ can be expressed as particle-based information $\{w_{1,1}^{i}, X_{1,1}^{i}\}^{i=1:R}$. The particles are generated by the initial state function $f_n(X_{0,n})$ and transferred to the current state by $S_{k,n}(X_{k,n}, \hat{X}_{k-1,n})$. It can be viewed as importance sampling of the current state function of $X_{1,s}$, namely

$$\mu_{\varphi_{1,1,s}\rightarrow X_{1,1}}(X_{1,1}) \cong \left\{w_{1,1}^{i}, X_{1,1}^{i}\right\}^{i=1:R} \tag{46}$$

where $X_{1,1}^{i} = X_{0,1}^{i}$, and $w_{1,1}^{i} = ([\sum_{j=1}^{R}\varphi_{1,1,s}(X_{1,s}^{j}, X_{1,1}^{i})]/[\sum_{i=1}^{R}\sum_{j=1}^{R}\varphi_{1,1,s}(X_{1,s}^{j}, X_{1,1}^{i})])$. The particle weights are determined accordingly.

If $X_{1,s}$ is an *anchor* node, the message (46) becomes

$$\mu_{\varphi_{1,1,s}\rightarrow X_{1,1}}(X_{1,1}) = \int \varphi_{1,1,s}(X_{1,s}, X_{1,1}) \times \mu_{X_{1,s}\rightarrow\varphi_{1,1,s}}(X_{1,s})dX_{1,s}. \tag{47}$$

Then, the information transmitted by $\mu_{X_{1,s}\rightarrow\varphi_{1,1,s}}(X_{1,s})$ is the coordinates $\bar{X}_{1,s}$ of the corresponding anchor. At the same time, the expression is transformed into

$$\mu_{\varphi_{1,1,s}\rightarrow X_{1,1}}(X_{1,1}) = \varphi_{1,1,s}\big(\bar{X}_{1,s}, X_{1,1}\big). \tag{48}$$

Similarly, using the importance sampling to construct the particle-based message, we could get

$$\mu_{\varphi_{1,1,s}\rightarrow X_{1,1}}(X_{1,1}) \cong \left\{w_{1,1}^{i}, X_{1,1}^{i}\right\}^{i=1:R} \tag{49}$$

where $X_{1,1}^{i} = X_{0,1}^{i}$, and $w_{1,1}^{i} = ([\varphi_{1,1,s}(\bar{X}_{1,s}, X_{1,1}^{i})]/[\sum_{i=1}^{R}\varphi_{1,1,s}(\bar{X}_{1,s}, X_{1,1}^{i})])$.

2) *Particulation of $X_{k,n}$:* We take $X_{1,1}$ in Fig. 2 as an example to draw the message passing process. Based on the sum-product algorithm, the message passing from variable nodes to the function is as follows:

$$\mu_{x\rightarrow f}(x) = \prod_{h\in n(x)\backslash\{f\}} \mu_{h\rightarrow x}(x) \tag{50}$$

where $n(x)$ is a collection of function nodes with x.

It is important to note that the message $\mu_{X_{1,1}\rightarrow P_{1,1}}(X_{1,1})$ will only be passed after the UCBP iteratively converges, as shown in Fig. 4(b). In the iterative process, Fig. 4(b) represents the message passed to the function node $\varphi_{1,1,n}$, while it is the same as the message passed to the function node $P_{1,1}$. Therefore, the method described below can also be applied to the message passed to the function node $\varphi_{1,1,n}$.

In the situation of Fig. 4(b), the message (50) becomes

$$\mu_{X_{1,1}\rightarrow P_{1,1}}(X_{1,1}) = \mu_{S_{1,1}\rightarrow X_{1,1}}(X_{1,1}) \boldsymbol{\cdot} \prod_{s=2}^{N} \mu_{\varphi_{1,1,s}\rightarrow X_{1,1}}(X_{1,1}) \tag{51}$$

where $\mu_{S_{1,1}\rightarrow X_{1,1}}(X_{1,1})$ and $\mu_{\varphi_{1,1,s}\rightarrow X_{1,1}}(X_{1,1})$ can be expressed as particle-based message $\{w_{1,1}^{i}(f), X_{1,1}^{i}\}^{i=1:R}$. Namely, $f = \{S_{1,1}, \varphi_{1,1,2}, \ldots, \varphi_{1,1,N}\}$. The particle in the above message represents the position of $X_{1,1}$, so we only change its weight on the variable node. The particles can be expressed nonparametrically as the kernel density estimation (KDE) $N(X_{1,1}; X_{1,1}^{i}, \Lambda)$ [26] (for convenience, we assume the particles of each node come from the same distribution, i.e., their variance fits Λ).

Algorithm 2 UCBP

```
Input: $\{X_{0,n}\}_{n\in\Omega_T}$ ← priori information about the initial position
  of all target nodes n
      $\{X_m\}_{m\in\Omega_A}$ ← location information of all anchors m
Output: $\{\hat{X}_{k,n}\}_{k=1:K,n\in\Omega_T}$ ← posterior position information of all
  target nodes n during the movement
  for $k \leftarrow 0, 1, \ldots, K$ do
    for $n \in \Omega_T$ do
      if $k = 0$ then
        $\{X_{1,n}^{i}\}^{i=1:R} \sim N(X_{0,n}, \delta_x)$
        $\{w_{1,n}^{i}\}^{i=1:R} \sim N(\mu_w, \delta_w)$
      else
        for $i \leftarrow 0, 1, \ldots, R$ do
          $X_{k,n}^{i} \leftarrow f_1(Z_{k,n}^{\text{self}}, X_{k,m}^{i}, \hat{X}_{k-1,n}^{i})$ ▷ state transition
  function
        end for
        for $t \leftarrow 0, 1, \ldots, T$ do
          $\{w_{k,n}^{i}, X_{k,n}^{i}\}^{i=1:R} \leftarrow$
$f_2(\{w_{k,n}^{i}, X_{k,n}^{i}\}_{n\in\Omega_T}^{i=1:R}, \{X_m\}_{m\in\Omega_A}, \{d_{k,n,l}\}_{l\in\Omega_N\backslash\{n\}})$ ▷ NBP
  Algorithm
          $(x_p, y_p) \leftarrow \sum_{i=1}^{R}\left(w_{k,n}^{R} \boldsymbol{\cdot} X_{k,n}^{R}\right)$
          $\{\hat{w}_{k,n}^{i}, \hat{X}_{k,n}^{i}\}^{i=1:R} \leftarrow$
  $UCR(\{w_{k,n}^{i}, X_{k,n}^{i}\}^{i=1:R}, (x_p, y_p))$ ▷ UCR Algorithm
          $\{w_{k,n}^{i}, X_{k,n}^{i}\}^{i=1:R} \leftarrow \{\hat{w}_{k,n}^{i}, \hat{X}_{k,n}^{i}\}^{i=1:R}$
          if $t = T$ then
            $\{w_{k+1,n}^{i}\}^{i=1:R} \leftarrow \{\hat{w}_{k,n}^{i}\}^{i=1:R}$
          end if
        end for
        $\hat{X}_{k,n} \leftarrow \sum_{i=1}^{R}\left(\hat{w}_{k,n}^{i} \boldsymbol{\cdot} \hat{X}_{k,n}^{i}\right)$
      end if
    end for
  end for
```

The particulation $\{w_{1,1}^{i}(S_{1,1}), X_{1,1}^{i}\}^{i=1:R}$ of the message $\mu_{S_{1,1}\rightarrow X_{1,1}}(X_{1,1})$ represents the prior state of node $X_{1,1}$, while the particulation $\{w_{1,1}^{i}(\varphi_{1,1,s}), X_{1,1}^{i}\}^{i=1:R}$ of the message $\mu_{\varphi_{1,1,s}\rightarrow X_{1,1}}(X_{1,1})$ is expressed as the state of $X_{1,1}$ after the propagation of function $\varphi_{1,1,s}$. Finally, we bring each particle's information for $X_{1,1}$ into (51), then lead to

$$\mu_{X_{1,1}\rightarrow P_{1,1}}(X_{1,1}^{i}) = N\big(X_{1,1}; X_{1,1}^{i}, \Lambda\big) \times w_{1,1}^{i}(S_{1,1}) \prod_{s=2}^{N} w_{1,1}^{i}\big(\varphi_{1,1,s}\big). \tag{52}$$

Above all, we detailed the massage passing of the proposed UCBP. In summary, the pseudocode is given in Algorithm 2.

V. Experimental Results and Analysis

A. Experimental Environment

In this article, we use MATLAB to carry out numerical simulations based on the random walk of the targets. The computer used is configured with Windows 10 operating system, 4-core i5 CPU, and 8-GB memory. The target starts from the initial position in each simulation and walks 60 steps randomly. Experimental parameters are detailed in Table I.

Due to the difference in the number of anchors N_A, the following two typical scenarios are considered in Sections V-B and V-C: 1) multitarget tracking with external

TABLE I
PARAMETER SETTING

Parameter	Value
Δt_s	1(s)
V	1.5(m/s)
a	0(m/s^2)
$\mu_{2,k}$	1(°)
$\sigma_{2,k}$	1(°)
$\mu_{3,k}$	0.02(m)
$\sigma_{3,k}$	0.1(m)
$\mu_{4,k}$	0.1(m)
$\sigma_{4,k}$	0.04(m)
$1-\beta_1$	0.8
$1-\beta_2$	0.125
Num	1000

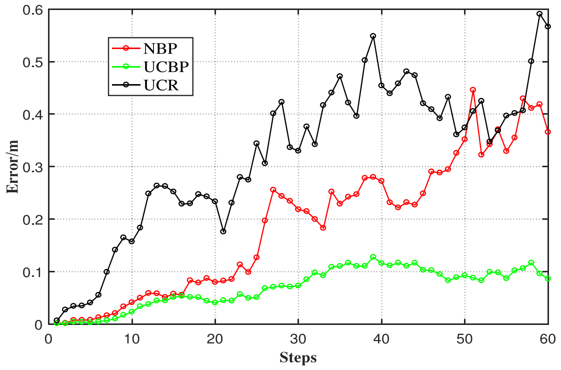


Fig. 5. Error distribution diagrams of different algorithms under multitarget tracking.

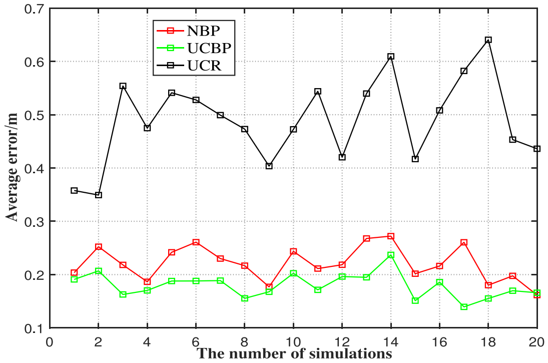


Fig. 6. RMSE of different algorithms in multitarget tracking.

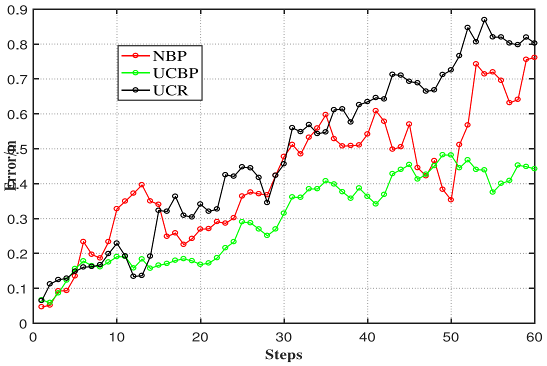


Fig. 7. Error distribution diagrams of different algorithms under multitarget tracking without anchors.

anchors ($N_A > 0$) and 2) multitarget without external anchors ($N_A = 0$). In each scenario, we compare our proposed UCBP with NBP [26] and UCR [14]. In Section V-D, we further analyze the parameter performance of the proposed UCBP.

B. Multitarget Tracking With Anchors ($N_A > 0$)

In this experiment, $N_T = 3$ and $N_A = 2$. The target node was guided to randomly walk 60 steps from the initial position, and its movement trajectory $\{(x_i, y_i)\}_{i=1:60}$ was recorded as the ground truth. Then, we recorded the inertial measurements of each step, and the target measured distances with all anchors at each moment. Finally, we used the above information to comprehensively estimate the position $(\hat{x}_i, \hat{y}_i)$. The root-mean-square error (RMSE) between the actual and predicted position is used as the evaluation metric, namely

$$e_i = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\left(\hat{x}_{i,n} - x_{i,n}\right)^2 + \left(\hat{y}_{i,n} - y_{i,n}\right)^2\right)} \tag{53}$$

where i indicates the current time step, and N is the total number of targets.

As shown in Fig. 5, UCBP combines distance and inertial information to obtain the highest positioning accuracy, effectively suppressing the speed of error accumulation. In addition, NBP also gets relatively high accuracy. Its estimation mainly relies on distance measurements, which do not accumulate errors. The UCR is based on inertial constraint optimization, and it slows down the error accumulation to a certain extent. However, the accumulation issue still exists. UCBP uses UCR to resample the particles and alleviate the particle degradation in NBP. At the same time, UCR uses inertial information to filter particles to mitigate the loss of particle diversity caused by multiple sampling of particles with large weights in NBP. We repeated the simulation 20 times and calculated the mean square error, shown in Fig. 6. UCBP maintains the highest accuracy, which verifies its superiority in multitarget tracking.

C. Multitarget Tracking Without Anchors ($N_A = 0$)

In this experiment, $N_T = 3$ and $N_A = 0$. The other parameters remain consistent with Section V-B. As shown in Fig. 7, UCBP maintains the highest accuracy when canceling all anchors in the collaborative network. Then, we repeated the experiment 20 times, and UCBP maintained the highest accuracy, as shown in Fig. 8. Although the loss of anchor points reduces the confidence of distance information, that of the inertial information remains. UCBP corrects the error caused by canceling anchors through constraint optimization based on the fusion of distance and inertial information.

D. Parameter Analysis

The number of particles may significantly impact the performance and execution time of the PF. Thus, this section will conduct numerical simulations to assess how the number

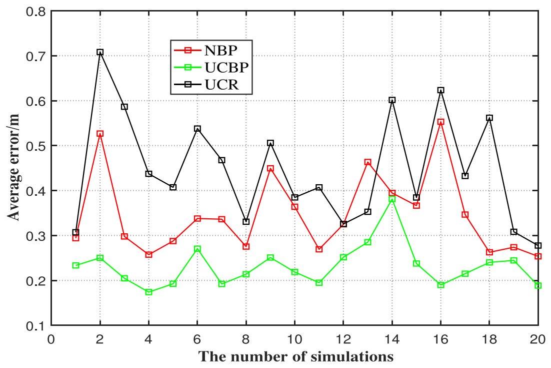


Fig. 8. RMSE of different algorithms in multitarget tracking without anchors.

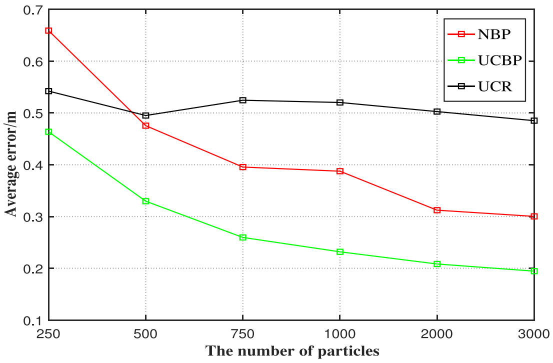


Fig. 9. Average RMSE conditions under different particle numbers.

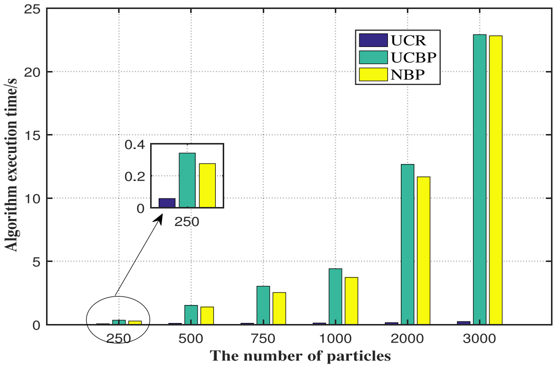


Fig. 10. Execution time under different particle numbers.

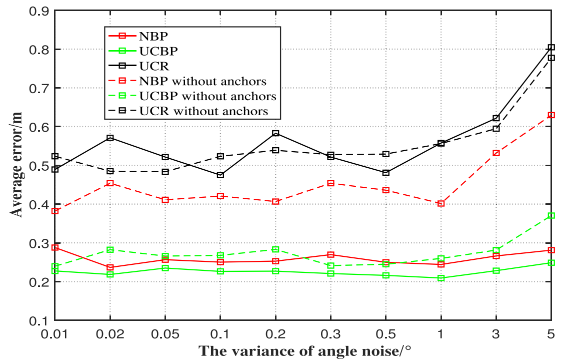


Fig. 11. Positioning errors under various angle variances.

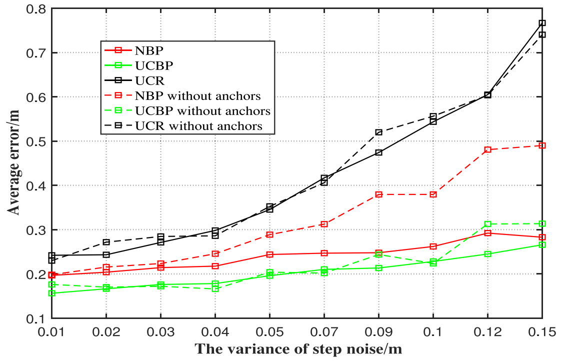


Fig. 12. Positioning errors under various step variances.

of particles affects the performance. In addition, various environment parameters are considered to verify the robustness of UCBP and related issues.

1) Number of Particles: We take the scenario in Section V-C to evaluate the impact of the number of particles on the accuracy and the execution time. Then, the optimal number of particles could be obtained and, at the same time, verify the execution efficiency of UCBP.

The selection range of particles' number is set as: Num = {250, 500, 1000, 1500, 2000, 2500, 3000}. The error plotted in Fig. 9 is the average of RMSE in 20 simulation runs in Section V-C. The execution time of each algorithm in Fig. 10 is the sum of 60 steps taken by all N target nodes.

1) As shown in Fig. 9, when the number of particles is small, NBP suffers from particle degradation and impoverishment, leading to excessive errors. When the number of particles is between 750 and 3000, all these three algorithms maintain a relatively stable trend, and the relative position remains unchanged.
2) From Fig. 10, we conclude that the optimal number of particles selected is between 750 and 1000, while the execution time of NBP and UCBP does not change much. Ranging from 1000 to 3000 particles, the algorithm's execution time has increased significantly. It is a more reasonable choice to consider comprehensively to set the particles' number between 750 and 1000.
3) From (18) and (19), we know that each particle of any target node in NBP is calculated with all particles of the others, so the time complexity of NBP is $O(n^2)$. From (28) and (29), we know that UCR resamples each particle of any target node, so its time complexity is $O(n)$. UCBP combines NBP and UCR, so its time complexity is consistent with NBP. It is also confirmed by the results shown in Fig. 10. It is worth mentioning that, compared with NBP, UCBP has obtained a significantly improved accuracy without introducing much computational cost.

2) Robustness Verification: This section first compares six algorithms under the two scenarios in Sections V-B and V-C as a comparative experiment. In this way, it is possible to judge whether the UCBP is the optimal choice for positioning under various conditions. The evaluation metric of the errors in Figs. 11–13 is the same as that in Fig. 10, and both are the

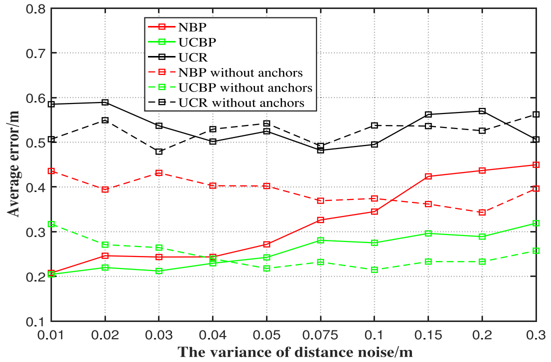


Fig. 13. Positioning errors under various distance variances.

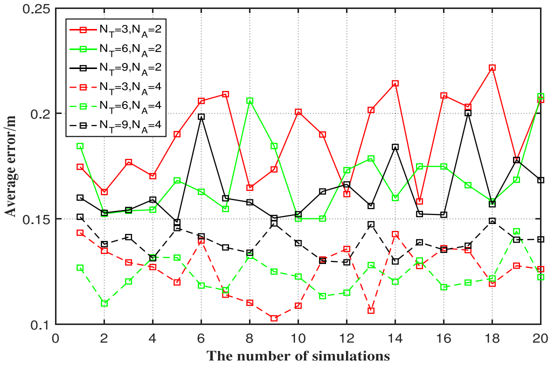


Fig. 14. Positioning errors under different numbers of target nodes and anchors. N_T represents the number of targets, and N_A represents the number of anchors.

average errors of 20 simulation rounds. We could draw the following conclusions.

1) UCBP maintains the best performance with or without external anchors. As shown in Figs. 11–13, when one single variance increases, the performance of UCBP remains the best. The reason is that the information used by UCBP is a fusion of multiple sources, e.g., heading angle, step length, and distances. Whenever the uncertainty of one source increases, that of the others remains unchanged. With fusion, the uncertainty could be substantially decreased to guarantee accuracy.
2) In Fig. 11, when the angle's variance is small, the "NBP with anchors" error is smaller than that of "UCBP without anchors." Therefore, when the uncertainty of the angle and distance is small, the positioning accuracy improvement brought by the distance alone is greater than that caused by the fusion of inertial and distance.
3) In Fig. 12, when the step size's variance is large, the error of "NBP" is smaller than UCBP without anchors. Likewise, in Fig. 13, when the distance variance is small, the error of NBP is also smaller than UCBP without anchors. The above phenomena all reflect that when the uncertainty of inertial information is much greater than that of distance, the fusion of inertial and distance information's confidence is lower than that of single-source distance. Above all, it can be concluded that the accuracy of multisource information fusion is not necessarily higher than that of one single source. This may have a guiding significance in practical multisource fusion applications, such as equipment selections.
4) In Fig. 14, when the number of anchors remains unchanged, the increase of target nodes' number has little impact on the errors. While keeping the number of target nodes unchanged, the number of anchors decreases the errors, making the positioning more accurate and consistent with common sense. UCBP maintains high accuracy under different numbers of nodes.

VI. Conclusion

This article proposes an uncertainty-constrained BP algorithm using IMU/TOA fusion for cooperative target tracking. We introduced uncertainty-constrained optimization into BP to address particle degradation and impoverishment. Besides, we demonstrate the FG framework of UCBP to mitigate the accumulated errors through message fusion over the graph. Furthermore, we defined the message passing rules of the UCBP to merge and propagate multisource messages. We further derived particle-based message passing regulations based on the particle sequential sum-product framework. The fused information is interacted in a collaborative network to maintain computational efficiency. By numerical simulations, UCBP performs better than state-of-the-art methods, with or without external anchors. At the same time, we conduct ablation experiments to verify the robustness of the UCBP. Simulations show that the UCBP has significantly improved accuracy without introducing much computational cost. Nevertheless, there is not much difference between its computational efficiency and NBP's. In our follow-on work, we may focus on reducing the computational overhead as much as possible while maintaining high precision.

References

[1] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, Apr. 2018, doi: 10.1109/JIOT.2018.2812300.

[2] R. P. Jain, J. B. de Sousa, and A. P. Aguiar, "Three dimensional moving path following control for robotic vehicles with minimum positive forward speed," in *Proc. Amer. Control Conf.*, New Orleans, LA, USA, 2021, pp. 250–255, doi: 10.23919/ACC50511.2021.9482784.

[3] H. Wen, J. Wu, Y. Duan, W. Qi, and S. Zhao, "A methodology of timing co-evolutionary path optimization for accident emergency rescue considering future environmental uncertainty," *IEEE Access*, vol. 7, pp. 131459–131472, 2019, doi: 10.1109/ACCESS.2019.2940315.

[4] H. Wen, Y. Lin, and J. Wu, "Co-evolutionary optimization algorithm based on the future traffic environment for emergency rescue path planning," *IEEE Access*, vol. 8, pp. 148125–148135, 2020, doi: 10.1109/ACCESS.2020.3014609.

[5] C. Xu, H. Wu, and S. Duan, "Constrained Gaussian condensation filter for cooperative target tracking," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1861–1874, Feb. 2002, doi: 10.1109/JIOT.2021.3088297.

[6] C. Xu, X. Wang, S. Duan, and J. Wan, "Spatial-temporal constrained particle filter for cooperative target tracking," *J. Netw. Comput. Appl.*, vol. 176, Feb. 2021, Art. no. 102913, doi: 10.1016/j.jnca.2020.102913.

[7] X. Niu, Y. Li, J. Kuang, and P. Zhang, "Data fusion of dual foot-mounted IMU for pedestrian navigation," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4577–4584, Jun. 2019, doi: 10.1109/JSEN.2019.2902422.

[8] A. F. García-Fernández, L. Svensson, and S. Särkkä, "Cooperative localization using posterior linearization belief propagation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 832–836, Jan. 2018, doi: 10.1109/TVT.2017.2734683.

[9] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 41–53, Jul. 2005, doi: 10.1109/MSP.2005.1458284.

[10] J. S. Russell, M. Ye, B. D. O. Anderson, H. Hmam, and P. Sarunic, "Cooperative localization of a gps-denied uav using direction-of-arrival measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 3, pp. 1966–1978, Jun. 2020, doi: 10.1109/TAES.2019.2942704.

[11] M. A. M. Marinho, A. Vinel, E. P. de Freitas, and S. M. A. Fernandez, "Cooperative localization for the Internet of Things," in *Proc. IEEE WONS*, Klosters, Switzerland, 2021, pp. 1–5, doi: 10.23919/WONS51326.2021.9415583.

[12] H. Oman, "Global positioning system, opportunities and problems," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 10, no. 7, p. 35, Jul. 1995, doi: 10.1109/62.400979.

[13] S. Zhang, X. Pan, and H. Mu, "A multi-pedestrian cooperative navigation and positioning method based on UWB technology," in *Proc. IEEE ICAIIS*, Dalian, China, 2020, pp. 260–264, doi: 10.1109/ICAIIS49377.2020.9194888.

[14] X. Wang, C. Xu, S. Duan, and J. Wan, "Error-ellipse-resampling-based particle filtering algorithm for target tracking," *IEEE Sensors J.*, vol. 20, no. 10, pp. 5389–5397, May 2020, doi: 10.1109/JSEN.2020.2968371.

[15] C. Xu, J. He, Y. Li, X. Zhang, X. Zhou, and S. Duan, "Optimal estimation and fundamental limits for target localization using IMU/TOA fusion method," *IEEE Access*, vol. 7, pp. 28124–28136, 2019, doi: 10.1109/ACCESS.2019.2902127.

[16] X. Wang, Z. Wang, and B. O'Dea, "A TOA-based location algorithm reducing the errors due to non-line-of-sight (NLOS) propagation," in *Proc. IEEE Veh. Technol. Conf.*, vol. 1. Atlantic City, NJ, USA, 2001, pp. 97–100, doi: 10.1109/VTC.2001.956563.

[17] Z. Wanlong, M. Weixiao, and H. Shuai, "A survey of multi-source information fusion navigation," *J. Telemetry Tracking Command*, vol. 37, no. 6, pp. 54–60, 2016.

[18] J. Wan, C. Xu, Y. Qiao, and X. Zhang, "Error constraint enhanced particle filter using quantum particle swarm optimization," *IEEE Sensors J.*, vol. 21, no. 21, pp. 24431–24439, Nov. 2021, doi: 10.1109/JSEN.2021.3113364.

[19] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "InnoHAR: A deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902, 2019, doi: 10.1109/ACCESS.2018.2890675.

[20] M. Basso, M. Galanti, G. Innocenti, and D. Miceli, "Triggered INS/GNSS data fusion algorithms for enhanced pedestrian navigation system," *IEEE Sensors J.*, vol. 20, no. 13, pp. 7447–7459, Jul. 2020, doi: 10.1109/JSEN.2020.2979335.

[21] X. Zhou, C. Xu, J. He, and J. Wan, "A cross-region wireless-synchronization—Based TDOA method for indoor positioning applications," in *Proc. IEEE WOCC*, Beijing, China, 2019, pp. 1–4, doi: 10.1109/WOCC.2019.8770637.

[22] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001, doi: 10.1109/18.910572.

[23] M. Wang, X. Pan, L. An, Z. Chen, Z. Tu, and C. Chu, "An optimal cooperative navigation algorithm based on factor graph for pedestrians," in *Proc. IEEE ICMSP*, Xi'an, China, 2021, pp. 51–56, doi: 10.1109/ICMSP53480.2021.9513365.

[24] H. Oliveira, S. S. Dias, and M. G. da Silva Bruno, "Cooperative terrain navigation using hybrid GMM/SMC message passing on factor graphs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3958–3970, Oct. 2020, doi: 10.1109/TAES.2020.2985316.

[25] H. Lan, J. Ma, Z. Wang, Q. Pan, and X. Xu, "A message passing approach for multiple maneuvering target tracking," *Signal Process.*, vol. 174, Sep. 2020, Art. no. 107621, doi: 10.1016/j.sigpro.2020.107621.

[26] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins, "Factored pose estimation of articulated objects using efficient nonparametric belief propagation," in *Proc. ICRA*, Montreal, QC, Canada, 2019, pp. 7221–7227, doi: 10.1109/ICRA.2019.8793973.

[27] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[28] Y. Ren, Y. Shen, Z. Zhang, X. You, and C. Zhang, "Efficient belief propagation polar decoder with loop simplification based factor graphs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5657–5660, May 2020, doi: 10.1109/TVT.2020.2979334.

[29] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004, doi: 10.1109/MSP.2004.1267047.

[30] H. Chen, W. Xian-Bo, J. Liu, J. Wang, and W. Ye, "Collaborative multiple uavs navigation with GPS/INS/UWB jammers using sigma point belief propagation," *IEEE Access*, vol. 8, pp. 193695–193707, 2020, doi: 10.1109/ACCESS.2020.3031605.

[31] F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, and X. Kong, "Random walks: A review of algorithms and applications," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 2, pp. 95–107, Apr. 2020, doi: 10.1109/TETCI.2019.2952908.

[32] F. Gustafsson *et al.*, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002, doi: 10.1109/78.978396.

[33] W. Li, Z. Yang, and H. Hu, "Sequential particle-based sum-product algorithm for distributed inference in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 1, pp. 341–348, Jan. 2013, doi: 10.1109/TVT.2012.2221484.

Cheng Xu (Member, IEEE) received the B.E., M.S., and Ph.D. degrees from the University of Science and Technology Beijing (USTB), Beijing, China, in 2012, 2015, and 2019, respectively.

He is currently an Associate Professor with the School of Computer and Communication Engineering, USTB. He is supported by the Postdoctoral Innovative Talent Support Program from Chinese Government in 2019. His current research interests include swarm intelligence, multiagent reinforcement learning, multimodal navigation, and Internet of Things.

Dr. Xu is an Associate Editor of *International Journal of Wireless Information Networks*.

Yuchen Shi is currently pursuing the master's degree with the University of Science and Technology Beijing, Beijing, China.

His research interests include wireless indoor positioning, pattern recognition, and Internet of Things.

Jiawang Wan is currently pursuing the Doctoral degree with the University of Science and Technology Beijing, Beijing, China.

His research interests include quantum optimization, multimodal navigation, and Internet of Things.

Shihong Duan (Member, IEEE) received the Ph.D. degree in computer science from the University of Science and Technology Beijing (USTB), Beijing, China, in 2012.

She is an Associate Professor with the School of Computer and Communication Engineering, USTB. Her research interests include multiagent reinforcement learning, multimodal navigation, and Internet of Things.