

Spatial-temporal constrained particle filter for cooperative target tracking

Cheng Xu ^{a,b,c,*}, Xinxin Wang ^{a,b}, Shihong Duan ^{a,b}, Jiawang Wan ^{a,b}

^a School of Computer and Communication Engineering, University of Science and Technology Beijing, China

^b Shunde Graduate School, University of Science and Technology Beijing, China

^c Beijing Key Laboratory of Knowledge Engineering for Materials Science, China



ARTICLE INFO

Keywords:

Cooperative target tracking

Accumulative error

Particle filter

Error ellipse resampling

Spatial temporal constraints

ABSTRACT

Localization is one of the most important topics of cyber physical system. In the last decades, much attention has been paid to the precise localization and performance evaluation in wireless sensor networks. Inertial-measurement-unit and Time-of-Arrival fusion is a state-of-the-art method to solve the accumulative error and drifting problem faced by sole inertial-measurement-unit positioning and navigation. Network cooperative technology could effectively suppress the accumulative error. This paper presents a spatial-temporal constrained particle filter algorithm for cooperative target tracking, so as to solve the problem of multi-target high-precision position tracking in complex and highly dynamic environments. Firstly, we propose an error-ellipse-resampling particle filter method. In the resampling stage of the particle filter, error ellipses with different confidence probabilities would be established with the use of the known estimated center and confidence scale, to achieve hierarchical resampling optimization based on the geometrical position of particles. As for cooperative tracking, an optimization method of state estimation based on spatial distance constraint is proposed, so that the Bayesian filter can benefit from spatial information and achieve cooperative tracking of spatial-temporal information fusion. Numerical experimental results show that the proposed error-ellipse-resampling particle filter could decrease the growth rate of cumulative errors and reach a positioning accuracy of 1.05 m, multi-target cooperative error-ellipse-resampling particle filter can effectively eliminate the cumulative error and achieve a positioning accuracy of 0.24 m.

1. Introduction

Target tracking has attracted extensive attention in applications such as intelligent traffic management systems and autonomous driving (Yuan et al., 2017; Wang et al., 2020a; Chen et al., 2019a). The Global Positioning System (GPS) is generally suitable for substantial outdoor positioning applications. However, in urban areas with dense buildings, GPS positioning cannot take advantage of its high accuracy due to the effects of signal blocking and multipath. Conventional wireless positioning technologies, such as Received Signal Strength (RSS) (Duan et al., 2020), Time of Arrival (TOA) (Xu et al., 2017), Time Difference of Arrival (TDOA) (Nicholas et al., 2018), and Angle of Arrival (AOA) (Zheng et al., 2018), could provide real-time accurate position estimation in areas that cannot be covered by GPS signals. However, in practical applications, these methods require pre-deployment of anchors, which makes them unsuitable for positioning estimation in unknown areas (Bianchi et al., 2019). Currently, Inertial Navigation System (INS) that utilizes an Inertial Measurement Unit (IMU), could provide a

low-cost solution for position estimation in unknown areas. The advantage of INS is that its position estimation is self-contained without additional infrastructure (Huang et al., 2010). However, it still faces the shortcoming of accumulative errors. In other words, it has difficulties to provide accurate position estimation for long-time applications (Ahmed and Tahir, 2017).

Filtering methods provide a practical and reliable positioning solution for improving the positioning accuracy of IMU (Vagle et al., 2018). Jonasson et al. (2020) developed two Extended Kalman Filter (EKF) concepts to estimate the vehicle position during a safe stop. In (Enayati et al., 2015), an improved Unscented Kalman Filter (UKF) is designed for robust estimation of position. Liu et al. (2020) proposed an improved Particle Filter (PF) algorithm for reducing the long-term accumulative error inherent in inertial positioning. However, they could only suppress the growth rate of the cumulative errors to a certain extent, but not eliminate it completely. So far, there are still challenges in improving the accuracy of positioning algorithms with filtering techniques. In practical applications, system models and measurement models are usually

* Corresponding author. School of Computer and Communication Engineering, University of Science and Technology Beijing, China.

E-mail address: xucheng@ustb.edu.cn (C. Xu).

non-linear. EKF estimates the mean and covariance of the state by linearizing the state equation, but it is accompanied by a tedious calculation process of the Jacobian matrix (Ghobadi et al., 2017). Since errors are introduced by linearization, UKF is still not suitable for higher-order nonlinear system models (Lee and McBride, 2019).

Compared with other filtering methods, although PF based on Monte Carlo sampling (Wang et al., 2020b) would pay more computational cost, it has better adaptability to nonlinear non-Gaussian systems (Fu and Jia, 2010; Veeramalla and Talari, 2020). However, PF still faces the problem of sample degeneracy and impoverishment (Ghobadi et al., 2017; Fu and Jia, 2010). Existing studies have proposed solutions to improve resampling, such as systematic resampling (Ala-Luhtala et al., 2016), stratified resampling (Li et al., 2015), etc. Both resamplings are based on a layered idea. After a series of iterations, the number of particles will be reduced, which would result in that most of the weights are occupied by a little part of them. Thus, the final state estimation results of general methods are not always as satisfactory as expected. To address this issue, in this paper, we proposed a resampling algorithm based on error ellipse estimation. The error ellipses are constructed based on different confidence probabilities, and the particle set is implemented to solve the problem of sample degeneracy and impoverishment.

In addition, the multi-source information fusion method is also a typical means to solve the accumulative errors of IMU (Chen et al., 2019b). IMU/GPS integrated navigation system could achieve centimeter-level positioning accuracy in vehicle navigation, but it cannot provide smooth positioning services in densely-built urban areas (Zhao, 2016). IMU/WiFi fusion provides a relatively stable positioning solution but requires pre-deployment of anchors and a large number of pre-measurements (Zou et al., 2017). With the implementation of IMU/UWB (Ultra-Wideband) fusion positioning technology, a reliable implementation for the positioning requirements under a long time and large distance span could be satisfied (Xu et al., 2018). UWB positioning method bases on TOA distance measurements, and it could provide decimeter-level positioning accuracy. Nevertheless, most IMU/TOA fusion positioning methods still need to deploy communication base stations.

In anchor-free scenarios, the mobile node only need integrate inertial sensor and ranging sensor to achieve multi-target cooperative positioning based on its own-collected inertial and distance measurements (Fan et al., 2019). The distance measurements between nodes could be used as spatial state constraints. With the use of Bayesian statistical inference, the posterior state mean under distance constraints would be obtained (Zachariah et al., 2012). Xu (Xu et al., 2019) and Zihajehzadeh (Zihajehzadeh and Park, 2016) respectively proposed cooperative positioning techniques based on IMU/TOA fusion, which utilized spatial information to optimize the tracking of multi-target positions. However, these static fusion optimization methods also require the deployment of external anchors. Wireless signals are easily blocked due to non-line-of-sight (NLOS) factors and finally result in positioning errors. To address this issue, we introduced a statically optimized posterior state mean to the filtering process in time series, and achieved spatial-temporal fusion in anchor-free cooperative target tracking.

Generally, in this paper, we propose a cooperative target tracking algorithm that combines spatial-temporal measurements. In the aspect of time series, an optimized historical position is used as the initial prediction center to establish error ellipses, and an improved layered resampling algorithm is put forward to solve the particle degradation and depletion problem faced by traditional particle filtering. Furthermore, we propose a constrained Bayesian optimization method, integrating spatial distance ranging to achieve position optimization under multi-target distance constraints. To further improve the positioning performance, the optimized position status would be applied to establish error ellipses, contributing to the cooperative algorithm with spatial-temporal fusion. The main contributions of this paper are summarized as follows:

- We propose a particle filter algorithm based on spatial-temporal constraints. On account of the IMU estimated center and confidence scales, error ellipses with different confidence probabilities are established, and hierarchical resampling based on particle geometric position is implemented. Through the particle division and resampling mechanism of different weight levels, the sample degeneracy and impoverishment problem is to some extent resolved.
- To achieve multi-target cooperative tracking, we propose an optimization method for filtering state estimation based on spatial distance constraints. Bayesian recursive filtering estimation in time series could benefit from the spatial measurements. Based on the optimized posterior state estimation, the error ellipse estimation center in time series is closer to the true position, thereby eliminating the cumulative error with the use of proposed cooperative tracking algorithm.

Table 1 summarizes the notations used later in this paper. The remainder of this paper is organized as follows: Section 2 presents the symbolic descriptions, related definitions, and model descriptions of the problems described. Section 3 focuses on the multi-target cooperative constrained particle filter algorithm, which describes the implementation of layered resampling and cooperative constrained optimization using error ellipse estimation. Section 4 derives the posterior Cramér Rao Lower Bound (PCRLB) of the cooperative tracking and details the complexity analysis. Numerical simulation and practical experiments are carried out in Section 5. Conclusions are drawn in Section 6.

Table 1
Notations.

Symbol	Description
X	Position state
\hat{X}	Estimate state
K	Joint position state
m_l, C_l	Position mean and covariance
$\widehat{m}_l, \widehat{C}_l$	Posterior position mean and covariance
$m_{z_1 c}, B_{z_1 c}$	Conditional position mean and covariance
(x_p, y_p)	IMU estimate center
W	Particle weight
P	Coordinate
V	Velocity
K	Time scale
t_s	Sampling interval
N_a	Number of target nodes
Z	Measurement
A	Acceleration
μ	Process noise
D	True distance
\hat{d}	Distance measurement
θ	True angle
$\hat{\theta}$	Angle measurement
$\epsilon_{1,k}$	Step noise of intra-node
$\epsilon_{2,k}$	Angle noise of intra-node
$\epsilon_{3,k}$	Distance noise of inter-node
$\gamma_{1,k}^2$	Noise variance of step
$\gamma_{2,k}^2$	Noise variance of angle
$\gamma_{3,k}^2$	Noise variance of distance
ϕ	Particle initialization noise
N	Number of particles
$\delta(\cdot)$	Dirac function
$E\{\cdot\}$	Expectation operator
α	Confidence probability of error ellipse
s	Confidence scale of the error ellipse
φ	Tilt angle of the error ellipse
λ	Feature vector
I	Identity matrix

2. Problem formulation

2.1. Dynamic model

Consider the cooperative network contains N_a target nodes. Each target is equipped with a distance measurement sensor and an inertial measurement sensor. It is represented by $\mathcal{N}_a = \{1, 2, 3, \dots, N_a\}$. The measurement process and state transition process are performed in discrete time t_k , $k = 0, 1, 2, \dots, K$. $X_{k,n} \in \mathbb{R}^4$ defined as the state information of node n at time k , which includes coordinate vector $P_{k,n} = [x_{k,n}, y_{k,n}]^T$ and velocity vector $V_{k,n} = [vx_{k,n}, vy_{k,n}]^T$. Then, the augmented state X_k could be expressed as:

$$X_k = [P_{k,1:N_a}, V_{k,1:N_a}]^T \quad (1)$$

For multi-target cooperative tracking problem, we firstly present the dynamic model under the cooperative condition. As shown in Fig. 1, it is assumed that the motion law of the mobile node conforms to the dynamic random walk process (Ahn et al., 2019). Thus, we use the first-order Markov model for dynamic modeling, and the state $X_{k-1,n}$ of the target node n at time $k-1$ would contribute to the state $X_{k,n}$ at time k , denoted as:

$$X_{k,n} = FX_{k-1,n} + GA_{k-1,n} + G\mu \quad (2)$$

where $F = \begin{bmatrix} 1 & 0 & t_s & 0 \\ 0 & 1 & 0 & t_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $G = \begin{bmatrix} t_s^2/2 & 0 \\ 0 & t_s^2/2 \\ t_s & 0 \\ 0 & t_s \end{bmatrix}$ are state transition

matrices. t_s denotes the sampling interval, and $\mu = [\mu_x, \mu_y]^T$ is Gaussian process noise with mean 0 and variance σ^2 , i.e., $\mu_x, \mu_y \sim N(0, \sigma^2)$. The acceleration at time $k-1$ is notated as $A_{k-1,n} = [a_{x_{k-1,n}}, a_{y_{k-1,n}}]^T$, and the components in both directions are randomly generated from $\{0, -g, g\}$ with probabilities modeled as a random Markov jump (Li et al., 2012). In actual physical measurements, step length and angle measurement could be obtained by integrating the acceleration and angular velocity. The step length measurement of the target node could be simplified as:

$$\hat{d}_{k,n} = d_{k,n} + \epsilon_{1,k}, \quad \epsilon_{1,k} \sim N(0, \gamma_{1,k}^2) \quad (3)$$

where $d_{k,n}$ represents the true distance of the target node n moving from time k to time $k+1$, namely

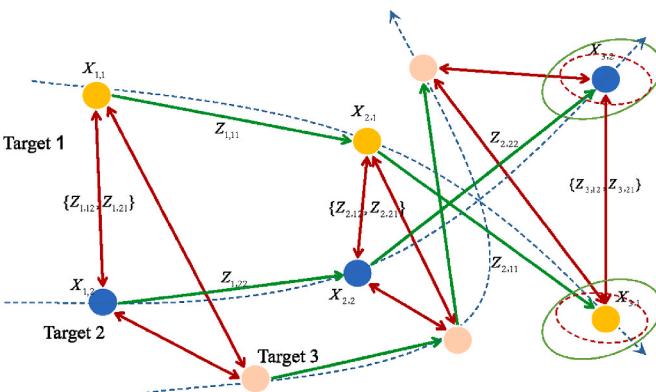


Fig. 1. Schematic diagram of multi-target cooperative tracking in the center of the city: a red one-way curve represents the inertial measurement of the intra-node, and a red two-way arrow denotes the range measurement of the inter-node. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

$$d_{k,n} = \sqrt{(x_{k+1,n} - x_{k,n})^2 + (y_{k+1,n} - y_{k,n})^2} \quad (4)$$

and $\epsilon_{1,k}$ is Gaussian step length noise with mean of 0 and variance of $\gamma_{1,k}^2$.

Then, the vector $\hat{d} = [\hat{d}_{0,n}, \hat{d}_{1,n}, \dots, \hat{d}_{K-1,n}]^T$ reflects the step measurement information. The angle measurement is denoted as:

$$\hat{\theta}_{k,n} = \theta_{k,n} + \epsilon_{2,k}, \quad \epsilon_{2,k} \sim N(0, \gamma_{2,k}^2) \quad (5)$$

where $\theta_{k,n}$ is the true horizontal angle, namely

$$\theta_{k,n} = \arctan \frac{y_{k+1,n} - y_{k,n}}{x_{k+1,n} - x_{k,n}} \quad (6)$$

and $\epsilon_{2,k}$ is Gaussian angle noise with mean of 0 and variance of $\gamma_{2,k}^2$.

Then, the vector $\hat{\theta} = [\hat{\theta}_{0,n}, \hat{\theta}_{1,n}, \dots, \hat{\theta}_{K-1,n}]^T$ reflects the angle measurement information. The external measurement between different nodes at time k is denoted as:

$$\hat{d}_{k,nj} = \sqrt{(x_{k,n} - x_{k,j})^2 + (y_{k,n} - y_{k,j})^2} + \epsilon_{3,k}, \quad \epsilon_{3,k} \sim N(0, \gamma_{3,k}^2) \quad (7)$$

where $\epsilon_{3,k}$ is Gaussian distance noise with mean of 0 and variance of $\gamma_{3,k}^2$.

In short, $Z_{k,nn} := [\hat{d}_{k,n}, \hat{\theta}_k]$ reflects the internal inertial measurement of the intra-node, and $Z_{k,nj} := [\hat{d}_{k,nj}]$ reflects the distance measurement of the inter-node.

2.2. Bayesian model of cooperative tracking

The position and measurements of the target node are represented as a set of random variables. According to the state evolution process of Markov model, the joint probability density function of the measurement and the augmented state from time t_1 to time t_K is defined as:

$$f(Z|X) = \prod_{k=1}^K f(X_k | X_{k-1}) \cdot f(Z_k | X_k) \quad (8)$$

where $X := X_{1:K}$, $Z := Z_{1:K}$. Assuming that the dynamics of different nodes are independent of each other, the state parameters of the dynamic model could be decomposed into:

$$f(X_k | X_{k-1}) = \prod_{n \in N_a} \left[f(X_{k,n} | X_{k-1,n}) \times \prod_{j \in N_a \setminus \{n\}} f(X_{k,nj} | X_{k-1,nj}) \right] \quad (9)$$

where $X_{k-1,nj} := X_{k-1,n} - X_{k-1,j}$. The measurements between different nodes are independent of each other, and they are also affected by the position states and various parameters. Thus, the probability density function of the measurement model could be denoted as:

$$f(Z_k | X_k) = \prod_{n \in N_a} \left[\begin{array}{c} \text{intra-node measurements} \\ f(Z_{k,nn} | X_{k,n}) \end{array} \times \prod_{j \in N_a \setminus \{n\}} \begin{array}{c} \text{inter-node measurements} \\ f(Z_{k,nj} | X_{k,nj}) \end{array} \right] \quad (10)$$

where $X_{k,nj} := X_{k,n} - X_{k,j}$. As shown in Fig. 2, we use the red connection to indicate the relationship of intra-node measurements, and the green connection denotes the dependence of the inter-node measurements. In scenarios with spatial cooperation, the Fisher Information Matrix (FIM) (Win et al., 2018) has an off-diagonal block that corresponds to the spatial cooperation measurements, and the graph has measurement nodes that connect different agents. In scenarios with spatial-temporal cooperation, the FIM has off-diagonal blocks corresponding to temporal measurements, and the graph has measurement nodes that connect agents at consecutive time instants.

Assume that the state transition process follows the first-order Markov model, that is, $p(X_k | X_{1:k-1}) = p(X_k | X_{k-1})$. The prior probability density function $p(X_k | X_{1:k-1})$ of the dynamic model could be calculated

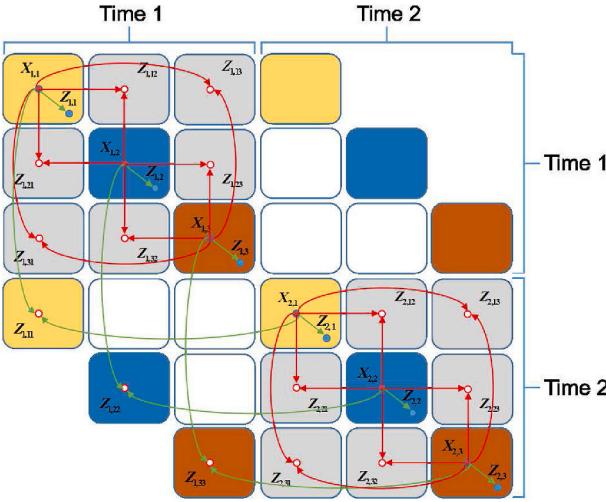


Fig. 2. Bayesian graph model of multi-target cooperative: the spatial-temporal cooperative is shown by two nodes.

from the prior probability distribution $p(X_{k-1}|X_{1:k-1})$ at the previous moment, namely

$$p(X_k|Z_{1:k-1}) = \int p(\hat{X}_k|X_{k-1}) p(X_{k-1}|\hat{Z}_{1:k-1}) dX_{k-1} \quad (11)$$

among which $p(X_{k-1}|Z_{1:k-1})$ is a known quantity, and $p(X_k|Z_{1:k-1})$ is determined by the above dynamic model. After obtaining the independent measurement Z_k at time t , the posterior probability density function can be updated to:

$$p(X_k|Z_{1:k}) = \frac{p(\hat{Z}_k|X_k) p(X_k|\hat{Z}_{1:k-1})}{p(\hat{Z}_k|Z_{1:k-1})} \quad (12)$$

where $p(X_k|L_k)$ is a likelihood function defined by the measurement model and $p(Z_k|Z_{1:k-1})$ is a normalized constant. In summary, with obtaining the prior estimate $p(X_k|Z_{1:k-1})$ at the current time and the conditional probability $p(Z_k|X_k)$, the posterior probability $p(X_k|Z_{1:k})$ at the current time could be derived from Bayesian recursion theory.

3. Cooperative constrained PF algorithm

In this section, we firstly proposed the error-ellipse-based resampling algorithm for filtering estimation of target nodes. On this basis, the multi-target cooperative particle filter algorithm based on distance measurement is described to realize the spatial-temporal fusion for cooperative tracking.

3.1. Error-ellipse-based resampling

In this subsection, we first briefly reviewed the flow of particle filtering, then described the error elliptic constraint method, and finally introduced the particle weight redistribution based on the error resampling algorithm through the screening and replication of particles, so as to reduce the particle dilution problem.

3.1.1. Review of PF

PF provides a method for recursively generating the posterior probability density function, that is, using a set of particles to represent the posterior probability distribution of the state, which also meets the idea of Monte Carlo sampling (Duan et al., 2020). The standard process of PF contains initialization, prediction, and update. The sampled particles are generated at the initial stage. Then, the prediction and

updating process would be executed cyclically at time $\{t_k, 1 \leq k \leq K\}$, and finally put forward the state estimation results of the target node.

Initialization: Generally, PF completes the estimation of state X_k by updating a set of random measurements $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^N$ which consists of N particles $X_k^{(i)}$ and the corresponding weight $w_k^{(i)}$ at time t_k . When the number of particles is sufficient, the state estimation approximates the posterior probability density $p(X_{1:k}|Z_{1:k})$ of the unknown distribution $X_{1:k}$. In the initialization phase, it is necessary to generate a series of particles on account of prior knowledge, that is, $X_0^{(1:N)} \sim p(X_0)$. Considering the motion state of a single target in a 2-D coordinate system, the particles at the initial moment could be generated with random noise, i.e.,

$$\begin{bmatrix} \tilde{x}_0 \\ \tilde{y}_0 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \phi \quad (13)$$

where $[x_0, y_0]^T$ is the initial position of the target to be tracked, and $\phi = [\phi_x, \phi_y]^T$ is Gaussian noise in the horizontal and vertical axes, that is, $\phi_x, \phi_y \sim N(0, \gamma_0^2)$.

Prediction: Based on the Bayesian recursive estimation criterion, to estimate the posterior probability of the current state, the prior probability of the current moment needs to be calculated in advance, based on the posterior probability of the previous moment. Thus, Monte Carlo method is taken into consideration. A set of random weighted particles (samples) are collected from the state space of the known distribution to replace the posterior probability, i.e., $\{X_k^{(i)}\} \sim q(X_k^{(i)}|X_{k-1}^{(i)}, Z_k)$.

Update: In this stage, the particles are resampled and the weighted average of all particles is chosen as the output of overall state estimation. The initial weight of the particle is assigned as $1/N$, and it is updated by the following equation:

$$w_k^{(i)} \sim w_{k-1}^{(i)} \frac{p(Z_k|X_k^{(i)}) p(X_k^{(i)}|X_{k-1}^{(i)})}{q(X_k^{(i)}|X_{k-1}^{(i)}, Z_k)} \quad (14)$$

where $q(\cdot)$ denotes the importance probability density, and it is generally represented as the transition prior probability density function, that is, $q(X_k^{(i)}|X_{k-1}^{(i)}, Z_k) = p(X_k^{(i)}|X_{k-1}^{(i)})$ (Xiao and Pan, 2020). Furthermore, the weights of these particles are normalized, $\tilde{w}_k^{(i)} = w_k^{(i)} / \sum_{i=1}^N w_k^{(i)}$. Finally, the approximated posterior probability density function can be expressed as:

$$p(X_k|Z_{1:k}) \approx \sum_{i=1}^N \tilde{w}_k^{(i)} \delta(X_k - X_k^{(i)}) \quad (15)$$

where $\tilde{w}_k^{(i)}$ and $X_k^{(i)}$ denotes the weight and state information of particle i at time t_k , and $\delta(\cdot)$ is Dirac function. When N approaches infinity, the approximation will gradually converge to the true posterior density.

3.1.2. Error ellipse constraint

Based on the above considerations, we introduce the rules of error ellipse constraints to the implementation of resampling. When performing interval estimation on the scale s (that is, the value range of the estimation scale), if a small probability β is given in advance, an interval (s_1, s_2) can be found to satisfy:

$$\Pr(s_1 < s < s_2) = 1 - \beta \quad (16)$$

The interval (s_1, s_2) is the confidence interval of the scale s , where s_1 and s_2 are called confidence limits (or critical values). The probability β represents the significance level (or risk), and $1 - \beta$ denotes the confidence level (or confidence). We use the symbol α to express the confidence probability, i.e., $\alpha = 1 - \beta$. In 2-D/3-D parameter estimation (such as 2-D/3-D target tracking), this confidence interval could be formalized

as an error ellipse/ellipsoid.

In this paper, we consider the 2-D covariance matrix of N particles at time t_k :

$$C = \begin{bmatrix} \text{cov}(x, x) & \text{cov}(x, y) \\ \text{cov}(y, x) & \text{cov}(y, y) \end{bmatrix} \quad (17)$$

where $\text{cov}(x, x)$ and $\text{cov}(y, y)$ are the variance in the x-axis and y-axis directions, respectively. The covariance could be obtained:

$$\text{cov}(x, y) = E[(x - E(x))(y - E(y))] \quad (18)$$

If x is positively related to y , then y and x are also positively correlated, i.e., $\text{cov}(x, y) = \text{cov}(y, x)$. Therefore, the covariance matrix is always a symmetric matrix, whose values on the diagonal are variances, and those on the off-diagonal are covariances. Then, equation of the ellipse whose center is not at the origin point could be expressed as:

$$\frac{(x - x_p)^2}{\lambda_1} + \frac{(y - y_p)^2}{\lambda_2} = s \quad (19)$$

where λ_1 and λ_2 are the maximum and minimum eigenvalues corresponding to the covariance matrix, respectively. (x_p, y_p) is the estimated center position, and s is the scale of error ellipses. When the error ellipse is tilted relative to the coordinate system, the tilt angle φ of the x-axis and y-axis could be obtained from the following equation:

$$\varphi = \arctan\left(\frac{\lambda_1(y)}{\lambda_1(x)}\right) \quad (20)$$

Furthermore, we obtain the error ellipse constraint represented by the rotated coordinate (x', y') and the angle φ as:

$$\frac{((x' - x_p)\cos\varphi + (y' - y_p)\sin\varphi)^2}{\lambda_1} + \frac{(-(x' - x_p)\sin\varphi + (y' - y_p)\cos\varphi)^2}{\lambda_2} \leq s \quad (21)$$

3.1.3. Resampling method

Fig. 3 depicts the resampling process for particle sets based on error ellipse constraints. Two different confidence levels (confidence probabilities) are established. During the resampling process, N potential particles are divided into three levels by geometric positions. The particles outside the outer ellipse are N_l negligible particles which would be

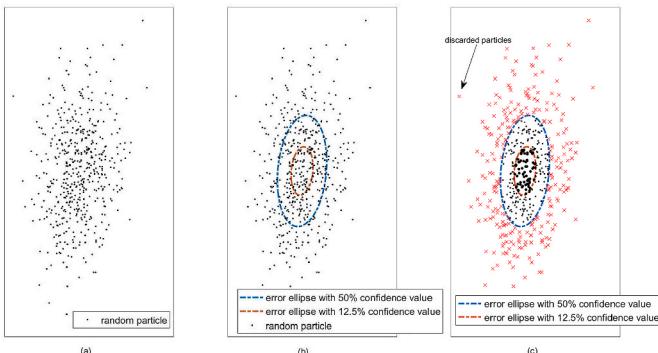


Fig. 3. An example of a typical resampled particle screening process: a) shows the initial distribution of the particles. b) two different confidence level ellipse (50% and 12.5% confidence probability respectively) based on the covariance matrix of the particle set. c) Screening results for particles of different levels. The red particles could be removed and the bold points are copied, to emphasize the high weight particles and reduce the low weight ones. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

discarded in later stages; the particles located in the middle of these two ellipses are regarded as the moderate particles, which would be reserved; N_h dominating particles inside the inner ellipse would be duplicated. During the duplication process, since the rounding operation may lead to the duplication times of particles to be less than 1, the first $N_t = (N_l - [N_l / N_h] N_h)$ dominating particles are duplicated $c_1 = [N_l / N_h] + 2$ times, then the remaining $N_h - N_t$ dominating particles are duplicated $c_2 = [N_l / N_h] + 1$ times. After all particles have performed the duplication operation, the quantity of particles N still holds. The selected set of particles would also be used as initial input for the next iteration.

During the resampling process, due to the discarding and duplication of some particles, the weights of the discarded are reassigned to those being duplicated. The weights are redistributed using the following equation:

$$\hat{w}_k^{(i)} = \begin{cases} (1 - M)/(N_l + N_h), & s_k^{(i)} < s_1 \\ \tilde{w}_k^{(i)}, & s_1 \leq s_k^{(i)} \leq s_2 \\ 0, & s_k^{(i)} > s_2 \end{cases} \quad (22)$$

where M is the sum of the weights of moderate particles, N_l and N_h are respectively the number of negligible particles and dominating particles. $s_k^{(i)}$ represents the distance between the particle and the prediction center at time k . The state estimation result is updated by weighted average, namely

$$\begin{cases} x_k = \sum_{i=1}^N \hat{w}_k^{(i)} x_k^{(i)} \\ y_k = \sum_{i=1}^N \hat{w}_k^{(i)} y_k^{(i)} \end{cases} \quad (23)$$

The error ellipse resampling algorithm is described in [Algorithm 1](#).

Algorithm 1

Error-Ellipse-Resampling Algorithm (EER)

Require: $\left[\{X_k^i, w_k^i\}_{i=1}^N, (x_p, y_p)\right] \leftarrow$ status, weight information
Ensure: $\left[\{\hat{X}_k^i, \hat{w}_k^i\}_{i=1}^N, \hat{X}\right] \leftarrow$ status, weight, estimated value

```

for i ← 1, N do
     $s_k^i \leftarrow \text{ellipse}(X_k^i, (x_p, y_p))$  ▷ Equation(21);
    if  $s_k^i > s_1$  then
        index( $N_l$ ) = i;
         $N_l = N_l + 1$ ;
    else if  $s_k^i < s_2$  then
        index( $N_h$ ) = i;
         $N_h = N_h + 1$ ;
    else
         $M = M + \tilde{w}_k^i$ ;
         $\hat{X}_k^{\text{index}(n)} = X_k^i$ ,  $\hat{w}_k^{\text{index}(n)} = w_k^i$ ;
         $n = n + 1$ ;
         $\hat{X} = \hat{X} + w_k^i * X_k^i$ ;
    end if
end for
 $N_t = N_l - [N_l / N_h] * N_h$ ;
weight =  $(1 - M)/(N_l + N_h)$ ;
for j ← 1,  $N_h$  do
    while  $j \leq N_t$  do
         $c^{(j)} = [N_l / N_h] + 2$ ;
        for do l←1,  $c^{(j)}$ 
             $\hat{X}_k^{\text{index}(n)} = X_k^i$ ,  $\hat{w}_k^{\text{index}(n)} = weight$ ;
             $n = n + 1$ ;
             $\hat{X} = \hat{X} + weight * X_k^i$ ;
        end for
    end while
     $c^{(j)} = [N_l / N_h] + 1$ ;

```

(continued on next page)

Algorithm 1 (continued)

```

for do  $l \leftarrow 1, c^{(j)}$ 
     $\hat{X}_k^{\text{index}(n)} = X_k^{\text{index}(j)}, \hat{W}_k^{\text{index}(n)} = \text{weight};$ 
     $n = n + 1;$ 
     $\hat{X} = \hat{X} + \text{weight} * X_k^{\text{index}(j)};$ 
end for
end for

```

3.2. Distance constrained optimization

In this section, we fuse spatial distance measurements between mobile nodes for cooperative estimation. According to the Bayesian recursive criterion, when the previous estimates are biased, the state estimates at the subsequent moments will be affected. In the spatial cooperation, Bayesian optimization under the distance constraint is used to optimize the position state of the target node. As shown in Fig. 4, we focus on the 2-D coordinate state information. The PF will output the initial mean value and covariance of position state. This information will be used as prior knowledge of Bayesian optimization estimation. Under the distance constraint of given node position, the posterior is calculated by convex combination approximation of the integral conditional mean and covariance to complete the optimization of spatial cooperative constraints.

3.2.1. Bayesian optimization under distance constraints

During dynamic estimation in the time domain, the particle filter algorithm will continuously yield the mean estimation of state variables. The state estimation value $\{X_{k,1}, X_{k,2} \dots X_{k,N_a}\}$ of N_a nodes at each moment could be used as prior knowledge for optimization. Ideally, the target node n could receive the measurements $Z_k = \{Z_{k,nj} : j \in N_a / \{n\}\}$ from other $N_a - 1$ neighbor nodes, and node n will select the value within the distance constraint range from these measurements for the next optimization, that is, satisfying the distance constraint $c : \|pX\| \leq Z_{k,nj}$, where $p = [I_2, -I_2, 0_{2 \times 4}]$.

Furthermore, we define the joint state of node n and node j at time k as $\kappa_k = [P_{k,n}, P_{k,j}]^T$. Since the estimated value in time series is calculated by the weighted average of particles, then the mean value m_k =

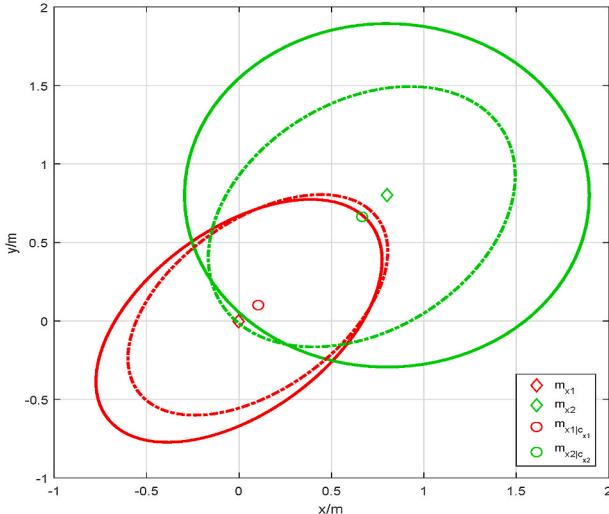


Fig. 4. Schematic diagram of distance constraint optimization. The target node's location information is indicated by the mean m_l and the covariance C_l . The confidence probability corresponding to the ellipse is set as 50%. The position and error ellipse before optimization are expressed as diamond points and solid ellipses. Correspondingly, the optimized position is expressed as a circular dot and a dotted ellipse. Using the error ellipse to constrain the position would get an estimation more closer to the true position.

$[m_{k,n}, m_{k,j}]^T$ and covariance $C_k = [C_{k,n}, C_{k,j}]^T$ of the joint state at time k will also be used as prior knowledge of the optimization process. So far, as long as the posterior probability density under the given distance constraint c is calculated, the posterior mean and covariance of the target state could be obtained.

In order to simplify the calculation, we firstly introduce a new state vector $z = Tk \in \mathbb{R}^4$, where z is a Gaussian distribution with a mean of m_z and a variance of C_z , i.e., $z \sim N(m_z, C_z)$, and κ indicates the joint state vector, namely $\kappa = [P_1, P_2]^T$. The state vector z is given by a reversible linear transformation and consists of two sub-vectors $z_1 = P_1 - P_2$ and $z_2 = P_1 + P_2$. Here, the simplified symbols P_1 and P_2 represent the position states of the two mobile nodes before optimization, respectively. We could verify that $T = \begin{bmatrix} I_2 & -I_2 \\ I_2 & I_2 \end{bmatrix}$. Further reasoning leads to $m_z = Tm_\kappa$ and $C_z = TC_\kappa T^T$. Then, we can obtain new constraints $c : \|z_1\| \leq Z$, where the distance constraint is only related to z_1 .

The calculation of posterior $\hat{m}_{\kappa|c}$ and $\hat{C}_{\kappa|c}$ requires two steps. Firstly, perform an affine transformation to map the infinite integration area to a finite area, so as to reduce the integration dimension. Therefore, the transformation process will simplify the calculation of conditional mean and covariance. Secondly, the approximate calculation of integrals performs deterministic sampling method.

3.2.2. Affine transformation

As to the analysis above, we need to calculate the posterior mean $\hat{m}_{\kappa|c}$ and covariance $\hat{C}_{\kappa|c}$. Moreover, we conclude that $\hat{m}_{\kappa|c} = T^{-1}m_{z|c}$ and $\hat{C}_{\kappa|c} = T^{-1}C_{z|c}T^{-T}$. Thus, we could turn the problem into solving $m_{z|c}$ and $C_{z|c}$ with transforming by affine transformation:

$$\begin{aligned} m_{z_2|z_1} &= u_{z_2} + Az_1 \\ C_{z_2|z_1} &= C_{z_2} - AC_{z_1}^T \end{aligned} \quad (24)$$

we could obtain the mean value $m_{z_2|z_1}$ and covariance $C_{z_2|z_1}$ of the conditional probability density function $p(z_2|z_1)$. The intermediate variables in the above equation are expressed as $A = C_{z_2 z_1} C_{z_1}^{-1}$ and $u_{z_2} = m_{z_2} - Am_{z_1}$, respectively. Let $m_{z|c} = [m_{z_1|c}, m_{z_2|c}]^T$, then the conditional mean $m_{z_1|c}$ is denoted as:

$$m_{z_1|c} = \int_{z_1} z_1 p(z_1|c) dz_1 \quad (25)$$

Since z_1 is given the norm $\|z_1\|$ provides no additional information, then the pdf $p(z_2|z_1, \|z_1\|)$ is identical to $p(z_2|z_1)$. The same goes that given z_1 a bound on the norm $\|z_1\| \leq Z$ yields no additional information. Hence $p(z_2|z_1, c) = p(z_2|z_1)$ for all valid z_1 , i.e., $\forall z_1 \in \{z_1 \in \mathbb{R}^2 : \|z_1\| \leq Z\}$. The conditional mean $m_{z_2|c}$ is denoted as:

$$\begin{aligned} m_{z_2|c} &= \int_{z_1} \int_{z_2} z_2 p(z_2|z_1) dz_2 p(z_1|c) dz_1 \\ &= \int_{z_1} (u_{z_2} + Az_1) p(z_1|c) dz_1 \\ &= u_{z_2} + Am_{z_1|c} \end{aligned} \quad (26)$$

and the conditional covariance matrix is denoted as:

$$C_{z|c} = B_{z|c} - m_{z|c} m_{z|c}^T \quad (27)$$

where the covariance matrix $B_{z|c}$ is denoted as $\begin{bmatrix} B_{z_1|c} & B_{z_1 z_2|c} \\ B_{z_1 z_2|c} & B_{z_2|c} \end{bmatrix}$. The covariance of the posterior probability $p(z_1|c)$ could be denoted as:

$$B_{z_1|c} = \int_{z_1} z_1 z_1^T p(z_1|c) dz_1 \quad (28)$$

Therefore, the covariance of $p(z_2 z_1 | c)$ and $p(z_2 | c)$ could be obtained according to the affine transformation:

$$\begin{aligned} B_{z_1 z_2 | c} &= \int_{z_1} \left[\int_{z_2} z_2^T p(z_2 | z_1) dz_2 \right] p(z_1 | c) dz_1 \\ &= m_{z_1 | c} u_{z_2}^T + P_{z_1 | c} A^T \end{aligned} \quad (29)$$

$$\begin{aligned} B_{z_2 | c} &= \int_{z_1} \left[\int_{z_2} z_2 z_2^T p(z_2 | z_1) dz_2 \right] p(z_1 | c) dz_1 \\ &= \int_{z_1} \left(C_{z_2 | z_1} + m_{z_2 | z_1} m_{z_2 | z_1}^T \right) p(z_1 | c) dz_1 \\ &= C_{z_2} - A C_{z_2 | z_1}^T + u_{z_2} u_{z_2}^T + u_{z_2} m_{z_1 | c}^T A^T + A m_{z_1 | c} u_{z_2}^T + A R_{z_1 | c} A^T \end{aligned} \quad (30)$$

Therefore, we only need to calculate the integral (25) and (28) corresponding to $p(z_1 | c)$, and we can find the solutions of the posterior mean $\hat{m}_{z|c}$. Thus, covariance $\hat{C}_{z|c}$ through affine transformation without directly solving the posterior probability $p(X|c)$.

3.2.3. Integral approximation

In order to avoid calculating complex numerical integration, we use a convex combination to approximate the conditional mean and covariance:

$$\begin{aligned} \hat{m}_{z_1 | c} &\simeq \sum_{i=0}^{2\Lambda} w^{(i)} z_1^{(i)} \\ \hat{B}_{z_1 | c} &\simeq \sum_{i=0}^{2\Lambda} w^{(i)} z_1^{(i)} (z_1^{(i)})^T \end{aligned} \quad (31)$$

where $z_1^{(i)}$ and $w^{(i)}$ denotes sampling points and corresponding weights, and n is the dimension of the state variable (Candy, 2016). When the probability mass α of the sampling point $z_1^{(i)}$ is within the constraint range, the approximated value holds. Otherwise, resample the points to ensure that the approximated average value falls within the convex boundary, thereby reducing the dispersion. Therefore, parameter α determines the effectiveness of sampling points. Select $2n + 1$ points by the following equation:

$$h^{(i)} = \begin{cases} m_{z_1}, & i = 0 \\ m_{z_1} + s_\alpha^{1/2} [C_{z_1}^{1/2}]_i, & i = 1, \dots, \Lambda \\ m_{z_1} - s_\alpha^{1/2} [C_{z_1}^{1/2}]_i, & i = \Lambda + 1, \dots, 2\Lambda \end{cases} \quad (32)$$

where Λ indicates the dimension of position variable, s_α is the confidence scale that satisfies $\Pr(s \leq s_\alpha) = \alpha$ and $s = (z_1 - m_{z_1})^T C_{z_1}^{-1} (z_1 - m_{z_1})$. This is equivalent to the confidence scale η_α of the confidence ellipse, which is directly affected by the confidence level.

The deterministic sampling method will select $2\Lambda + 1$ sampling points, and the sampling points that violate the constraint condition c will be orthogonally projected onto the constraint boundary. Then, we use the following equation for screening:

$$z_1^{(i)} = \begin{cases} h^{(i)}, & \text{if } \|h^{(i)}\| \leq Z \\ \frac{Z}{\|h^{(i)}\|} h^{(i)}, & \text{other} \end{cases} \quad (33)$$

where $i = 0, \dots, 2\Lambda$. The weights of sampling points will be regenerated based on the following criterion:

$$w^{(i)} = \begin{cases} 1 - \frac{2}{s_\alpha}, & i = 0 \\ \frac{1}{\Lambda s_\alpha}, & i = 1, \dots, 2\Lambda \end{cases} \quad (34)$$

The estimated values of the posterior mean $\hat{m}_{z_1 | c}$ and covariance $\hat{B}_{z_1 | c}$ could be obtained by weighting the resampled points $\{z_1^{(i)}\}_{i=0}^{2\Lambda}$. The effectiveness of resampling under distance constraint is displayed in Fig. 5. The description of spatial cooperative constrained optimization algorithm is shown in Algorithm 2.

Algorithm 2

Spatial Cooperative Constrained Optimization Algorithm (SCCO)

Require: $\{m_1 = [p_1, p_2]^T, C_1 = [C_1, C_2]^T, Z_{12}\} \leftarrow$ Joint state mean and covariance, distance measurement

Ensure: $\{\hat{m}_1, \hat{B}_1, \hat{m}_2, \hat{B}_2\} \leftarrow$ Posterior state mean and covariance

if $\|m_1 - m_2\| \geq Z_{12}$ **then**

1. Calculate the converted covariance C_z through C_l
2. Calculate the conditional covariance C_{z_1} and C_{z_2} in equation (27).
3. Calculate the intermediate variable A and u_{z_2} of the affine transformation

for $i \leftarrow 0, 4$ **do**

$h^{(i)} \leftarrow \text{sample}(\Lambda = 2)$ \triangleright Equation (32)

if $h^{(i)} > Z_{k,nj}$ **then**

$z_1^{(i)} \leftarrow \text{norm}(h^{(i)});$

else

$z_1^{(i)} = h^{(i)};$

end if

if $i = 0$ **then**

$w^{(i)} = 1 - 2/s_\alpha;$

else

$w^{(i)} = 1/\Lambda s_\alpha;$

end if

end for

for $i \leftarrow 0, 4$ **do**

$\hat{m}_{z_1 | c} = \hat{m}_{z_1 | c} + w^{(i)} z_1^{(i)};$

$\hat{B}_{z_1 | c} = \hat{B}_{z_1 | c} + w^{(i)} z_1^{(i)} (z_1^{(i)})^T;$

end for

end if

3.3. Cooperative constrained particle filtering

When particle filtering is performed for state estimation in time series, we firstly conduct a rough estimation of the target state based on the center point of the particle set. Then, we adopt two error ellipses with different confidence levels (s_1, s_2) to resample the particles. After

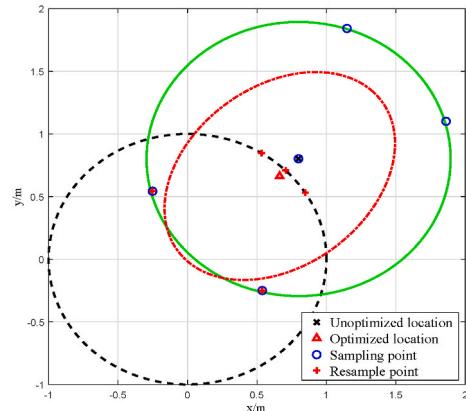


Fig. 5. A diagram of error ellipse resampling under distance constraints: solid ellipse represents the error ellipse before the constraint optimization, and circle points represent the sampling points before sampling. If these sampling points are outside the distance constraint (black dotted circle), then they will be mapped to the boundary of the constraint circle. The resampled points are shown as red plus signs in the figure. The error ellipse area after optimization is smaller, and the center of the ellipse (red triangle) is closer to the constraint boundary than that before optimization (black cross). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

the resampling stage, spatial cooperative constrained optimization is used to drive the estimated value closer to the true position, that is, the optimized position estimate is used to update the original center point. Furthermore, the filter estimation at the next moment will benefit from the spatial information at the previous moment, so as to achieve cooperative tracking of spatial-temporal fusion. The overall description of our proposed cooperative constrained PF is illustrated in [Algorithm 3](#).

Algorithm 3

Multi-target cooperative particle filter algorithm (CEER-PF)

Require: $[X_0] \leftarrow$ Initial position
Ensure: $\{\hat{m}_k\}_{k=1}^K \leftarrow$ State estimate
for $k \leftarrow 0, K - 1$ **do**
 if $k = 0$ **then**
 $X_0^{i=1:N} \sim N(X_0, \phi)$;
 else
 for do $i \leftarrow 1, N$
 $X_k^i \leftarrow f_1(X_{k-1}^i)$ % State transition
 $w_k^i \leftarrow f_2(Z_k)$ % Computational likelihood
 end for
 $\{\tilde{X}_k^i, \tilde{w}_k^i\}_{i=1}^N, \hat{X}_k \leftarrow \{\{X_k^i, w_k^i\}_{i=1}^N, (x_p, x_p)\}$ \triangleright [Algorithm 1](#)
 $m_1 = \bar{X}_k, C_1 = \text{cov}(\tilde{X}_k^{1:N}) \leftarrow$ Mean and covariance of particles at the current moment
 $m_l = [m_1, m_2]^T, C_l = [C_1, C_2]^T \leftarrow$ Spatial status information
 $(m_1, m_2) \leftarrow \text{SCCO}(m_l, C_l, Z_{12})$ \triangleright [Algorithm 2](#)
 $(x_p, y_p) \leftarrow f_1(m_1) \leftarrow$ Update estimate center
 end if
 end for

4. Theoretical performance analysis

In this section, we carry out the theoretical derivation to prove the validity of the proposed algorithm. Firstly, posterior Carmér-Rao Lower Bound (PCRLB) of multi-target cooperative tracking accuracy is derived. Next, the complexity analysis of both proposed and comparative methods are conducted.

4.1. Posterior Cramér Rao Lower Bound

The posterior Cramér Rao Lower Bound (PCRLB) ([Win et al., 2018](#)) is defined as the inverse of the Fisher information matrix (FIM), which provides the theoretical lower bound of the unbiased estimated variance, i.e.,

$$E\left\{\left(\hat{X}_k - X_k\right)^2\right\} \geq J(X_k)^{-1} \quad (35)$$

Considering the information in time series, the joint probability density function is:

$$p(\hat{d}_{k,1}, \hat{d}_{k,2}, \hat{\theta}_k, \hat{X}_k) = \left\{\prod_{j=1}^{N_a} p(\hat{d}_{k,1}|X_k)\right\} p(\hat{d}_{k,2}|X_{k-1}, X_k) p(\hat{\theta}|X_{k-1}, X_k) \quad (36)$$

where the simplified symbols $\hat{d}_{k,1}$ and $\hat{d}_{k,2}$ represent the inter-node distance measurement $\hat{d}_{k,nj}$ and the step measurement $\hat{d}_{k,n}$, respectively. $p(\hat{d}_{k,nj}|X_k)$, $p(\hat{d}_k|X_{k-1}, X_k)$ and $p(\hat{\theta}|X_{k-1}, X_k)$ could be obtained from Equation (3) ~ (7). According to the joint probability density function, we define the Fisher information matrix as:

$$J(k)_{n,j} = -E\left[\frac{\partial \ln p(\hat{d}_{k,1}, \hat{d}_{k,2}, \hat{\theta}_k, \hat{X}_k)}{\partial \hat{X}_{k,n}, \partial \hat{X}_{k,j}}\right], i, j \in N_a \quad (37)$$

When the target node travels at a constant speed, the dynamic model (2)

can be expressed as:

$$X(k) = F(k-1)X(k-1) + u(k-1) \quad (38)$$

where $F(k) = \text{diag}(F_1(k), F_2(k), \dots, F_{N_a}(k))$ is the block diagonal state transition matrix. u is an independent process noise vector, and the covariance matrix $Q(k) = \text{diag}(Q_1(k), Q_2(k), \dots, Q_{N_a}(k))$ is given. Thus, the vector $X(k) = [X_1^T(k), X_2^T(k), \dots, X_{N_a}^T(k)]^T$ reflects the augmented state information of all mobile nodes. Then, the recursive calculation of FIM is expressed as ([Win et al., 2018](#)):

$$J(k) = H_{k-1}^{22} - H_{k-1}^{21}\left(J(k-1) + H_{k-1}^{11}\right)^{-1}H_{k-1}^{12} + J_z(k) \quad (39)$$

where the matrix H and $J_z(k)$ are described as follows:

$$H_{k-1}^{11} = E\left\{-\Delta_{X(k-1)}^{X(k-1)} \ln p(X(k)|X(k-1))\right\} \quad (40)$$

$$H_{k-1}^{12} = E\left\{-\Delta_{X(k-1)}^{X(k)} \ln p(X(k)|X(k-1))\right\} = (H_{k-1}^{21})^T \quad (41)$$

$$H_{k-1}^{22} = E\left\{-\Delta_{X(k)}^{X(k)} \ln p(X(k)|X(k-1))\right\} \quad (42)$$

$$J_z(k) = E\left\{-\Delta_{X(k)}^{X(k)} \ln p(Z(k)|X(k-1))\right\} \quad (43)$$

where Δ_a^b represents the second-order derivative of vectors a and b , namely $\Delta_a^b = \nabla_a \nabla_b$. When the state transition model (2) is linear Gaussian, the FIM recursive calculation form described by Equation (39) could be simplified as:

$$J(k) = [Q(k-1) + F(k-1)J(k-1)^{-1}F(k-1)^T]^{-1} + J_z(k) \quad (44)$$

Since the distance measurement are generated by the mobile node independently, we could obtain the representation of the matrix $J_z(k)$ under the assumption of distributed Gaussian noise, namely

$$J_z(k) = \begin{bmatrix} R(1, 1) & R(2, 1) & \cdots & R(N_a, 1) \\ R(1, 2) & R(2, 2) & \cdots & R(N_a, 2) \\ \vdots & \vdots & \ddots & \vdots \\ R(1, N_a) & R(2, N_a) & \cdots & R(N_a, N_a) \end{bmatrix} \quad (45)$$

where the sub-matrix $R(n, j) = \begin{bmatrix} \chi_{x_j}^{x_n} & \chi_{x_j}^{y_n} & 0 & 0 \\ \chi_{y_j}^{x_n} & \chi_{y_j}^{y_n} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$. The parameters $\chi_{x_j}^{x_n}$,

$\chi_{y_j}^{y_n}$, and $\chi_{y_j}^{x_n}$ are respectively represented as:

$$\chi_{x_j}^{x_n} = -\frac{1}{\sigma_{nj}^2} \frac{(x_n - x_j)^2}{d_{nj}^2} \psi_{nj} \quad (46)$$

$$\chi_{y_j}^{y_n} = -\frac{1}{\sigma_{nj}^2} \frac{(y_n - y_j)^2}{d_{nj}^2} \psi_{nj} \quad (47)$$

$$\chi_{y_j}^{x_n} = -\frac{1}{\sigma_{nj}^2} \frac{(x_n - y_j)(y_n - y_j)}{d_{nj}^2} \psi_{nj} \quad (48)$$

In above equations, d_{nj} is the inter-node true distance, and σ_{nj}^2 is the variance of ranging noise. ψ_{nj} is an indicator function, which is set as 0 when n is equal to j , and 1 in all other cases. The FIM at the initial moment could be obtained from the ranging information, namely $J(0) = J_z(0)$. In the next section, we will take advantages of this derived PCRLB to compare with the statistical results of numerical simulations.

4.2. Complexity analysis

In our proposed cooperative tracking schema, the system overhead mainly consists of two parts, namely arithmetic computation in the local node and communication overhead between neighbors.

In the respect of arithmetic computation in the local node, our proposed algorithm needs to complete the screening and replication of N individual particles to realize the resampling optimization. Considering the extreme case, based on the proposed EER-PF resampling algorithm, if only particle screening operation is performed and no operations such as replication are performed, the complexity of the resampling algorithm is $O(N)$. In the worst case, it is necessary to copy c times for each particle, and the complexity of the resampling algorithm is $O(N^*c)$, where c is constant. Since the number of particles remains unchanged after resampling, the number of high-weight particles N_h will not exceed $N/3$, and the maximum number of replications will be no greater than 2. Thus, the complexity of the resampling algorithm remains as $O(N)$. Then, we can know the overall complexity of the particle filter algorithm is $O(N^*\Lambda^2)$, where N and Λ denote the quantity of the particles and dimension of the position variable respectively.

For cooperative algorithms, additional spatial distance constraints are calculated, which is related to the number of target nodes N_a in the whole network. UKF and our proposed algorithm are both determined by sigma point sampling. Therefore, the complexity of cooperative algorithms is directly related to the state dimension Λ , which means that the complexity of proposed CEER-PF is $O(N_a^*N^*\Lambda^2)$, while that of cooperative Unscented Kalman Filter (CUKF) is $O(N_a^*\Lambda^3)$. A detailed comparison of the algorithm complexity is shown in Table 2.

In terms of broadcast communication between neighbors, all communication overhead is introduced by a distance ranging and paired status interaction. We currently adopt the principle of maximum matches number first, that is, the nodes that are not optimized will be paired first. To avoid multiple optimizations for a single node, if the current node is paired, we will mark its status as optimized. When it is paired again, only the optimization of the other node will be performed.

Above all, theoretical analysis shows that the current strategy indeed works in improving the performance of the algorithm, with a reasonable computation and communication overhead. Further verification will be demonstrated in subsequent simulation experiments.

5. Experiments and discussion

In this section, we carried out numerical simulations and physical experiments to validate the effectiveness of our proposed algorithm. Firstly, numerical simulations are conducted for both single and multi-target tracking with the use of our proposed spatial-temporal constrained particle filtering method. Then, a typical physical scenario was set up to prove its effectiveness and wide application value.

5.1. Experimental setup

We carried out a random walk numerical simulation experiment based on MATLAB. The experimental computer runs Windows 10 Professional Edition system, with 4-core i5 CPU and 16 GB memory. The experimental scene is set as 50 m*50 m, and the target node travels 100 steps, starting from the initial position (0, 0). The trajectories of target

Table 2
A detailed comparison of the algorithm complexity.

Algorithms	Complexity
SYR-PF	$O(N^*\Lambda^2)$
STR-PF	$O(N^*\Lambda^2)$
EER-PF	$O(N^*\Lambda^2)$
CUKF	$O(N_a^*\Lambda^3)$
CEER-PF	$O(N_a^*N^*\Lambda^2)$

node movements are recorded. The basic parameter settings are displayed in Table 3. Numerical simulations confirm the efficiency of the scheme put forth.

5.2. Single target tracking

Typical comparison algorithms are chosen as the non-resampling PF algorithm (NR-PF), unscented Kalman filter algorithm (UKF) (Lee and McBride, 2019), systematic resampling PF (SYR-PF) (Ala-Luhtala et al., 2016), and stratified resampling PF (STR-PF) (Li et al., 2015). All algorithms are performed and predict the position reached by each travel of the target node. Euclidean distances between the predicted position and the real position are counted.

In order to verify that the particle filter algorithm based on error ellipse resampling (EER-PF) can effectively alleviate the problem of sample degeneracy and impoverishment, we performed a simulation of state estimation on a single-target random walking model. Fig. 6 shows the weight distribution of STR-PF, SYR-PF, and EER-PF. The particle weights are displayed in descending order, and the confidence interval of the EER-PF is (0.125, 0.5).

It can be drawn from Fig. 6 that SYR-PF and STR-PF appear sample degeneracy and impoverishment at the beginning of the iteration. The number of particles occupying a larger weight and that of particles with a weight of 0, decreases as the iteration progresses, which verifies that SYR-PF and STR-PF only slow down sample degeneracy and impoverishment, but not completely. The EER-PF proposed in this paper has a relatively balanced weight distribution and effectively improves particle degradation and depletion problems.

During the above experiment, tracking errors were also counted. The error is expressed as the Euclidean distance between the predicted position and the true position, which is defined as follows:

$$e = \left\| \hat{P}_k - P_k \right\|_2 \quad (49)$$

We can draw the following conclusions from Fig. 7-(a). The positioning error of the target node shows a gradually increasing trend as the target moves over time, which to some extent verifies the cumulative error and drift problem of the sole IMU. Among them, the particle filter algorithm based on error ellipse resampling (EER-PF) has a smaller error growth rate than the others and is closer to the PCRLB, which also proves that EER-PF can suppress the cumulative error to a certain extent. Compared with SYR-PF and STR-PF, UKF has higher positioning accuracy, but it is slightly inferior to EER-PF.

To further verify the effectiveness and stability of the EER-PF algorithm in single target tracking, the root mean square error and experimental variance of the positioning algorithms are counted. The root mean square error (RMSE) is defined as:

$$RMSE = \sqrt{\frac{1}{K-1} \sum_{k=1}^K (e)^2} \quad (50)$$

From Fig. 7-(b), we could see that the resampled algorithm curve is

Table 3
Experimental parameter settings.

Parameter	Numerical value
Sampling interval t_s	1(s)
Acceleration a	0(m/s)
Process noise variance σ^2	0.1(m ²)
Initial noise variance γ_0^2	0.1(m ²)
Step noise variance $\gamma_{1,k}^2$	0.5(m ²)
Angle noise variance $\gamma_{2,k}^2$	10°
Distance measurement noise variance $\gamma_{3,k}^2$	0.5(m ²)
Confidence probability (α_1, α_2)	(0.125, 0.5)
Number of particles N	4000

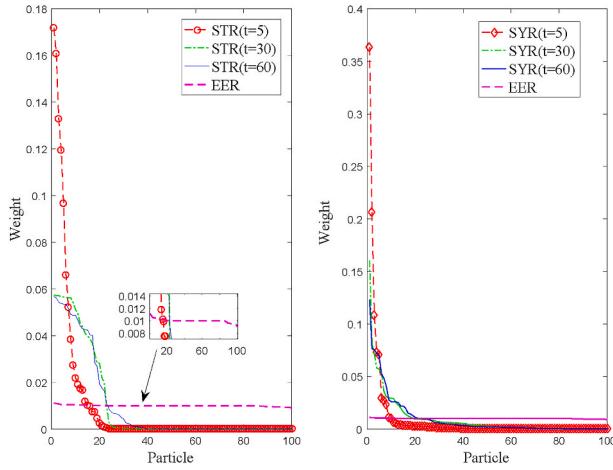


Fig. 6. Particle weight distribution of different algorithms.

obviously lower than that of non-resampled, which is closer to the true value. This is probably caused by the fact that the posterior distribution can be obtained using Monte Carlo sampling. Besides, the error distributions of SYR-PF and STR-PF are very similar, while the performance of STR-PF is slightly better than that of SYR-PF due to the diversity of particles screened by stratification. Anyway, EER-PF holds the best performance.

Fig. 8 shows cumulative distribution function curve of the root mean square error, and various resampling algorithms are repeated 100 random walk experiments. There is a 97% probability that the positioning accuracy of error ellipse resampling particle filter (EER-PF) proposed in this paper is less than 2 m, and there is a 50% probability that it is less than 1 m.

In order to show how confidence probabilities impact on the positioning accuracy in proposed algorithm, 6 sets of confidence intervals are selected for simulation of single target random walking. The positioning accuracy of EER-PF under different confidence probabilities is represented by the root mean square error as shown in Table 4, from which the following conclusions could be drawn:

- 1) If the overall confidence probabilities are smaller, that is, the number of negligible particles outside the outer ellipse is larger, the positioning accuracy of EER-PF is better. This is because the weights of negligible particles are redistributed to dominant ones, which realizes the optimization of particle screening.
- 2) If the span of the confidence interval is larger, that is, the number of moderate particles between the two ellipses is larger, the algorithm can provide higher positioning accuracy. Generally, it can be seen that setting the confidence interval to (0.125, 0.5) is a more appropriate choice.

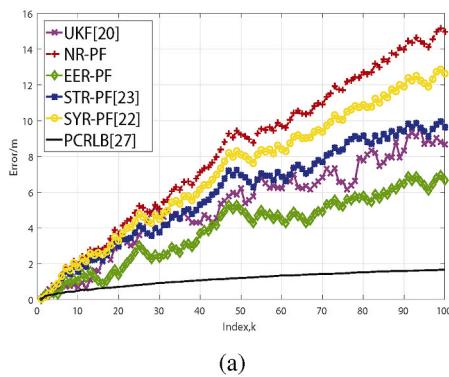


Fig. 7. Statistical results of typical single target tracking. (a) A typical error distribution without cooperative measurements. (b) Statistical distribution of RMSE.

5.3. Multiple targets tracking

Cooperative tracking can effectively integrate the information of a single target node to achieve higher-precision tracking. In this paper, we proposed a constrained particle filter method for cooperative tracking to reduce the impacts of cumulative errors caused by inertial measurements.

In order to verify the effectiveness and advancement of the error-ellipse-resampling-based particle filter for cooperative tracking(CEER-PF), we use CEER-PF and Unscented Kalman Filter for cooperative tracking(CUKF) to repeat random walk experiments 100 times and compare with the PCRLB under the same measurement noise. The mean value of the root mean square error for cooperative tracking is defined as:

$$CRMSE = \frac{1}{N_a} \sum_{i=1}^{N_a} \sqrt{\frac{1}{K-1} \sum_{t=1}^K (e)^2} \quad (51)$$

The statistics of the experimental results are shown in Fig. 9-(a). The following conclusions can be obtained from the figure:

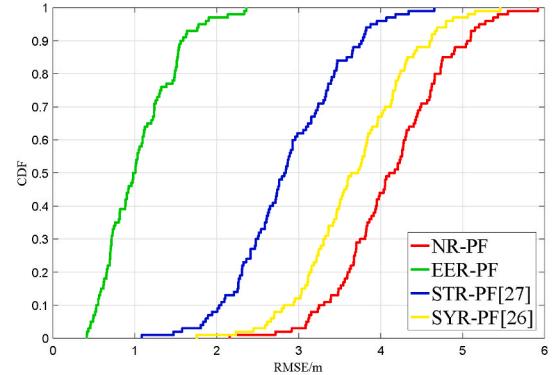
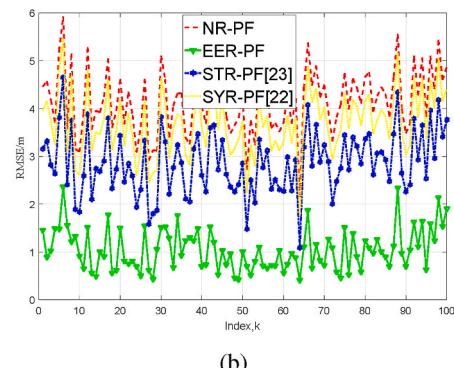


Fig. 8. Cumulative distribution function (CDF) curves of RMSE with non-cooperative algorithms.

Table 4

The influence of confidence interval on algorithm accuracy.

Confidence interval	Tracking accuracy
(0.6, 0.8)	2.30
(0.4, 0.6)	1.70
(0.4, 0.8)	1.86
(0.25, 0.5)	1.16
(0.125, 0.5)	1.14
(0.125, 0.5)	1.05



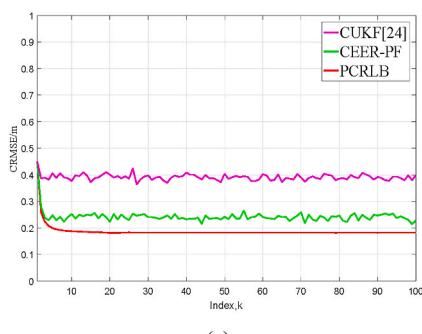
- 1) Generally, the errors of both cooperative algorithms are relatively stable, and no obvious error accumulation is observed. This to some extent verifies the effectiveness of the algorithm proposed in this paper on restraining the influence of accumulated errors.
- 2) CEER-PF has a higher positioning accuracy than CUKF, and it is closer to the cooperative PCRLB curve, that is, closer to the optimal value of the theoretical situation. Among them, the positioning accuracy of the CUKF algorithm can reach 0.39 m, while CEER-PF could reach to 0.24 m.

Fig. 9-(b) shows the cumulative distribution function of different cooperative tracking algorithms. The positioning errors of CUKF and CEER-PF are both below 1 m. Among them, the probability that CUKF and CEER-PF algorithms are better than 0.5 m are respectively 76% and 98%, respectively. This confirms that the CEER-PF algorithm proposed in this paper can achieve higher precision for multiple target tracking.

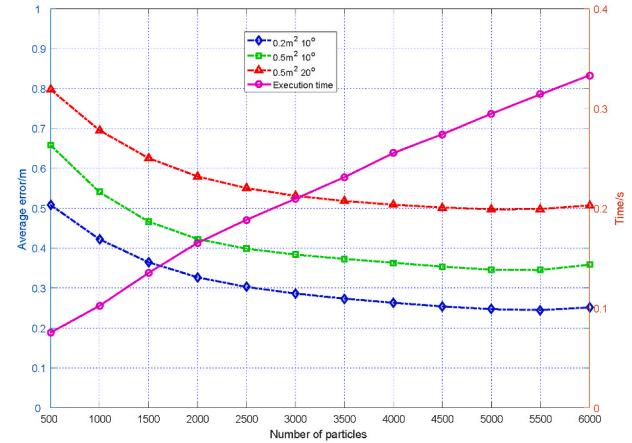
Furthermore, we discuss how measurement noise and the number of particles impact on the performance of proposed algorithm. We selected 3 sets of parameters for comparative experiments. Correspondingly, the measurement noise is step noise and angle noise, and the variance parameter (0.2, 10), (0.5, 10), (0.5, 20) are selected. CEER-PF is used to estimate the position of the target, and RMSE is counted to represent the positioning accuracy. The statistical results are shown in **Fig. 10**, from which following conclusions could be drawn:

- 1) The blue curve in the figure is lower than the green one, which means that when the variance of the angle noise is unchanged, the smaller the variance of step length noise, the more accurate the positioning performance of CEER-PF.
- 2) The green curve is lower than the red one, indicating that when the variance of step length noise is unchanged, the smaller the variance of angle noise, the more accurate the positioning performance of CEER-PF.
- 3) As the number of particles increases from 500 to 6000, the execution time of the algorithm shows a linear growth trend. However, when the number of particles increases to 4000, the performance reaches its optimum. Thus, the choice of 4000 is more suitable in practice use, which could achieve as high accuracy as possible with acceptable time overhead.
- 4) Generally, one location estimate usually takes 0.1 ~0.2 s, which indicates that if the speed of moving target is 5 m/s (the normal walking speed of a pedestrian), the target moves only 0.5 ~1.0 m, which is quite reasonable for pedestrian tracking applications. If 80 km/h (a fast-moving car), the target moves ~2 m during this period, which may still meet the requirements of most applications.

Then, how target nodes' number impacts on the positioning performance for multi-target cooperative tracking is taken into consideration. The following conclusions could be drawn from the experimental results



(a)

**Fig. 10.** Performance analysis of CEER-PF.

shown in **Fig. 11**:

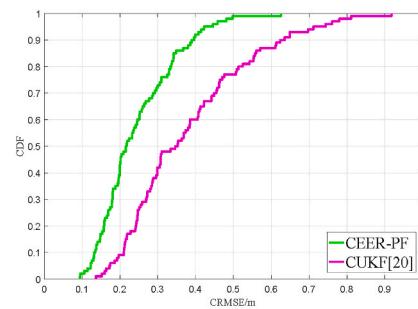
- 1) When target nodes' number increases, the positioning error of both algorithms didn't change dramatically. It indicates that both of them are suitable for large-scale deployment applications, while CEER-PF is significantly better than CUKF.
- 2) The execution time of both algorithms increases with the number of target nodes, while that of CEER-PF is slightly higher than that of CUKF. However, it is completely acceptable for large-scale real-time applications.

Table 5 counts the positioning accuracy of different algorithms, summarized as follows:

- 1) The accuracy of EER-PF in single target positioning can reach 1.05 m, which have a 63% improvement in accuracy compared with STR-PF. The variance is smaller than the others as well, that is, the algorithm consistency is better. Due to the implementation of particle replication, the time consumption of EER-PF is slightly larger, but it can also meet the real-time requirements.
- 2) Compared with CUKF, CEER-PF needs more time for particle calculation, but it can also to a great extent improve the positioning accuracy on the basis of meeting the real-time requirements of the whole system. Compared with EER-PF without cooperation, its accuracy performance has been improved by nearly 77%. Compared with CUKF tracking, it has progressed by about 38%.

5.4. Practical use case

In order to better verify the effectiveness and application value of our proposed method, a multi-target tracking experiment for cooperative



(b)

Fig. 9. Statistical results of typical multiple target tracking. (a) Error distribution with cooperative measurements. (b) Cumulative distribution function (CDF) curves of CRMSE with cooperative algorithms.

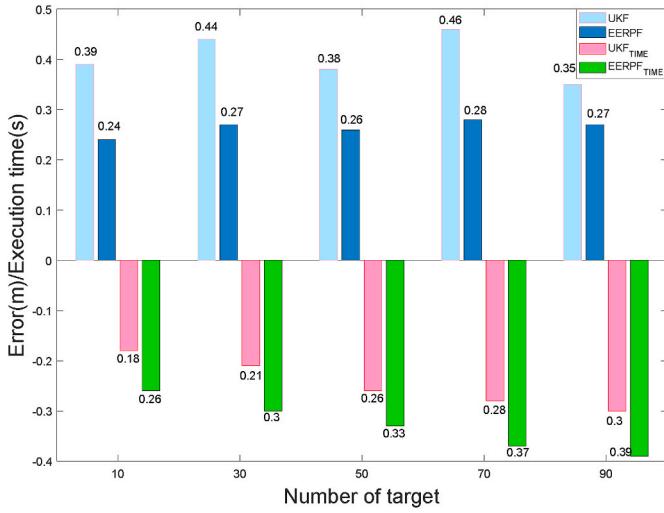


Fig. 11. The number of Target nodes' influence on the localization performance of various algorithms.

Table 5
Positioning accuracy of various algorithms ($N = 4000$).

Algorithms	Maximum variance (m^2)	Minimum variance (m^2)	Average error (m)	Execution time (s)
NR-PF	35.41	4.70	4.15	0.10
STR-PF	21.89	1.19	2.87	0.12
SYR-PF	30.13	3.11	3.67	0.15
EER-PF	5.63	0.17	1.05	0.22
CUKF	3.41	0.22	0.39	0.18
CEER-PF	0.29	0.03	0.24	0.26

robots is carried out. The wheeled navigation robot adopted in 2D tracking applications, as shown in Fig. 12-(a), has a modular structure, mainly including the main board (provides a variety of interfaces, easy to expand), the core board (processor cortex-m4 STM32F405RG + 802.15.4 standard RF chip Atmel At86rf231), sensor board (integrated with a variety of sensors, including IMU and TOA module). The core board is connected with other equipment through the mainboard so that the system can be disassembled and expanded.

A minimized sensor board is specially designed, as shown in Fig. 12-(b). It is aiming to collect spatial and temporal information during the moving process, which covers accelerated velocity, angular velocity, and the distances between targets. Among these all, distance information is especially special when compared with other platforms. Each sensor board has a 6-axes sensor (MPU6050, which integrates a triaxial

accelerometer and a triaxial gyroscope), a barometer sensor (MS5611), and a UWB-TOA ranging module (DWM1000). The MEMS sensors are connected to a microcontroller (STM32F103) for the sake of sampling efficiency at a rate of 10Hz.

To verify the effectiveness, practical experiments are conducted to compare the proposed method with above-mentioned state-of-the-art methods. A total of 10 navigation robots are programmed to travel randomly by applying the collision-avoidance algorithm (Thrun, 2002), and each robot maintains a distance-ranging frequency of once a second with its neighboring nodes. A high precision optical motion tracking system (Mars 2H) (<https://www.nokov.com/en/>, 2020) is deployed so as to provide the ground truth measurements. Considering the size of the tracking target and the deployment requirements of optical devices, an open field covered 10 m * 10 m is chosen as a testing scenario.

As shown in Fig. 13, EER-PF and CEER-PF, as well as state-of-the-art algorithms are tested to show the superiorities of our proposed method. The results are similar to those in the numerical simulation, namely 1) the accuracy of cooperative algorithms is better than those of non-cooperative ones. 2) CEER-PF performs better than the state-of-the-art CUKF (Chen et al., 2019b). 3) Although the experimental results of simulation and actual verification are very similar, a slightly larger error is introduced by physical tests, possibly caused by unpredictable environmental noise and hardware control parameters.

6. Conclusion

In this paper, we proposed a layered resampling algorithm based on error-ellipse constraints. According to the error distribution characteristics of particles, error ellipses with different confidence probabilities are established to achieve layered screening. Compared with other resampling algorithms, the error-ellipse-resampling algorithm proposed in this paper could effectively alleviate sample degeneracy and impoverishment, thereby improving the positioning accuracy.

In order to further solve the accumulative errors problem faced by sole IMU, a multi-target cooperative constrained particle filter algorithm is proposed. Combined with the instantaneous, high-precision, and non-accumulative characteristics of TOA-based distance measurement, the estimated results in time series are used as prior knowledge of spatial cooperative optimization. Based on the constrained conditions of inter-node distance measurements, the optimized posterior position estimation is obtained to achieve spatial-temporal fusion in cooperative tracking. Experimental results verify that CEER-PF could greatly improve the positioning performance, and effectively eliminate the accumulative errors. Compared with the non-cooperative method, its positioning performance has been improved by nearly 77%. Compared with state-of-the-art CUKF, it has progressed by about 38%.

Furthermore, in this paper, the pair-wise optimization for two nodes is adopted. To avoid multiple optimizations for a single node, if the

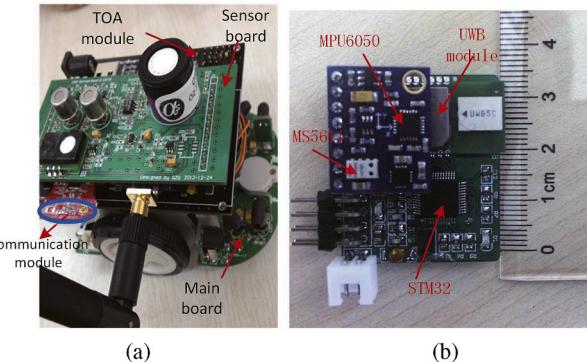


Fig. 12. Experimental Platform Settings. (a) The wheeled navigation robot adopted in this paper. (b) A minimized sensor board specially designed for targets localization.

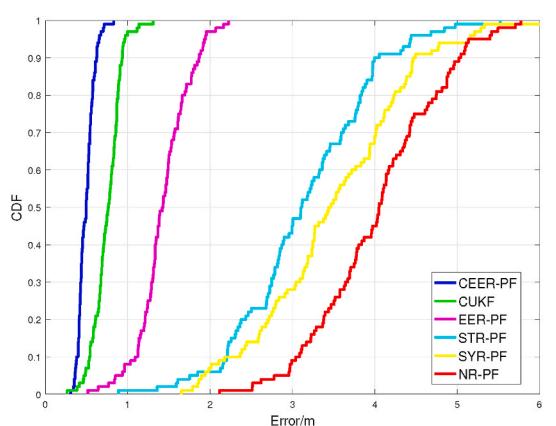


Fig. 13. Experiment results of demonstrated practical use case.

current node is paired, we will mark its status as optimized. When it is paired again, only the optimization of another node will be performed. This leaves us some room for future studies. Joint optimization for all target nodes and consensus communication protocol may further improve the performance.

Credit author statement

Cheng Xu: Conceptualization, Methodology, Supervision, Writing-Original draft, Writing- Reviewing and Editing, Funding acquisition. Xinxin Wang: Methodology, Validation, Writing- Original draft, Writing-Reviewing and Editing. Shihong Duan: Investigation, Methodology, Funding acquisition. Jiawang Wan: Methodology, Investigation.

Declaration of competing interest

The authors declare no conflict of interest.

Acknowledgment

This work is supported in part by China National Postdoctoral Program for Innovative Talents under Grant BX20190033, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515110325, in part by Project funded by China Postdoctoral Science Foundation under Grant 2020M670135, in part by Postdoctor 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.

References

- Ahmed, Hamad, Tahir, Muhammad, 2017. Improving the accuracy of human body orientation estimation with wearable IMU sensors. *IEEE Transactions on instrumentation and measurement* 66 (3), 535–542.
- Ahn, Choon Ki, et al., 2019. Bayesian state estimation for Markovian jump systems: employing recursive steps and pseudocodes. *IEEE Systems, Man, and Cybernetics Magazine* 5 (2), 27–36.
- Ala-Luhtala, Juha, et al., 2016. An introduction to twisted particle filters and parameter estimation in non-linear state-space models. *IEEE Trans. Signal Process.* 64 (18), 4875–4890.
- Bianchi, Valentina, et al., 2019. IoT wearable sensor and deep learning: an integrated approach for personalized human activity recognition in a smart home environment. *IEEE Internet of Things Journal* 6 (5), 8553–8562.
- Candy, James V., 2016. *Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods*, vol. 54. John Wiley & Sons.
- Chen, Huihui, et al., 2019a. Crowdtracking: real-time vehicle tracking through mobile crowdsensing. *IEEE Internet of Things Journal* 6 (5), 7570–7583.
- Chen, Diliang, et al., 2019b. Smart insole-based indoor localization system for Internet of Things applications. *IEEE Internet of Things Journal* 6 (4), 7253–7265.
- Duan, S., Su, R., Xu, C., Chen, Y., He, J., et al., July 2020. Ultra-wideband radio channel characteristics for near-ground swarm robots communication vol. 19, no. 7. In: *IEEE Transactions on Wireless Communications*, pp. 4715–4726. <https://doi.org/10.1109/TWC.2020.2986446>.
- Enayati, Nima, De Momi, E., Ferrigno, G., 2015. A quaternion-based unscented kalman filter for robust optical/inertial motion tracking in computer-assisted surgery. *IEEE Transactions on Instrumentation & Measurement* 64 (8), 2291–2301.
- Fan, Chengfei, et al., 2019. An extended joint spatial and temporal cooperation model for the range-based localization problem. *IEEE Trans. Veh. Technol.* 68 (12), 12121–12134.
- Fu, Xiaoyan, Jia, Yingmin, 2010. An improvement on resampling algorithm of particle filters. *IEEE Trans. Signal Process.* 58 (10), 5414–5420.
- Ghobadi, Mostafa, Singla, Punit, Esfahani, Ehsan T., 2017. Robust attitude estimation from uncertain observations of inertial sensors using covariance inflated multiplicative extended Kalman filter. *IEEE Transactions on Instrumentation and Measurement* 67 (1), 209–217.
- <https://www.nokov.com/en/products/motion-capture-cameras/Mars.html>. (Accessed 29 September 2020).
- Huang, Chengliang, Liao, Zaiyi, Zhao, Lian, 2010. Synergism of INS and PDR in self-contained pedestrian tracking with a miniature sensor module. *IEEE Sensor. J.* 10 (8), 1349–1359.
- Jonasson, Mats, et al., 2020. Inertial navigation and position uncertainty during a blind safe stop of an autonomous vehicle. *IEEE Trans. Veh. Technol.* 69 (5), 4788–4802.
- Lee, Sangjin, McBride, James, 2019. Extended object tracking via positive and negative information fusion. *IEEE Trans. Signal Process.* 67 (7), 1812–1823.
- Li, Wei, Yang, Zhen, Hu, Haifeng, 2012. Sequential particle-based sum-product algorithm for distributed inference in wireless sensor networks. *IEEE Trans. Veh. Technol.* 62 (1), 341–348.
- Li, Tiancheng, Bolic, Miodrag, Djuric, Petar M., 2015. Resampling methods for particle filtering: classification, implementation, and strategies. *IEEE Signal Process. Mag.* 32 (3), 70–86.
- Liu, Hengzhi, et al., 2020. Application research of an array distributed IMU optimization processing method in personal positioning in large span blind environment. *IEEE Access* 8, 48163–48176.
- Nicholas, Capurso, et al., 2018. A survey on key fields of context awareness for mobile devices. *J. Netw. Comput. Appl.* 118, 44–60.
- Thrun, Sebastian, 2002. Probabilistic robotics. *Commun. ACM* 45 (3), 52–57.
- Vagle, Niranjana, Broumandan, Ali, Lachapelle, Gérard, 2018. Multiantenna GNSS and inertial sensors/odometer coupling for robust vehicular navigation. *IEEE Internet of Things Journal* 5 (6), 4816–4828.
- Wang, Xinxin, Cheng, Xu, Duan, Shihong, Wan, Jiawang, 2020a. Error-ellipse-resampling-based particle filtering algorithm for target tracking. *IEEE Sensor. J.* 20 (99), 5389–5397.
- Veeramalla, S.K., Talarji, V.H.R., 2020. Neural source localization using particle filter with optimal proportional set resampling. *ETRI J.* <https://doi.org/10.4218/etrij.2019-0020>.
- Wang, Xinxin, et al., 2020b. Error-ellipse-resampling-based particle filtering algorithm for target tracking. *IEEE Sensor. J.* 20 (10), 5389–5397.
- Win, Moe Z., Shen, Yuan, Dai, Wenhan, 2018. A theoretical foundation of network localization and navigation. *Proc. IEEE* 106 (7), 1136–1165.
- Xiao, Yuqi, Pan, Difu, 2020. Research on robust visual tracker based on multi-cue correlation particle filters. *IEEE Access* 8, 19555–19563.
- Xu, C., He, J., Zhang, X., et al., 2017. Toward near-ground localization: modeling and applications for TOA ranging error. *IEEE Trans. Antenn. Propag.* 65 (10), 5658–5662.
- Xu, Cheng, et al., 2018. Geometrical kinematic modeling on human motion using method of multi-sensor fusion. *Inf. Fusion* 41, 243–254.
- Xu, Cheng, et al., 2019. Optimal estimation and fundamental limits for target localization using IMU/TOA fusion method. *IEEE Access* 7, 28124–28136.
- Yuan, Yuan, Lu, Yuwei, Wang, Qi, 2017. Tracking as a whole: multi-target tracking by modeling group behavior with sequential detection. *IEEE Trans. Intell. Transport. Syst.* 18 (12), 3339–3349.
- Zachariah, Dave, et al., 2012. Bayesian estimation with distance bounds. *IEEE Signal Process. Lett.* 19 (12), 880–883.
- Zhao, Yingwei, 2016. Applying time-differenced carrier phase in nondifferential GPS/IMU tightly coupled navigation systems to improve the positioning performance. *IEEE Trans. Veh. Technol.* 66 (2), 992–1003.
- Zheng, Yang, et al., 2018. Exploiting AoA estimation accuracy for indoor localization: a weighted AoA-based approach. *IEEE Wireless Communications Letters* 8 (1), 65–68.
- Zihajehzadeh, Shaghayegh, Park, Edward J., 2016. A novel biomechanical model-aided IMU/UWB fusion for magnetometer-free lower body motion capture. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47 (6), 927–938.
- Zou, Han, et al., 2017. Accurate indoor localization and tracking using mobile phone inertial sensors, WiFi and iBeacon. In: *2017 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*. IEEE.



Cheng Xu received the B.E., M.S. and Ph.D. degree from the University of Science and Technology Beijing (USTB), China in 2012, 2015 and 2019 respectively. He is currently working as an associate professor in the Data and Cyber-Physical System Lab (DCPS) at University of Science and Technology Beijing. He is supported by the China National Postdoctoral Program for Innovative Talents from Chinese government in 2019. He is an associate editor of International Journal of Wireless Information Networks. His research interests now include swarm intelligence, multi-robots cooperation, wireless localization and hardware-software co-design. He is a member of the IEEE and CCF.



Xinxin Wang is currently working toward the Master degree at the Data and Cyber-Physical System Lab (DCPS), University of Science and Technology Beijing. His research interests include wireless localization, multi-robots network and internet of things.



Shihong Duan received Ph.D. degree in computer science from University of Science and Technology Beijing (USTB). She is an associate professor with the School of Computer and Communication Engineering, USTB. Her research interests include wireless indoor positioning, human gesture recognition and motion capture.



Jiawang Wan received the B.E. degree from the University of Science and Technology Beijing, China, in 2017, and he is currently working towards the Ph.D degree at University of Science and Technology Beijing. His research interests include wireless localization, swarm intelligence and Internet of things.