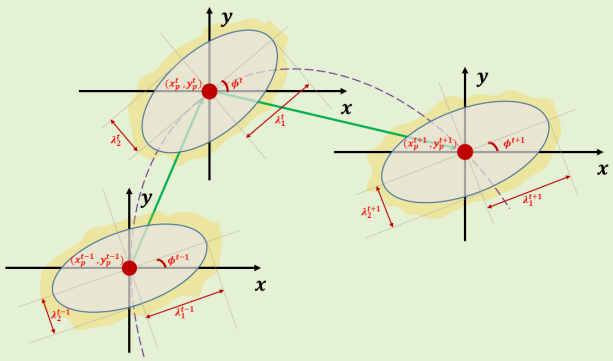
Error Constraint Enhanced Particle Filter Using Quantum Particle Swarm Optimization

Jiawang Wan, Cheng Xu, *Member, IEEE*, Yidan Qiao, and Xiaotong Zhang, *Senior Member, IEEE*

***Abstract*—The position is one of the foremost imperative properties of an object. The Inertial Measurement Unit (IMU) development has enabled the system to continuously decrease in size, reduce power consumption, and become more and more universal. Although IMU meets the real-time needs of the localization system, it faces accumulative error and drifting problems. This paper presents an error constraint enhanced particle filter algorithm using quantum particle swarm optimization (EC-QPF), which achieves high-precision position estimation of target tracking with IMU. First, we proposed a quantum particle swarm optimization-based resampling method taking the place of the traditional weight-based resampling method in the particle filter, which avoids the particle impoverishment problem and keeps the particles' diversity. The theoretical foundation of error constraint was raised applied to enhance the performance of the proposed particle filter. The error constraint is established utilizing the known estimated center and confidence scale to achieve particle screening based on the geometrical position of particles. Numerical experimental results show that the proposed EC-QPF has a 67% improvement instead of other modified particle filters and more efficiently eliminates the cumulative error. Furthermore, higher accuracy and stability are obtained under the exact condition, demonstrating better performance than traditional particle filter methods.**



***Index Terms*—Target tracking, particle filter, quantum particle swarm optimization, error constraint, accumulative error.**

I. Introduction

TARGET tracking has received extensive attention in various fields such as intelligent traffic management systems [1]–[3], and the central issue is the uncertainty of the target motion. Global Positioning System (GPS) [4] has been widely used in outdoor navigation. Still, it cannot deliver high accuracy in urban areas due to its weak signal blocked by density building materials. Traditional Radio Frequency Identification (RFID) [5] based positioning is an alternative method in areas that cannot be covered by GPS signals, including positioning technologies based on received signal strength (RSS) [6], time of arrival (ToA) [7], time difference of arrival (TDoA) [8] and Angle of Arrival (AoA) [9]. However, external infrastructure needs to be deployed in advance, which makes them unsuitable for positioning estimation in unknown areas [10]. Inertial navigation system (INS) that utilizes the inertial measurement unit (IMU) has been used in many commercial products due to its high accuracy and convenience [11]. However, it faces the problems of accumulative error and drifting, which to a great extent limits its applications [12].

The filtering methods provide a reliable solution for improving the positioning accuracy of IMU [13]. Jonasson *et al.* developed Extended Kalman Filter (EKF) [14] concepts to estimate the vehicle position during a safe stop. In [15], an improved Unscented Kalman Filter (UKF) is designed for robust estimation of position. Liu *et al.* [16] proposed an improved Particle Filter (PF) algorithm for reducing the long-term accumulative error inherent in inertial positioning.

Manuscript received August 23, 2021; accepted September 14, 2021. Date of publication September 16, 2021; date of current version October 29, 2021. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62101029, 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 the Project funded by China Postdoctoral Science Foundation under Grant 2020M670135, in part by the Postdoctoral Research Foundation of Shunde Graduate School, University of Science and Technology Beijing, under Grant 2020BH001, in part by the Fundamental Research Funds for the Central Universities under Grant 06500127, and in part by the Scientific and Technological Innovation Foundation of Shunde Graduate School, USTB, under Grant BK20BF009. The associate editor coordinating the review of this article and approving it for publication was Prof. Pierluigi Salvo Rossi. *(Jiawang Wan and Cheng Xu are co-first authors.) (Corresponding authors: Cheng Xu; Xiaotong Zhang.)*

The authors are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, and also with Shunde Graduate School, University of Science and Technology Beijing, Foshan 528399, China (e-mail: wjiawang@xs.ustb.edu.cn; xucheng@ustb.edu.cn; qiaoyidan@xs.ustb.edu.cn; zxt@ies.ustb.edu.cn).

Digital Object Identifier 10.1109/JSEN.2021.3113364

However, they could only suppress the growth rate of the cumulative errors to a certain extent but not eliminate it. 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 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 [17]. Since errors are introduced by linearization, UKF is still not suitable for higher-order non-linear system models [18].

Compared with other filter methods, although PF would pay more computational cost, it has better adaptability to non-linear non-Gaussian systems [19]. The particle filter utilizes the sequential Monte Carlo method to approximate posterior distribution using a set of weighted samples, so theoretically, it can represent any distribution. However, due to suboptimal sampling, generic particle filter possesses some disadvantages. (1) *Particle impoverishment*: this might happen when the likelihood is very narrow or the likelihood lies in the tail of prior distribution resulting in the overlap region of likelihood and prior distribution is very small. Consequently, only a small quantity of particles will have significant importance weights after the update process. Thus, the sample set only contains few dissimilar particles, and sometimes they will drop to a single sample after several iterations. As a result, essential samples (reasonable hypothesis) may be lost. (2) *Sample size dependency*: particle filter method highly relies on an ample number of particles to approximate a wide range of probability densities. Hence, a conflict arises between computational efficiency and the accuracy of the approximation. Thus, an estimation failure may occur if the sample sets are inadequate, especially when the system's initial state is unknown.

To solve those problems, some researchers have adopted analytical methods. Rudolph introduces an unscented particle filter (UPF) by combining the unscented Kalman filter (UKF) with a generic particle filter. The main idea of UPF is to use UKF to get a better proposal function to improve the sampling process. While UPF enhances the performance of generic particle filters by incorporating new observation into sampling to avoid particle impoverishment, it sacrifices computational efficiency. Other researchers tried to propose solutions to improve resampling, such as stratified resampling [20], systematic resampling [21], etc. Both resamplings are based on a layered idea. After a series of iterations, the number of particles will be reduced, resulting in most of the weights occupied by a bit of them. Thus, the final state estimation results of general methods are not always as satisfactory as expected. This paper proposed an error constraint enhanced particle filter algorithm using quantum particle swarm optimization to address these issues. The error constraint is constructed based on confidence probability, and the weight-based resampling method is replaced with the quantum particle swarm optimization-based resampling method. A particle set is implemented to solve the problem of particle impoverishment and sample size dependency. The main contributions of this paper are summarized as follows:

- We proposed a quantum particle swarm optimization-based particle resampling method, which avoids the problem of particle impoverishment in the particle filter. We replaced the traditional weight-based resampling method with a novel quantum particle swarm optimization-based resampling method as the breakpoint to differentiate the resampling process. The proposed resampling method keeps the diversity of particles, which enhances the performance of particle filter and decreases the dependency of sample size.
- We established the theoretical foundation of error constraint to improve the accuracy of the proposed particle filter. Based on the IMU estimated center and confidence scale, error constraint with confidence probability is established, and range constraint based on geometric particle position is implemented, which benefits the accuracy and stability of target tracking estimation.

The remainder of this paper is organized as follows: Section II thoroughly explains the key issues that need to be solved in this paper. The related definitions are defined, and the model description of the problem is established. Section III describes the specific process of proposed error constraint enhanced particle filter using quantum particle swarm optimization. Section IV shows the results and discussion of numerical experiments. Section V summarizes the paper.

II. Problem Statement

A. Dynamic Model

Consider a target node, represented by $\mathcal{T}$, which is outfitted with an inertial measurement unit. The measurement process and state transition process are acted in discrete time $t_k, k = 0, 1, 2, \ldots, K$. $X_k \in \mathbb{R}^4$ characterized as the state information of node $\mathcal{T}$ at time t_k, which incorporates coordinate vector $P_k = [x_k, y_k]^T$ and velocity vector $V_k = \left[v_x^k, v_y^k\right]^T$. At that point, X_k could be expressed as:

$$X_k = [P_k, V_k]^T \tag{1}$$

For the target tracking issue, we present the dynamic model firstly. It is accepted that the motion law of node $\mathcal{T}$ conforms to the dynamic random walk process [22]. In this way, we utilize the first-order hidden Markov model for dynamic modeling, and the state X_{k-1} of node $\mathcal{T}$ at time t_{k-1} would contribute to the state X_k at time t_k, denoted as:

$$X_k = f(X_{k-1}, A_{k-1}, \xi) \tag{2}$$

The acceleration at time t_{k-1} is notated as $A_{k-1} = [a_x^{k-1}, a_y^{k-1}]^T$, and the components in both directions are generated from $\{0, -g, g\}$ randomly with probabilities modeled as a random Markov jump [23]. The process noise generated during a state transition is notated as $\xi = \left[\xi_x, \xi_y\right]^T$, which is Gaussian distributed noise with mean 0 and variance σ^2, i.e., $\xi_x, \xi_y \sim \mathcal{N}\left(0, \sigma^2\right)$. The target node's measurement is expressed as a parameter related to the state. In real physical measurements, step size and heading angle measurement could be obtained by integrating the acceleration and angular velocity. In this paper, we assumed the noise of measurement to be Gaussian distributed noise, which simplified the issue

TABLE I
NOTATIONS

Symbol	Description
$\mathcal{T}$	Target node
t_k	Sampling interval
X	State information
$\hat{X}$	Estimation of state
P	Coordinate vector
x	Coordinate
V	Velocity vector
v	Velocity
A	Acceleration vector
a	Acceleration
ξ	Process noise
d	Actual step size
$\hat{d}$	Step size measurement
ε_1	Gaussian step size noise
δ_1^2	Noise variance of step size
θ	Actual heading angle
$\hat{\theta}$	Heading angle measurement
ε_2	Gaussian heading angle noise
δ_2^2	Noise variance of heading angle
N	Number of particles
Z	Measurement
w	Particle weight
φ	Gaussian particle initialization noise
δ_0^2	Noise variance of particle initialization
$q\{\cdot\}$	importance probability density
$\delta(\cdot)$	Dirac function
$pBest_i^t$	current personal best value
$gBest^t$	current global best value
ω	inertia factor
c_1 and c_2	learning factor
Q	Quantum particle swarm
np	Size of quantum particle swarm
Q_i	Quantum particle
D_i	Binary particle
$IterMax$	maximum number of iterations of QPSO
s	Confidence scale of the error ellipse
α	Confidence probability of error ellipse
λ	Feature vector
ϕ	Tilt angle of the error ellipse
(x_p, y_p)	Estimated center position

and facilitated the derivation of the formula. We will discuss the mixed noise problem further in the following work. The step size measurement of the target node could be simplified as:

$$\hat{d}_k = d_k + \varepsilon_{1,k},\ \varepsilon_{1,k} \sim \mathcal{N}\left(0, \delta_{1,k}^2\right) \tag{3}$$

where d_k represents the actual step size of node $\mathcal{T}$ moving from time t_k to time t_{k+1}, namely

$$d_k = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \tag{4}$$

and $\varepsilon_{1,k}$ is Gaussian step size noise with mean of 0 and variance of $\delta_{1,k}^2$. Then, the vector $\hat{d} = \left[\hat{d}_0, \hat{d}_1, \cdots, \hat{d}_{K-1}\right]^T$ reflects the step size measurement information. The heading angle measurement is denoted as:

$$\hat{\theta}_k = \theta_k + \varepsilon_{2,k},\ \varepsilon_{2,k} \sim \mathcal{N}\left(0, \delta_{2,k}^2\right) \tag{5}$$

where θ_k is the actual heading angle, namely

$$\theta_k = \arctan \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \tag{6}$$

and $\varepsilon_{2,k}$ is Gaussian heading angle noise with mean of 0 and variance of $\delta_{2,k}^2$. Then, the vector $\hat{\theta} = \left[\hat{\theta}_0, \hat{\theta}_1, \cdots, \hat{\theta}_{K-1}\right]^T$ reflects the angle measurement information.

B. Review of Particle Filter

PF provides a strategy for recursively generating the posterior probability density function, that is, employing a set of particles to speak to the posterior probability distribution of the state, which meets the thought of Monte Carlo sampling [6]. The standard handle of PF contains initialization, prediction, and update.

Initialization: By and large, PF completes the estimation of state X_k by updating a set of random measurements $\left\{X_k^{(i)}, w_k^{(i)}\right\}_{i=1}^N$ which comprises of N particles $X_k^{(i)}$ and corresponding weight $w_k^{(i)}$ at time t_k. When the number of particles is adequate, the state estimatation approximates the posterior probability density $p\left(X_{1:k}|Z_{1:k}\right)$ of the obscure distribution $X_{1:k}$. At this initialization stage, it is vital to generate a series of particles on account of prior knowledge, that is, $X_0^{(1:N)} \sim p\left(X_0\right)$. Considering the motion state of a single target in a two-dimensional (2-D) coordinate system, the particles at the initial moment could be generated from the initial position information and random noise, i.e.,

$$\begin{bmatrix} \hat{x}_0^i \\ \hat{y}_0^i \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \varphi \tag{7}$$

where $[x_0, y_0]^T$ is the initial position of the target, and $\varphi = \left[\varphi_x, \varphi_y\right]^T$ is Gaussian noise in the horizontal and vertical axes, that is, $\varphi_x, \varphi_y \sim \mathcal{N}\left(0, {\delta_0}^2\right)$.

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

Update: At 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 doled out as $1/N$, and the following equations updates it:

$$w_k^{(i)} \sim w_{k-1}^{(i)} \frac{p\left(Z_k|X_k^{(i)}\right) p\left(X_k^{(i)}|X_{k-1}^{(i)}\right)}{q\left(X_k^{(i)}|X_{k-1}^{(i)}, Z_k\right)} \tag{8}$$

where $q(\cdot)$ denotes the importance probability density, and it is generally represented as the transition prior probability density function, that is, $q\left(X_k^{(i)}|X_{k-1}^{(i)}, Z_k\right) = p\left(X_k^{(i)}|X_{k-1}^{(i)}\right)$ [24]. 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 communicated as:

$$p\left(X_k|Z_{1:k}\right) \approx \sum_{i=1}^{N} \tilde{w}_k^{(i)} \delta\left(X_k - X_k^{(i)}\right) \tag{9}$$

where $w_k^{(i)}$ 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.

Because the proposal function of the generic PF is sub-optimal, there are two severe problems in the traditional PF method:

Problem 1 (Particle Impoverishment): This might happen when the likelihood is very narrow or the likelihood lies in the tail of prior distribution - the overlap region of likelihood and prior distribution is very small. So, after the update process, only a few particles will have significant importance weights. Thus, the sample set only contains few dissimilar particles, and sometimes they will drop to a single sample after several iterations. As a result, essential samples (reasonable hypothesis) may be lost.

Problem 2 (Sample Size Dependency): If the sample set size is small, there might not have particles distributed around the actual state. So after several iterations, it is tough for particles to converge to the actual state. So it is challenging to meet the particular need of generic PF with a small sample set size. Nevertheless, there is no guarantee to meet the real-time requirement of the system due to the decreasing tremendously computation efficiency when we provide a large enough sample set covering whole state space to ensure a precise estimation.

To the best of our knowledge, many resampling methods are utilized to solve particle impoverishment, which might cause the loss of particle diversity. Furthermore, there is also no strategy to overcome the sample size Dependency now. We proposed a novel particle filter algorithm called error constraint enhanced particle filter using quantum particle swarm optimization (EC-QPF) to solve these problems, detailed in the next section.

III. Methodologies

This section introduces a new quantum particle swarm optimization (QPSO) based particle filter method enhanced by error constraint. As the breakpoint to keep the particle diversity, we adopted the QPSO-based resampling method rather than the weight-based resampling method into the update stage of the particle filter. Moreover, it could also deal with the sample size dependency problem of PF. We also applied the method of error constraint for improving the accuracy of the proposed algorithm. First, we will briefly introduce how quantum particle swarm optimization works; then, we put forward the theoretical foundation of error constraint; lastly, we will present our EC-QPF detailly.

A. Quantum Particle Swarm Optimization (QPSO)

Particle swarm optimization (PSO) is an evolutionary computation, which comes from studying the behavior of flocks of birds. By designing a set of particles to simulate a flock of birds, PSO aims to search for the global optimum through mimicking the behavior of particles, where each particle represents a candidate solution has only two properties: position x_i^t and velocity v_i^t, $i = 1, 2, \ldots, N$. N represents the number of particles, and t represents the t-th iteration of the algorithm. Each particle searches for the optimal solution separately in the search space, recorded as the current personal best value $pBest_i^t$. The $pBest_i^t$ is shared with other particles in the whole particle swarm to find the optimal personal best value $gBest^t$ as the current global optimal solution. To reach the final global optimal solution, the particles move iteratively in the search space, and the position x_i^t and velocity v_i^t of particle i at t-th iteration are updated by the following formulas:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 \left(pBest_i^t - x_i^t\right) + c_2 r_2 \left(gBest^t - x_i^t\right) \tag{10}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{11}$$

where $\omega \in [0, 1]$ is the inertia factor, c_1 and c_2 are two positive constants named learning factor, r_1 and r_2 are two random numbers in $[0, 1]$.

Since PSO is only applicable to optimization of continuous nonlinear functions, there are a number of PSO variants proposed to solve binary optimization. And quantum particle swarm optimization (QPSO) is a typical algorithm. In QPSO, a swarm $Q = \{Q_1, Q_2, \ldots, Q_m\}$ of m quantum particles is maintained and evolves, where each quantum particle Q_i is a n-dimensional real-valued vector $\left(q_1^i, q_2^i, \ldots, q_n^i\right)$ with $q_j^i \in [0, 1]$. For each component $q_j^i (1 \le j \le n)$ of quantum particle Q_i, its value represents the probability that the associated binary decision variable x_j takes the value of 0.

Starting with a randomly initialized Q^t in which the notation t denotes the current number of iterations, the algorithm first transforms each quantum particle $Q_i = \left(q_1^i, q_2^i, \ldots, q_n^i\right)$ of Q^t into a binary particle $D_i = \left(d_1^i, d_2^i, \ldots, d_n^i\right)$ by applying a random observation:

$$d_j^i = \begin{cases} 1, & \text{if } q_j^i < \text{rand}(0, 1) \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where rand(0, 1) denotes a random real number in $[0, 1]$. Then the quantum particle swarm updates by the following evolution formulas at each iteration t:

$$\bar{Q}^{t+1} = \alpha \bar{D}^t + (1 - \alpha)\left(\vec{e} - \bar{D}^t\right) \tag{13}$$

$$\hat{Q}_i^{t+1} = \alpha \hat{D}_i^t + (1 - \alpha)\left(\vec{e} - \hat{D}_i^t\right) \tag{14}$$

$$Q_i^{t+1} = c_1 Q_i^t + c_2 \hat{Q}_i^{t+1} + (1 - c_1 - c_2)\, \bar{Q}^{t+1} \tag{15}$$

where $\vec{e} = (1, 1, \ldots, 1)$ is a n-dimensional unit vector, $\hat{D}_i^t$ $(1 \le i \le m)$ and $\bar{D}^t$ denote respectively the personal and global best solution for the binary particle D_i at iteration t, and $\hat{Q}_i^{t+1}$ $(1 \le i \le m)$ and $\bar{Q}^{t+1}$ represent respectively the personal and global historical best solution for the quantum particle Q_i at iteration $t + 1$. In these equations, α is called control parameter satisfying $\alpha \in [0, 1]$. In addition, c_1 and c_2 satisfy $c_1 \in [0, 1]$, $c_2 \in [0, 1]$, and $0 < c_1 + c_2 < 1$.

The two latter coefficients represent the degree of belief in oneself and the personal best, respectively, while $(1 - c_1 - c_2)$ represents the degree of belief in the global best quantum particle. After a new quantum particle Q_i^{t+1} is generated by equations (13)-(15), Q_i^{t+1} is used to replace Q_i^t and is at the same time transformed into a binary particle d, which is then used to update $\bar{D}_i^{t+1}$ and $\hat{D}_i^{t+1}$ accordingly.

B. Error Constraint

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, the exact $(1-\beta)\%$-confidence interval for the scale s is defined as the interval $\langle s_1, s_2 \rangle$ [25]. s_1 and s_2 are the lower and upper confidence limits, respectively. Such that the following holds true

$$\Pr\left(s_1 < s < s_2\right) = 1 - \beta \tag{16}$$

The probability β represents the significance level, and $1 - \beta$ represents the confidence level. We use the symbol α to express the confidence probability, i.e., α=$1 - \beta$. In two-dimensional or three-dimensional (2-D/3-D) parameter estimation (such as target tracking), this confidence interval could be exploited as an error constraint.

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

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

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

$$\operatorname{cov}(x,y) = E\left[(x - E(x))(y - E(y))\right] \tag{18}$$

If x is positively related to y, then y and x are also positively correlated, i.e., $\operatorname{cov}(x,y) = \operatorname{cov}(y,x)$. Therefore, the covariance matrix is always a symmetric matrix. Then, equation of the constraint represented as an elliptical range could be expressed as:

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

where λ_1 and λ_2 are the maximum and minimum eigenvector corresponding to the covariance matrix, respectively. (x_p, y_p) is the estimated center position, and s is the scale of the ellipse. When the 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:

$$\phi = \arctan\left(\frac{\lambda_1(y)}{\lambda_1(x)}\right) \tag{20}$$

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

$$\frac{\left((x' - x_p)\cos\phi + (y' - y_p)\sin\phi\right)^2}{\lambda_1} + \frac{\left(-(x' - x_p)\sin\phi + (y' - y_p)\cos\phi\right)^2}{\lambda_2} \le s \tag{21}$$

shown as Fig. 1.

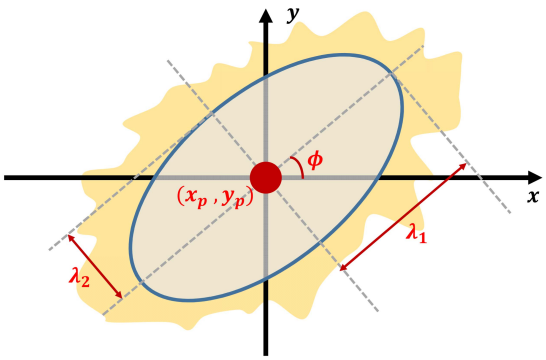


Fig. 1. The mathematical model of error constraint.

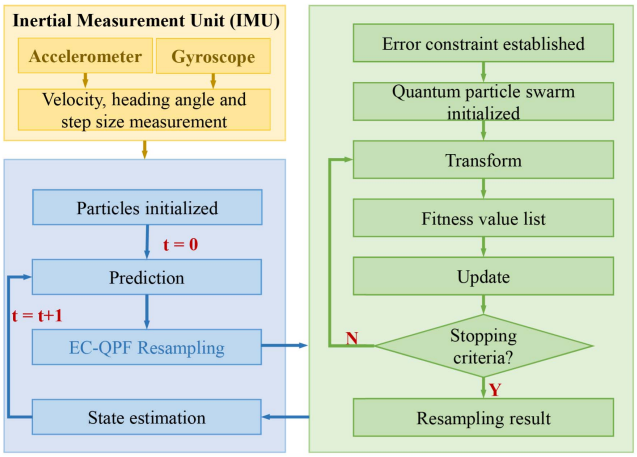


Fig. 2. The flow chart of EC-QPF.

C. EC-QPF

As shown in Fig. 2, how our proposed algorithm EC-QPF works are represented detailly. Considering a target node $\mathcal{T}$ equipped with an inertial measurement unit (IMU) which includes an accelerometer and gyroscope, we could obtain the velocity, heading angle, and step size of $\mathcal{T}$ as the input parameters of EC-QPF. Just like traditional PF, we generate a series of particles according to prior knowledge at time $t = 0$, i.e., $X_{1:N}^0 \sim p\left(X^0\right)$, where N is number of particles. When node $\mathcal{T}$ moves in a two-dimensional (2-D) coordinate system, particles move according to measuring velocity and heading angle. So we could calculate the prior probability and obtain an initial estimation $\hat{X}_{ini}^0$. Unlike the traditional PF algorithm, we do not have to calculate the weight of particles according to which we execute the resampling process. We deduce the error constraint whose center is $\hat{X}_{ini}^0$ by the distribution of particles and pick a set of particles $X_{1:M}^0$ within the constraint, where $M \le N$ generally. Then, as input of quantum particle swarm optimization resampling process, particles $X_{1:M}^0$ will be calculated to get final estimation $\hat{X}_{fin}^0$ at time $t = 0$.

Next, considering particles $X_{1:M}^0$, we will execute the resampling process using the quantum particle swarm optimization method, which is mentioned in Section 3.1. For resampling a new set of particles from $X_{1:M}^0$, that is, an optimal choice of particles to estimate the state of the target, we transform

the resampling problem into a problem to finding an optimal solution. First, we initialize a quantum particle swarm $Q = \{Q_1, Q_2, \ldots, Q_{np}\}$, where np denotes the number of quantum particles. Each quantum particle Q_i is a M-dimension real-valued vector $(q_1^i, q_2^i, \ldots, q_M^i)$ with $q_j^i \in [0, 1]$, where M is the same of quantity of $X_{1:M}^0$. For each component $q_j^i (1 \leq j \leq n)$ of quantum particle Q_i, its value represents the probability that the corresponding particle of $X_{1:M}^0$ is selected.

Algorithm 1 EC-QPF

Input: : The velocity v^t, heading angle θ^t and step size d^t of target node $\mathcal{T}$ measured at time t; Probability β of error constraint; Size of particle swarm (np), maximum number of iterations ($IterMax$), α, c_1 and c_2 of QPSO;

Output: : The estimation $\hat{X}_{fin}^t$ of state of target node $\mathcal{T}$.

1. INITIALIZATION AND PREDICTION

if $t = 0$ **then**

Initialize particles $X_{1:N}^t$ of particle filter;

else

Obtain $X_{1:N}^t$ after resampling at time $t-1$, i.e., $\bar{X}_{1:N}^{t-1}$;

end if

Predict the state $\hat{X}_{ini}^t$ of $\mathcal{T}$ with $X_{1:N}^t$;

2. EC-QPF

2.1 Select particles within error constraint

View $\hat{X}_{ini}^t$ as the estimated center position of error constraint;

Calculate the covariance matrix of $X_{1:N}^t$;

Calculate range of error constraint using Eqs. (19) - (21);

Select a set of particles $X_{1:M}^t$ within error constraint;

2.2 Quantum particle swarm optimization resampling

let $k = 0$, k denotes the quantum particle swarm at the iteration k;

Initialization of quantum particle swarm $Q^k = \{Q_1^k, Q_2^k, \ldots, Q_{np}^k\}$, where $Q_i^k = (q_1^i, q_2^i, \ldots, q_M^i)$;

while $k < IterMax$ **do**

for $i = 1$ to np **do**

Transform Q_i^k into binary particle D_i^k with Eq. (12);

Calculate the fitness value of D_i^k with Eq. (22);

end for

Update quantum particle swarm using Eqs. (23) - (25);

$k = k + 1$;

end while

Obtain the optimal solution D^*, according to which we can resampling a new set of particlea new set of particle;

Duplication to hold the quantity of particles $\bar{X}_{1:N}^t$;

3. ESTIMATION

Calculate the estimation $\hat{X}_{fin}^t$ of state of target node $\mathcal{T}$;

As described in Section 3.1, QPSO algorithm performs a number of evolution iterations until the stopping criteria of QPSO is met, i.e., until the maximum number of iterations of QPSO $IterMax$ is reached or the global best value remained unchanged after half of $IterMax$ iterations. For a randomly initialized Q^k in which the notation k denotes the current number of iterations, we transform each quantum particle $Q_i = (q_1^i, q_2^i, \ldots, q_M^i)$ into a M-dimensional binary vector (called binary particle) $D_i = (d_1^i, d_2^i, \ldots, d_M^i)$ by applying a random observation as Eqs. 12. For each component $d_j^i (1 \leq j \leq M)$, its value represents if it is selected, i.e., value "1" represents be selected and value "0" represents not. As we can see, each binary particle D_i expresses a resampling result of $X_{1:M}^0$. And in this article, we define the fitness function for each binary particle D_i as:

$$\text{Fitness}_{\ i} = \exp\left[-\frac{\sum_{j=0}^{M} d_j^i Dis_j}{2\mathbf{V}\sum_{j=0}^{M} d_j^i}\right] \tag{22}$$

where Dis_j is the distance between $\hat{X}_{ini}^0$ and X_j^0, i.e., $Dis_j = \left\|\hat{X}_{ini}^0 - X_j^0\right\|_2$, and $\mathbf{V}$ is the variance of $X_{1:M}^0$. The higher the fitness value, the more reliable the sampling results are. After getting each binary particle's fitness value, we could update the quantum particle swarm according to the following evolution formulas at each iteration k:

$$\bar{Q}^{k+1} = \alpha \bar{D}^k + (1-\alpha)\left(\vec{e} - \bar{D}^k\right) \tag{23}$$

$$\hat{Q}_i^{k+1} = \alpha \hat{D}_i^k + (1-\alpha)\left(\vec{e} - \hat{D}_i^k\right) \tag{24}$$

$$Q_i^{k+1} = c_1 Q_i^k + c_2 \hat{Q}_i^{k+1} + (1 - c_1 - c_2)\,\bar{Q}^{k+1} \tag{25}$$

As mentioned in Section 3.1, control parameter $\alpha \in [0, 1]$, coefficients $c_1 \in [0, 1]$, $c_2 \in [0, 1]$, and $0 < c_1 + c_2 < 1$. We performed some preliminary experiments to find the "good" values for the QPSO parameters. We noted that our algorithm displayed better performance for $\alpha > 0.7$. low values implied slow convergence and high values implied convergence to non-optimal performance values. Thus, we chose $\alpha = 0.8$. We also observed that the weight of best global position factor was more important than c_1 and c_2. Low values for $(1 - c_1 - c_2)$ implied slow convergence and high values result in a lack of diversity. A good trade-off was obtained for the values $c_1 = c_2 = 0.2$, and $(1 - c_1 - c_2) = 0.6$ [26]. $\vec{e} = (1, 1, \ldots, 1)$ is a M-dimensional unit vector, $\hat{D}_i^k$ $(1 \leq i \leq np)$ and $\bar{D}^k$ denote respectively the personal and global best solution for the binary particle D_i at iteration k, which could be inferred by fitnes value set of binary particles D^k. And $\hat{Q}_i^{k+1}$ $(1 \leq i \leq np)$ and $\bar{Q}^{k+1}$ represent respectively the personal and global historical best solution for the quantum particle Q_i at iteration $k+1$. After a new quantum particle Q_i^{k+1} is generated by equations (23)-(25), Q_i^{k+1} is used to replace Q_i^k and is at the same time transformed into a binary particle D_i^{k+1}, which is then used to update $\bar{D}_i^{k+1}$ and $\hat{D}_i^{k+1}$ accordingly.

Until a maximum number of iterations is reached, we obtain a final optimal solution D^*, which illustrates an optimal resampling result. After all particles selected corresponding to D^* have performed the duplication operation, the quantity of particles N still holds. The chosen set of particles is utilized to get the final estimation $\hat{X}_{fin}^0$ and would also be used as initial input for the next step.

To better illustrate the process of the proposed EC-QPF algorithm, pseudo-code is displayed in Algorithm 1. In the proposed algorithm, the most critical parameters that affect the computation time are the number of particles N, the size

TABLE II
PRIMARY EXPERIMENTAL PARAMETER SETTINGS

Paremeter	Numerical value
Sampling interval t_s	$1(s)$
Step noise variance $\delta_{1,t}^2$	$0.01(m^2)$
Angle noise variance $\delta_{2,t}^2$	$0.01(rad^2)$
Number of PF particles N	500
Size of particle swarm np of QPSO	500

of particle swarm np, and the maximum number of iterations $IterMax$. The algorithm's complexity is at most $O(N * IterMax * np)$ according to the Big O notation.

IV. NUMERICAL SIMULATION AND ANALYSIS

In this section, we consider various simulation scenarios to verify our proposed EC-QPF Algorithm. Numerical simulations confirm the efficiency of the scheme put forth.

A. Experimental Setup

We conducted a random walk numerical simulation experiment of the target node based on MATLAB. The personal computer's operating system is Windows 10 Professional Edition system, and the hardware configuration is four core i5 CPU and 16GB memory. The experimental scene is set as $200m * 200m$, and the initial position of the target node is random. The target node travels 180 steps at each simulation scenario, and the travel trajectories are recorded. The primary parameter setting of the experiment is shown in Table II. All algorithms are performed and predict the position reached by each travel of the target node. Euclidean distances between the predicted position and the actual position are counted.

B. Case 1: Experiment for Verifying Accuracy

To demonstrate the effectiveness and advantages of our proposed EC-QPF algorithm in target tracking, we carried out a single target tracking simulation experiment, in which EC-QPF was compared with two other modified PF algorithms, Compressed Monte Carlo Resampling PF (CMCR-PF) [27] and standard Systematic Resampling PF (SR-PF) [28]. The error is expressed as the Euclidean distance between the predicted position and the actual position, which is defined as follows:

$$e = \left\| \hat{P}_t - P_t \right\|_2 \tag{26}$$

As can be seen from Fig. 3, the red represents the result of EC-QPF, while the blue and the orange represent the result of CMCR-PF and SR-PF, respectively. We can draw the following conclusions from Fig. 3-(a): The estimation 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 drifting problem of IMU. Considering the three PF algorithms, the error constraint enhanced quantum particle swarm optimization-based particle filter (EC-QPF) has a lower error growth rate than the other two modified PF algorithms, proving that the EC-QPF algorithm could suppress

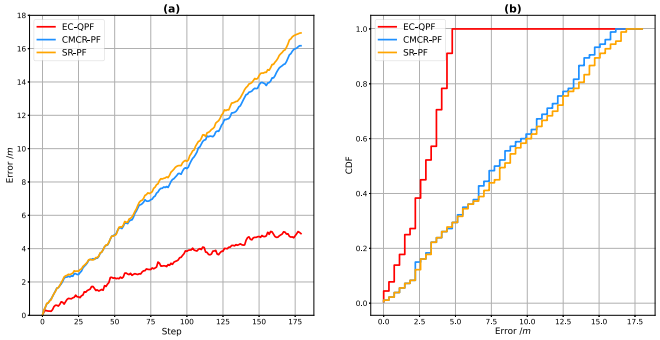


Fig. 3. (a) The estimation error of three PF algorithms; (b) CDF of estimation error.

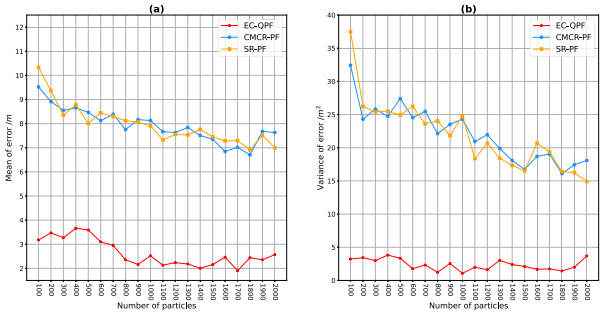


Fig. 4. Experiment for PF particle number. (a) The mean of estimation error; (b) The variance of estimation error.

the cumulative effect error to a certain extent. Furthermore, the maximum error of EC-QPF in 180 steps is 5.03 m, while CMCR-PF is 16.17 m and SR-PF is 16.94 m, which explains the superiority of our proposed EC-QPF algorithms.

Fig. 3-(b) shows the cumulative distribution function (CDF) versus estimation error for three PF algorithms in target tracking for a total of 180 steps. It is observed that using the proposed EC-QPF, we can attain 4.36 m error at 80% confidence level, where CMCR-PF is 13.25 m and SR-PF is 13.56 m. The proposed EC-QPF has a 67% improvement instead of CMCR-PF and SR-PF, which provides more estimation accuracy than the other two modified PF methods. The reason is that the QPSO-resampling method avoids particle impoverishment and keeps the diversity of particles. Furthermore, error constraint ensures particles in a range with high confidence, which also enhances the performance of EC-QPF.

C. Case 2: Effect of the Number of Particles

As well known, the accuracy of the traditional PF algorithm depends on the sample size. The larger the sample size, the more accurate the PF algorithm. However, there is no guarantee to meet the real-time requirement of the system because of the decreasing tremendously computation efficiency when setting a large sample size. We solve this problem by applying quantum particle swarm optimization to the resampling process of the particle filter. For verifying the validity, we take the number of PF particles as a variable, from 100 to 2000, and keep other parameters unchanged in Table II to execute a simulation.

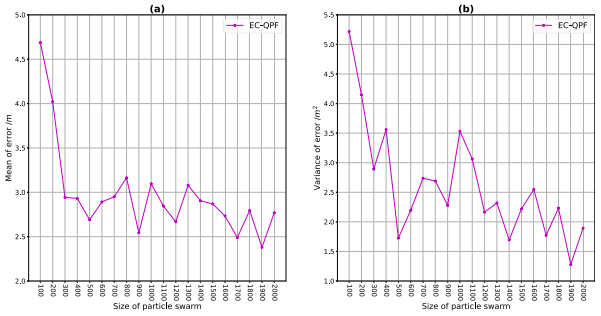


Fig. 5. Experiment for QPSO swarm size. (a) The mean of estimation error; (b) The variance of estimation error.

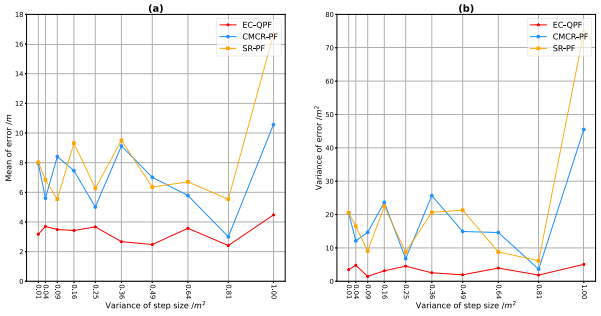


Fig. 6. Experiment for noise of step size. (a) The mean of estimation error; (b) The variance of estimation error.

As shown in Fig. 4-(a), with considering the three PF algorithms mentioned above, the mean of the estimation error of the target node shows a gradually decreasing trend as the PF particle number increases. The mean estimation error of EC-QPF tends to be stable when the number of particles is greater than 800, while the other two modified PFs continue to decline. This proves that our algorithm EC-QPF is less dependent on particle number. Furthermore, the maximum mean of EC-QPF 3.66 m is much less than the minimum of the other two (CMCR-PF is 6.71 m, and SR-PF is 6.92 m), which demonstrates the advantage of the proposed EC-QPF. As can be seen from Fig. 4-(b), the variance of error of EC-QPF keeps smooth and steady, while CMCR-PF and SR-PF present a clear downward trend. Moreover, in CMCR-PF, the maximum and minimum variance of estimation error are 32.43 m^2 and 16.11 m^2 respectively, nearly to SR-PF. They are much larger than the maximum of EC-QPF 3.82 m^2, which illustrates that EC-QPF has better stability than the other modified PF algorithms. Thus, in terms of the proposed algorithm EC-QPF, we set 500 as the parameter of the number of particles, balancing accuracy and computation cost.

D. Case 3: Effect of the Size of Particle Swarm

For finding an appropriate parameter of QPSO swarm size np, we took the size of QPSO particle swarm as a variable, from 100 to 2000, and keep other parameters unchanged in Table II to execute a simulation. Fig. 5 shows the trend of the mean and variance of the estimation error as the swarm size increasing. The variation trend of estimation error varies when the size of the particle swarm is smaller or larger than 500. The mean and variance of estimation error decrease sharply

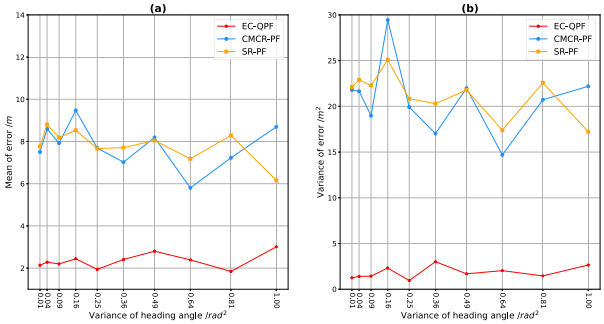


Fig. 7. Experiment for noise of heading angle. (a) The mean of estimation error; (b) The variance of estimation error.

when swarm size is smaller than 500. The maximum mean of error is 4.69 m, and the value reduces to 2.69 m when the size of particle swarm is 500. In addition, the maximum variance of the error is 5.22 m^2, and the value reduces to 1.73 m^2 when the size of the particle swarm is 500. There is a slight fluctuation when swarm size is larger than 500. From 500 to 2000, the maximum and minimum mean of error is 3.16 m and 2.38 m, while the error's maximum and a minimum of variance is 3.53 m^2 and 1.28 m^2. Thus, in terms of the proposed algorithm EC-QPF, we set 500 as the parameter of swarm size of quantum particle swarm optimization, balancing accuracy and computation cost.

E. Case 4: Effect of the Step Size and Heading Angle

Furthermore, due to the uncertainty of the target motion, we took the values of the noise of step size and heading angle as variables respectively and keep other parameters unchanged in Table II to execute simulations. Fig. 6 and Fig. 7 show the trend of the mean and variance of the estimation error as the variance of step size and heading angle increasing, respectively. As can be seen from Fig. 6 and Fig. 7, the mean and variance of the estimation error of the proposed EC-QPF keep a smooth trend while the variance of step size and heading angle are increasing. However, CMCR-PF and SR-PF show significant fluctuation, which means high stability and superiority of the proposed algorithm. Since our proposed algorithm is almost unaffected by the step size and heading angle, we set 0.01 as the parameter of step noise variance and angle noise variance without loss of generality.

V. Conclusion

In this paper, we proposed a new modified particle filter method named error constraint enhanced particle filter using quantum particle swarm optimization algorithm (EC-QPF), which achieves high-precision position estimation of target tracking with IMU. We avoided particle impoverishment problems occurring in the traditional particle filter method by replacing the weight-based resampling method with the quantum particle swarm optimization-based resampling method. Then we raised the theoretical foundation of error constraint according to the error distribution characteristics of particles applying to enhance the performance of the proposed particle filter. Experiment shows that the estimation of the target position of the proposed algorithm EC-QPF

is more accurate than the other two modified particle filter methods under the same condition, which proves that EC-QPF eliminates the cumulative error more efficiently. Furthermore, we discussed the impact of the number of particles, the size of the quantum particle swarm, the step size, and the heading angle. Experimental results verify our proposed algorithm's superiority and stability. In this paper, we assumed the noise to be Gaussian distributed noise, which simplified the issue and facilitated the derivation of the formula. We will discuss the mixed noise problem further in the following work.

REFERENCES

[1] Y. Yuan, Y. Lu, and Q. Wang, "Tracking as a whole: Multi-target tracking by modeling group behavior with sequential detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3339–3349, Dec. 2017, doi: 10.1109/tits.2017.2686871.

[2] 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.

[3] H. Chen, B. Guo, Z. Yu, and Q. Han, "CrowdTracking: Real-time vehicle tracking through mobile crowdsensing," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7570–7583, Oct. 2019.

[4] B. Jang and H. Kim, "Indoor positioning technologies without offline fingerprinting map: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 508–525, 1st Quart., 2019, doi: 10.1109/COMST.2018.2867935.

[5] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.

[6] J. Shi, G. Wang, and L. Jin, "Least squared relative error estimator for RSS based localization with unknown transmit power," *IEEE Signal Process. Lett.*, vol. 27, pp. 1165–1169, Jul. 2020, doi: 10.1109/LSP.2020.3005298.

[7] J. Shi, G. Wang, and L. Jin, "Moving source localization using TOA and FOA measurements with imperfect synchronization," *Signal Process.*, vol. 186, Sep. 2021, Art. no. 108113.

[8] W. Xiong, C. Schindelhauer, H. C. So, J. Bordoy, A. Gabbrielli, and J. Liang, "TDOA-based localization with NLOS mitigation via robust model transformation and neurodynamic optimization," *Signal Process.*, vol. 178, Jan. 2021, Art. no. 107774.

[9] N. H. Nguyen, K. Dogancay, and E. E. Kuruoglu, "An iteratively reweighted instrumental-variable estimator for robust 3-D AOA localization in impulsive noise," *IEEE Trans. Signal Process.*, vol. 67, no. 18, pp. 4795–4808, Sep. 2019.

[10] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. De Munari, "IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8553–8562, Oct. 2019, doi: 10.1109/JIOT.2019.2920283.

[11] S. Duan, H. Wu, C. Xu, and J. Wan, "Toward swarm robots tracking: A constrained Gaussian condensation filter method," in *Proc. 12th Int. Conf. Swarm Intell. (ICSI)*, Qingdao, China, Jul. 2021, pp. 129–136.

[12] 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.

[13] N. Vagle, A. Broumandan, and G. Lachapelle, "Multiantenna GNSS and inertial sensors/odometer coupling for robust vehicular navigation," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4816–4828, Dec. 2018.

[14] M. Jonasson, A. Rogenfelt, C. Lanfelt, J. Fredriksson, and M. Hassel, "Inertial navigation and position uncertainty during a blind safe stop of an autonomous vehicle," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 4788–4802, May 2020.

[15] N. Enayati, E. De Momi, and G. Ferrigno, "A quaternion-based unscented Kalman filter for robust optical/inertial motion tracking in computer-assisted surgery," *IEEE Trans. Instrum. Meas.*, vol. 64, pp. 2291–2301, 2015.

[16] H. Liu, Q. Li, C. Li, and H. Zhao, "Application research of an array distributed IMU optimization processing method in personal positioning in large span blind environment," *IEEE Access*, vol. 8, pp. 48163–48176, 2020, doi: 10.1109/ACCESS.2020.2979484.

[17] M. Ghobadi, P. Singla, and E. T. Esfahani, "Robust attitude estimation from uncertain observations of inertial sensors using covariance inflated multiplicative extended Kalman filter," *IEEE Trans. Instrum. Meas.*, vol. 67, pp. 209–217, 2018.

[18] S. Lee and J. McBride, "Extended object tracking via positive and negative information fusion," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1812–1823, Apr. 2019.

[19] X. Fu and Y. Jia, "An improvement on resampling algorithm of particle filters," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5414–5420, Oct. 2010.

[20] T. Li, M. Bolic, and P. M. Djuric, "Resampling methods for particle filtering: Classification, implementation, and strategies," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015, doi: 10.1109/MSP.2014.2330626.

[21] J. Ala-Luhtala, N. Whiteley, K. Heine, and R. Piche, "An introduction to twisted particle filters and parameter estimation in non-linear state-space models," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4875–4890, Sep. 2016, doi: 10.1109/TSP.2016.2563387.

[22] S. Zhao, C. K. Ahn, P. Shi, Y. S. Shmaliy, and F. Liu, "Bayesian state estimation for Markovian jump systems: Employing recursive steps and pseudocodes," *IEEE Syst., Man, Cybern. Mag.*, vol. 5, no. 2, pp. 27–36, Apr. 2019.

[23] 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.

[24] H. Bi, J. Ma, and F. Wang, "An improved particle filter algorithm based on ensemble Kalman filter and Markov chain Monte Carlo method," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 2, pp. 447–459, Feb. 2015.

[25] V. Witkovský and G. Wimmer, "Exact confidence intervals for parameters in linear models with parameter constraints," in *Proc. 13th Int. Conf. Meas.*, 2021, pp. 22–25, doi: 10.23919/Measurement52780.2021.9446783.

[26] B. Haddar, M. Khemakhem, S. Hanafi, and C. Wilbaut, "A hybrid quantum particle swarm optimization for the multidimensional Knapsack problem," *Eng. Appl. Artif. Intell.*, vol. 55, pp. 1–13, Oct. 2016.

[27] L. Martino and V. Elvira, "Compressed Monte Carlo with application in particle filtering," *Inf. Sci.*, vol. 553, pp. 331–352, Apr. 2021, doi: 10.1016/j.ins.2020.10.022.

[28] Q. Tian, K. I.-K. Wang, and Z. Salcic, "An INS and UWB fusion approach with adaptive ranging error mitigation for pedestrian tracking," *IEEE Sensors J.*, vol. 20, no. 8, pp. 4372–4381, Apr. 2020, doi: 10.1109/JSEN.2020.2964287.

Jiawang Wan received the B.E. degree from the University of Science and Technology Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interest includes the Internet of Things.

Cheng Xu (Member, IEEE) received the B.E., M.S., and Ph.D. degrees from the University of Science and Technology Beijing (USTB), China, in 2012, 2015, and 2019, respectively. He is currently working with the Data and Cyber-Physical System Laboratory (DCPS), University of Science and Technology Beijing. His research interest includes the Internet of Things.

Yidan Qiao received the B.E. degree from the University of Science and Technology Beijing, China, in 2021. Her research interests include particle filter and particle swarm optimization.

Xiaotong Zhang (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the University of Science and Technology Beijing, in 1997 and 2000, respectively. He was a Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing. His research interests include work in wireless sensor networks and computer architecture.