# Multitarget Cooperative Motion Tracking Based on Quantum Belief Propagation

Jiawang Wan, Cheng Xu, *Member, IEEE*, Yuchen Shi, Weizhao Chen, Fangwen Ye, Ran Wang, *Graduate Student Member, IEEE*, and Xiaotong Zhang, *Senior Member, IEEE*

***Abstract*—In this article, we introduce a novel cooperative target tracking algorithm, namely, the quantum-inspired belief propagation (BP), aimed at rectifying the limitations observed in existing localization algorithms employed in multitarget cooperative tracking scenarios. Leveraging the principles of quantum superposition, our algorithm seeks to alleviate the uncertainty inherent in message fusion within BP frameworks, thereby enhancing the accuracy and stability of multitarget cooperative localization. The utilization of the quantum Monte Carlo method facilitates the simulation of the message distribution process, with quantum particles embodying the superposition of multiple states concurrently. This approach effectively addresses the intractable integrations encountered in message updating on factor graphs (FGs), rendering the algorithm agnostic to the number of particles involved. Moreover, quantum unitary transformations and quantum closed box operations are deployed to encode FG function nodes for the propagation of quantum messages. This innovation surmounts the challenge posed by traditional FG function nodes' inability to process quantum messages. Experimental findings corroborated the superiority of the proposed algorithm in terms of accuracy and robustness.**

***Index Terms*—Cooperative tracking, multitarget, quantum belief propagation (BP).**

## I. Introduction

SEARCH, detection, and tracking of targets play pivotal roles in numerous military and civil missions [1]. In recent decades, multitarget cooperative localization (MCL) has found widespread applications, such as autonomous driving [2], autonomous aerial vehicles formation [3], and robot distribution [4], [5], among others. Within the realm of MCL, several challenges must be addressed, including the acquisition of accurate measurements, effective fusion of information, and estimation of desired target states [6].

The field of multitarget state estimation and tracking has witnessed significant advancements primarily rooted in the principles of Bayes' theorem [7]. Target state estimation has been explored by Esmailifar et al. [1], Martinez et al. [8], and Zhan et al. [9] using the extended Kalman filter (EKF). Ivey and Johnson [10] conducted a comparative analysis of target positioning performance, evaluating the efficacy of both the EKF and the square root unscented Kalman filter (UKF). Nevertheless, these conventional approaches encounter challenges, such as signal interference and error accumulation, which contribute to estimation errors and biases. Wang et al. [11] proposed the uncertainty-constrained resampling-based particle filter (UCR-PF) method to address these issues. However, the particle filter sampling process in this method relies on the establishment of an error ellipse, which inevitably increases computational complexity. Moreover, the UCR-PF does not account for message fusion between multiple targets.

Belief propagation (BP) [12], also known as the sum-product algorithm, offers an efficient solution for addressing the multitarget state estimation problem. BP operates by propagating messages along the edges of a factor graph (FG) [13], which serves as the statistical model representation for the inference problem. Meyer et al. [14] proposed a BP method for tracking an unknown number of targets. However, its applicability is restricted due to the need for performing message passing on a specific FG. Wu et al. [15] proposed a time-of-arrival (TOA)-based passive multitarget localization method with inaccurate receivers based on BP. They introduced a sample-based method to represent messages by particles to address the intractable integrations in message updating on FG. However, the computational complexity depends on the particles' number. Sharma et al. [16] proposed decentralized Gaussian filters for cooperative self-localization based on BP. It only works when the message distribution is Gaussian. The adaptive kernel Kalman filter-based BP (AKKF-BP) algorithm proposed in [17] is capable of tracking an unknown and time-varying number of targets. However, the stability needs to be considered.

To address the aforementioned issues, this article proposes a quantum-inspired BP algorithm (CT-QBP) for the MCL problem. The key contributions of this article are summarized as follows.

1) We propose a quantum-inspired BP algorithm (CT-QBP) for the MCL problem. The algorithm leverages the quantum superposition property to mitigate the uncertainty associated with message fusion in BP, thereby enhancing the accuracy and stability of MCL.
2) We employ the quantum Monte Carlo idea to simulate the quantum message distribution. Quantum particles can simultaneously represent the superposition

Received 5 November 2024; revised 4 December 2024 and 30 December 2024; accepted 4 January 2025. Date of publication 7 January 2025; date of current version 9 May 2025. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62101029, and in part by the China Scholarship Council Award under Grant 202006465043 and Grant 202306460078. *(Corresponding author: Cheng Xu.)*

The authors are with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, and also with the Shunde Innovation School, University of Science and Technology Beijing, Foshan 528399, China (e-mail: WJW_2333@163.com; xucheng@ustb.edu.cn; shiyuchen199@sina.com; weichiu_chan@foxmail.com; yfwen2000@outlook.com; wangran423@foxmail.com; zxt@ies.ustb.edu.cn).

Digital Object Identifier 10.1109/JIOT.2025.3526812

of multiple states, thereby addressing the intractable integrations in message updating on FGs.

3) We utilize quantum unitary transformations to quantize FG function nodes for propagating quantum messages. We introduce the Grover node and the Measure node to expedite the search for an optimal quantum state and collapse it into a classical state efficiently.

The remainder of this article is organized as follows. Section II briefly introduces FGs, BP and quantum computation. Section III briefly introduces the model of the MCL problem. Section IV describes the specific process of the proposed quantum-inspired BP algorithm. Section V shows the results and discussion of numerical experiments. Section VI summarizes this article.

## II. Related Work

### A. Factor Graph and Belief Propagation

In the domain of probabilistic modeling, FGs serve as a prevalent graphical representation of joint probability distributions and variable relationships. These graphs consist of nodes representing random variables or variable collections, and edges denoting dependencies or factors [13].

FGs have broad applications in probability inference, learning, planning, and beyond. In machine learning, they are employed for classification, regression, clustering, and similar tasks [18], [19], [20]. FGs also play crucial roles in natural language processing tasks, such as part-of-speech tagging, named entity recognition, and syntactic parsing [21], [22]. Additionally, they are valuable in image processing tasks like image segmentation and restoration [23]. Moreover, FGs facilitate collaborative localization and navigation by establishing interaction models among multiple agents and frameworks for state estimation [24], [25], [26]. They represent each agent's observation and state as nodes, using edges to describe information sharing and interaction among agents, thereby enabling comprehensive consideration of uncertainties and dependencies in multiagent systems for accurate modeling and inference of positions and trajectories.

FGs simplify problems and enhance computational efficiency by explicitly capturing variable dependencies. The BP algorithm is a potent computational technique for precise probability inference on FGs [12]. Operating on the principle of probability propagation, this algorithm iteratively computes the probability distribution of each node. Its applications span diverse domains, including speech recognition [27], machine translation [28], [29], and image recognition [30], [31]. Moreover, BP aids in collaborative localization and navigation by fusing observation and state information from different agents to determine confidence levels on each agent's position and trajectory [32], [33]. Integrating BP with sensor fusion, Kalman filtering, and other techniques enhances the accuracy and robustness of position and trajectory inference.

In practical applications, FGs and BP algorithms require optimization and adaptation to specific scenarios, considering real-time performance, computational efficiency, and robustness in fields like robotic collaboration and autonomous driving. Thus, further research is necessary to optimize FGs and BP algorithms to meet practical application requirements.

### B. Quantum Computation and Quantum Algorithm

Quantum computing represents an advanced computational paradigm rooted in the principles of quantum mechanics, offering computational capabilities and advantages that surpass classical computing [34]. In quantum computation, qubit is the primary object of information carrier, as bit in classical computation. Think of classical bits as light switches that can be either "off" (0) or "on" (1). In contrast, qubits are like dimmer switches that can be in a state of "on," "off," or any level of brightness in between simultaneously. This ability to exist in multiple states at once is known as superposition. Superposition allows a qubit to perform multiple calculations at once. Imagine flipping a coin; while it is in the air, it is not just heads or tails, but both possibilities at the same time. Only when it lands does it reveal a definite outcome. Similarly, qubits can represent multiple outcomes simultaneously until measured. A qubit can be denoted as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha$ and $\beta$ are called amplitudes and satisfy the normalized equation: $|\alpha|^2 + |\beta|^2 = 1$ [35]. Once a qubit is measured, it will collapse into certain eigenstates. In other words, under the probability of $|\alpha|^2$ or $|\beta|^2$ $|\psi\rangle$ collapses into $|0\rangle$ or $|1\rangle$ after being measured. Further, the quantum mechanics introduces an essential matrix named unitary transformation matrix $U$. The matrix $U$ is utilized to transform the quantum state $|\psi\rangle$ to different quantum state $|\psi'\rangle$ as $|\psi'\rangle = U|\psi\rangle$. $U$ is required satisfying $U^\dagger U = UU^\dagger \equiv I$, where $I$ represents identity matrix. Quantum logic gates can be interpreted as matrices that modify the state of qubits. This is similar to those of classical gates, but one key difference between quantum logic gates and classical ones is that quantum gates are always invertible (from the unitary condition), while some classical gates like NAND are irreversible (e.g., if the output of a NAND gate is 1, we do not know if the input is 00, 01, or 10). The general form of a quantum gate taking in $n$ input qubits is some unitary matrix in $U(2^n)$. There are a lot of significant quantum gates designed for a particular operation. More introduction about quantum gates can be found in [35]. Quantum circuits are usually composed of multiple quantum bits and quantum gates. It is common to use both classical bits and qubits in a quantum circuit. To distinguish between the two, the flow of classical bits is represented by two lines, while the flow of qubits is represented by a single line. Quantum circuits are acyclic, meaning that there are no loops, unlike classical circuits. Classical circuits allow wires to be joined together or split/copied. Quantum circuits cannot do this since it is not invertible. Every gate must be an $n \times n$ unitary operator.

Quantum algorithms are generally implemented by constructing quantum circuits using qubits and quantum gates. For instance, quantum search algorithms are search methods that locate a specific element in an unordered array in logarithmic time. These algorithms outperform classical counterparts by completing the search task in significantly less time. Grover's algorithm, for example, can find the target element in an unordered array in $O(\sqrt{N})$ time, whereas classical algorithms require $O(N)$ time [36]. Such algorithms have been effectively applied in chemistry calculations [37], optimization problems [38], and other fields. In this article, the messages

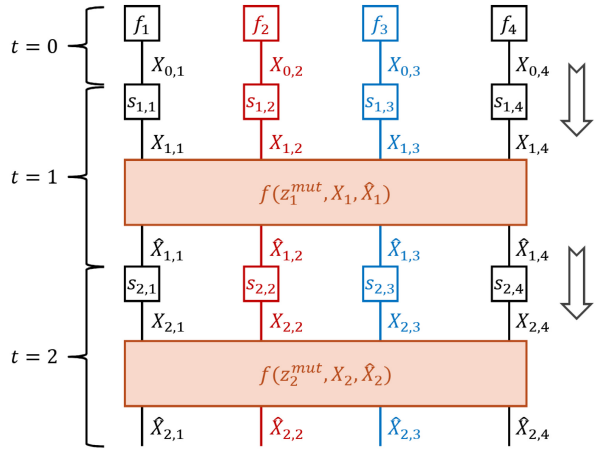



Fig. 1. Classical global FG of $p(\hat{X}_{1:2}|X_{0:2}, Z_{1:2})$, where $k = 0{:}2$, $M = 2$, and $N = 2$.

in BP will be represented by qubits, and Grover's algorithm will also be applied to the message propagation process.

## III. Multitarget Cooperative Motion Model

In the collaborative network, the system consists of $N$ target agents and $M$ fixed nodes. The target agents set is represented as $\mathcal{N} = \{1, 2, \ldots, N\}$, while the fixed nodes set is represented as $\mathcal{M} = \{1, 2, \ldots, M\}$. The time sign is represented by $t_k$. The target agent $n$ can obtain internal measurements $Z_n^{\text{int}}$ (heading angle and step length) via the inertial sensors and mutual measurements $Z_n^{\text{mut}}$ (distance) via the distance sensors. The position information $X_{k+1,p}$ of agent $p$ at time $t_{k+1}$ may be derived by transitioning from $X_{k,p}$ at $t_k$, using the heading angle $\theta_{k,p}$ and step length $S_{k,p}$

$$X_{k+1,p} = X_{k,p} + f\big(\theta_{k,p}, S_{k,p}\big). \tag{1}$$

Then, the internal measurements can be represented as follows:

$$Z_{1:K,p}^{\text{int}} = \big[\theta_{1:K,p}, S_{1:K,p}\big]. \tag{2}$$

Define the distance between the target agent $p$ and $q$ as $d_{k,p,q}$. Then, the mutual measurements can be represented as follows:

$$Z_{1:K,p}^{\text{mut}} = \sum_{q \in \{\mathcal{N} \cup \mathcal{M}\} \setminus \{p\}} d_{1:K,p,q}. \tag{3}$$

## IV. Quantum Belief Propagation for Cooperative Tracking

### A. Global Function Decomposition

This work aims to determine the target agents' location by using their internal and mutual measurements. We mapped it as a global function $f(\hat{X}_{1:K}, X_{0:K}, Z_{1:K})$. $\hat{X}_{1:K}$ and $X_{0:K}$ represent the posterior information and prior information, respectively. $Z_{1:K}$ represents the internal and mutual measurements. The global function could be represented as follows:

$$\begin{aligned} f\Big(\hat{X}_{1:K}, X_{0:K}, Z_{1:K}\Big) &= p\Big(\hat{X}_{1:K}|X_{0:K}, Z_{1:K}^{\text{int}}, Z_{1:K}^{\text{mut}}\Big) \\ &= \frac{p\Big(Z_{1:K}^{\text{mut}}|\hat{X}_{1:K}, X_{0:K}, Z_{1:K}^{\text{int}}\Big)p\Big(\hat{X}_{1:K}, X_{0:K}, Z_{1:K}^{\text{int}}\Big)}{p\big(X_{0:K}, Z_{1:K}^{\text{int}}, Z_{1:K}^{\text{mut}}\big)}. \end{aligned} \tag{4}$$

Given that the mutual measurements are only dependent on the position information, the global function can be simplified as

$$\begin{aligned} &p\Big(\hat{X}_{1:K}|X_{0:K}, Z_{1:K}\Big) \\ &\propto p\big(Z_{1:K}^{\text{mut}}|X_{0:K}\big)p\Big(Z_{1:K}^{\text{int}}|\hat{X}_{1:K}, X_{0:K}\Big)p\Big(\hat{X}_{1:K}, X_{0:K}\Big). \end{aligned} \tag{5}$$

Given that the movement adheres to the principles of a Markov process, it is possible to change the global function as follows:

$$\begin{aligned} &p\big(Z_{1:K}^{\text{mut}}|X_{0:K}\big)p\Big(Z_{1:K}^{\text{int}}|\hat{X}_{1:K}, X_{0:K}\Big)p\Big(\hat{X}_{1:K}, X_{0:K}\Big) \\ &\propto \prod_{k=1}^{K} p\big(Z_k^{\text{mut}}|X_k\big) \prod_{k=1}^{K} p\Big(Z_{1:k}^{\text{int}}|X_k, \hat{X}_{k-1}\Big) \\ &\quad \left(p(X_0) \prod_{k=1}^{K} \big(p(X_k|\hat{X}_{k-1})p(\hat{X}_k|X_k)\big)\right) \\ &\propto \prod_{n=1}^{\mathcal{M}\cup\mathcal{N}} p\big(X_{0,n}\big) \prod_{k=1}^{K} \Big(p\big(Z_k^{\text{mut}}|X_k\big)p\big(\hat{X}_k|X_k\big) \\ &\quad \prod_{n=1}^{\mathcal{M}\cup\mathcal{N}} \big(p(X_{k,n}|\hat{X}_{k-1,n})p(Z_{1:k,n}^{\text{int}}|X_{k,n}, \hat{X}_{k-1,n})\big)\Big) \\ &\propto \prod_{n=1}^{\mathcal{M}\cup\mathcal{N}} \Bigg(p(X_{0,n}) \prod_{k=1}^{K} \big(p(X_{k,n}|\hat{X}_{k-1,n})\big) \\ &\quad p(Z_{1:k,n}^{\text{int}}|X_{k,n}, \hat{X}_{k-1,n})\Bigg) \prod_{k=1}^{K} \Big(p\big(Z_k^{\text{mut}}|X_k\big)p\big(\hat{X}_k|X_k\big)\Big). \end{aligned} \tag{6}$$

Therefore, the global function could be expressed as an FG as shown in Fig. 1. Two black boxes represent fixed nodes, while the red and blue represent two target agents. $S_{k,n}(X_{k,n}, \hat{X}_{k-1,n}) = p(X_{k,n}|\hat{X}_{k-1,n})p(Z_{1:k,n}^{\text{int}}|X_{k,n}, \hat{X}_{k-1,n})$, $f_n(X_{0,n}) = p(X_{0,n})$, and $f(Z_k^{\text{mut}}, X_k, \hat{X}_k) = p(Z_k^{\text{mut}}|X_k)p(\hat{X}_k|X_k)$.

### B. Quantum Representation of Message

The BP method is often used for inference in graphical models, particularly FGs. The joint probability distribution is obtained by multiplying the continuous probability distributions as the message propagates on the FG. However, it is difficult for traditional BP to handle nonlinear and non-Gaussian probability distributions. Drawing inspiration from the concept of Monte Carlo, it is possible to use a collection of particles to estimate the probability distribution of the agent's initial state. Specifically, given the prior distribution $p(X_{0,n})$ of a single agent, the expected value $E[g(x)]$ of the distribution is supposed to be estimated by the Monte Carlo method, where $g(x)$ is some function. The set of sample points $X_{0,n}^i, i \in \{1, 2, \ldots, R\}$ can be obtained by randomly sampling from the distribution $p(X_{0,n})$. The expected value $E[g(x)]$ can be obtained by using the function $g(x)$ to calculate the function value corresponding to the sample points and averaging it, i.e.,

$$E[g(x)] \approx \frac{1}{R} \sum_{i=1}^{R} g\big(X_{0,n}^i\big). \tag{7}$$

The propagation and calculation of messages could be transformed into an operation of particles. Nevertheless, the

precision and complexity of the method will be influenced by the quantity of particles. According to the postulates of quantum computing, a composite system of multiple qubits can be used to map to a particle state space representing a distribution. The basis vectors of a quantum composite system map to one of the particles representing the distribution. The transition of the particle state is evolved in the way of quantum unitary transformation. Let $N_s = 2^m$ denote the quantum particle space size, where $m$ indicates the number of qubits. Then, the mapping relationship between the particle swarm and the composite quantum system can be expressed as

$$\left|\boldsymbol{\mathcal{P}}_k^{(N_s)}\right\rangle \rightarrow \left|\boldsymbol{\mathcal{P}}_k^{(m)}\right\rangle = \sum_{p=00\ldots0}^{\overbrace{11\ldots1}^{m}} C_p|p\rangle \tag{8}$$

where $|\boldsymbol{\mathcal{P}}_k^{(N_s)}\rangle$ denotes the quantum particle set at time $t_k$, whose space size is $N_s$. $|\boldsymbol{\mathcal{P}}_k^{(m)}\rangle$ denotes the composite system consisting of $m$ qubits, used to represent the quantum particle states. $C_p$ denotes the amplitude of state $|p\rangle$, and the condition $\sum_{p=00\ldots0}^{\overbrace{11\ldots1}^{m}} |C_p|^2 = 1$ is satisfied. In other words, $|C_p|^2$ describes the probability that $|\boldsymbol{\mathcal{P}}_k^{(m)}\rangle$ collapses to $|p\rangle$ after measurement. Usually, the initial state of a qubit is the ground state $|0\rangle$, which could be transformed to a superposition by the Hadamard gate $H$ as follows:

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{9}$$

Then, the state space could be initialized by the Hadamard gate with the superposition of $2^m$ possible eigenstates as follows:

$$\begin{aligned}\left|\boldsymbol{\mathcal{P}}_k^{(m)}\right\rangle &= H^{\otimes m}|\overbrace{00\ldots0}^{m}\rangle \\ &= \frac{1}{\sqrt{2^m}}(|00\ldots0\rangle + \ldots + |11\ldots1\rangle) = \frac{1}{\sqrt{2^m}}\sum_{p=0}^{2^m-1}|p\rangle\end{aligned} \tag{10}$$

where $\otimes$ denotes the tensor product operation, which describes the joint state of multiple qubits. According to the quantum representation of particle state, $\frac{1}{\sqrt{2^m}}$ is corresponding to the probability amplitude $C_p$. The Hadamard transform can be represented as follows:

$$H^{\otimes m} = \frac{1}{\sqrt{2^m}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}^{\otimes m} \tag{11}$$

whose function is to transform the ground state into a superposition state. Then, the initial state could be represented by the form of a quantum particle swarm, with each quantum particle executing state transition by function node $\boldsymbol{S}$ with different velocities simultaneously. The process of message transmission is equivalent to the propagation of quantum particles. As shown in Fig. 2, the quantization process is represented by

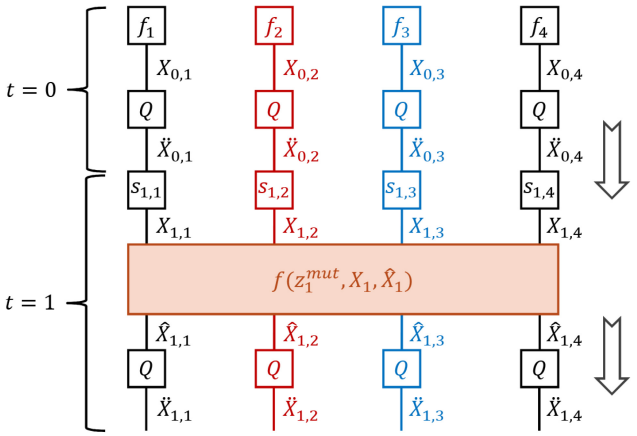



Fig. 2. Quantum global FG of $p(\hat{X}_{1:2}|X_{0:2}, Z_{1:2})$, where $k = 0{:}2$, $M = 2$, and $N = 2$. The defined function node $\boldsymbol{Q}(X, \ddot{X})$ represents the quantization process.

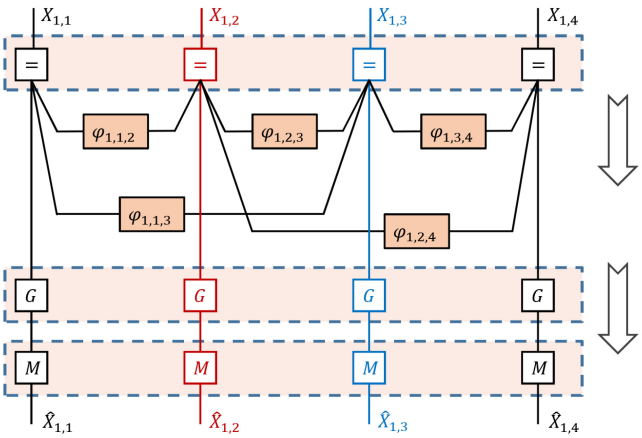



Fig. 3. Quantum local FG of $f(Z_1^{\text{mut}}, X_1, \hat{X}_1)$, where $k = 1$, $M = 2$, and $N = 2$.

a new function node $\boldsymbol{Q}(X, \ddot{X})$, where $X$ denotes the classical variable and $\ddot{X}$ denotes the quantum one. In contrast to the classical FG of Fig. 1, Fig. 2 shows which part of the function node $\boldsymbol{Q}$ plays a role. Due to the quantization of the message, the classical FG of the local function $f(Z_k^{\text{mut}}, X_k, \hat{X}_k)$ will be changed. The following section will detail the quantum message propagation, which introduces some new function nodes.

### C. Quantum Message Propagation

In a classical FG, a message is a probability distribution of being passed from one node to another. Drawing inspiration from the concept of Monte Carlo, it is possible to use a collection of particles to simulate the probability distribution. And quantum superposition allows fewer particles to cover a wider distribution space, which is detailed in Section III-B. However, classical function nodes cannot process quantum messages. Therefore, this article proposes the corresponding quantum function nodes to solve the problem. The quantum message propagation process is divided into four stages, which will be described in detail as follows.

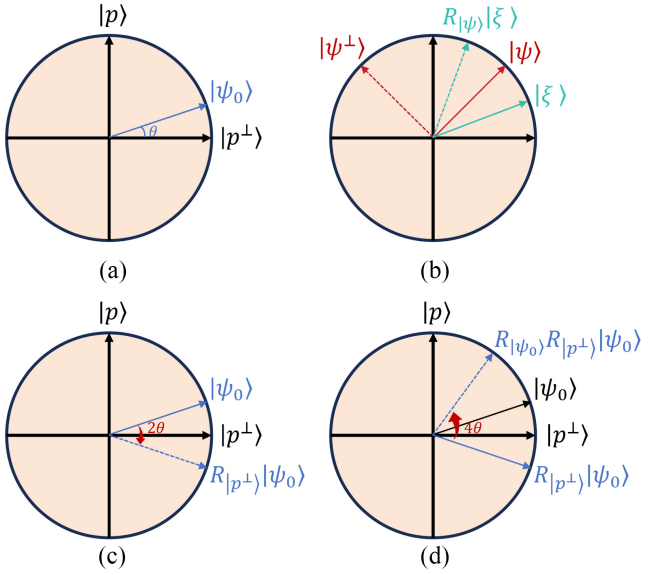



Fig. 4. (a) Particle state space. (b) Reflection operator. (c) and (d) Geometric representation of the Grover node.

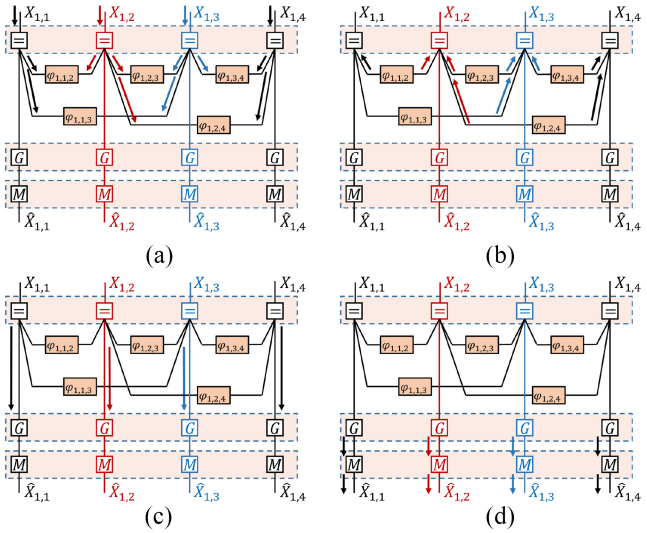



Fig. 5. Four stages of quantum message propagation on FGs. (a) Quantum particle propagation; (b) Share the mutual measurements; (c) Amplify the amplitude of high-weight eigenstates; (d) The quantum states collapse.]. Please confirm if this revision is clear.

*1) Quantum Particle Propagation:* The initial state of an agent could be represented by the form of a quantum particle swarm $|\boldsymbol{\mathcal{P}}^{(N_s)}\rangle$. The $N_s$ quantum particles make state transitions with equal probability according to internal measurements, represented by the node $\boldsymbol{S}$.

*2) Share the Mutual Measurements:* After state transition based on internal measurements, the quantum particle swarm meets the equivalent node, denoted as $=$ as shown in Fig. 3. The equivalent node makes the input message consistent with the output message. It is connected with the other agent's equivalent node by the function node $\varphi_{k,m,n}(X_{k,m}, X_{k,n})$, which means that they are sharing their mutual measurements. After receiving the posterior information from other agents, each agent will update the weight of quantum message. A quantum particle could be viewed as a superposition of classical particles with different weights. However, quantum states cannot provide any information unless they collapse. The quantum message will collapse into a set of high-weight particles if each quantum particle collapses into a high-weight particle, which leads to accurate estimation. For this purpose, we introduce two quantum function nodes: 1) the Grover node $\boldsymbol{G}$ and 2) the Measure node $\boldsymbol{M}$. The Grover node $\boldsymbol{G}$ integrates the famous quantum search algorithm, the Grover's search algorithm, and amplifies the amplitude of the high-weight particles. The Measure node $\boldsymbol{M}$ utilizes the measurement operators to collapse the quantum states.

*3) Amplify the Amplitude of High-Weight Eigenstates:* To effectively use the two kinds of quantum function nodes, it is necessary to carry out a formal transformation on the initial state mentioned above

$$f(\boldsymbol{\mathcal{P}}) = \left|\boldsymbol{\mathcal{P}}^{(m)}_{k,k=0}\right\rangle = \frac{1}{\sqrt{2^m}}|p\rangle + \frac{1}{\sqrt{2^m}}\sum_{x\neq p}|x\rangle = \frac{1}{\sqrt{2^m}}|p\rangle + \frac{\sqrt{2^m-1}}{\sqrt{2^m}}\left|p^{\perp}\right\rangle. \tag{12}$$

The particle state space could be represented as a linear formula of two orthogonal vectors via a transformation process. Let $(1/\sqrt{2^m}) \equiv \sin\theta$, then (12) could be rewritten as

$$f(\boldsymbol{\mathcal{P}}) = |\psi_0\rangle = \sin\theta|p\rangle + \cos\theta\left|p^{\perp}\right\rangle \tag{13}$$

where $|p\rangle$ is the high-weight particle state superposition and $|p^{\perp}\rangle$ is the low-weight one, shown as Fig. 4(a). We will use the Grover node $\boldsymbol{G}$ to perform the unitary transformation on the initial state after obtaining the self-information and mutual measurements, thereby expanding the probability of high-probability particle state collapse. The Grover node $\boldsymbol{G}$ consists of several reflection operators. The reflection operator of any quantum state $|\psi\rangle$ is denoted as $R_{|\psi\rangle}$, which is expressed as follows:

$$R_{|\psi\rangle} = 2|\psi\rangle\langle\psi| - I. \tag{14}$$

The reflection operator [36] is used to convert quantum state $|\xi\rangle = \mu|\psi\rangle + \nu|\psi^{\perp}\rangle$ to $\mu|\psi\rangle - \nu|\psi^{\perp}\rangle$, shown as Fig. 4(b), which could be proved as follows:

$$\begin{aligned} R_{|\psi\rangle}|\xi\rangle &= (2|\psi\rangle\langle\psi| - I)\left(\mu|\psi\rangle + \nu\left|\psi^{\perp}\right\rangle\right) \\ &= 2\mu|\psi\rangle - \mu|\psi\rangle - \nu\left|\psi^{\perp}\right\rangle = \mu|\psi\rangle - \nu\left|\psi^{\perp}\right\rangle. \end{aligned} \tag{15}$$

According to the definition of reflection operator, the reflection operator of $|p^{\perp}\rangle$, i.e., $R_{|p^{\perp}\rangle}$ is represented as $2|p^{\perp}\rangle\langle p^{\perp}| - I$, while the reflection operator of $|\psi_0\rangle$, i.e., $R_{|\psi_0\rangle}$ is represented as $2|\psi_0\rangle\langle\psi_0| - I$. So the Grover node $\boldsymbol{G}$ is shown as follows:

$$\boldsymbol{G} = \prod R_{|\psi_0\rangle}R_{|p^{\perp}\rangle}. \tag{16}$$

It is worth mentioning that $R_{|\psi_0\rangle}$ is prepared by $R_{|0\rangle}$ as follows:

$$R_{|\psi_0\rangle} = H^{\otimes m}(2|0\rangle\langle 0| - I)H^{\otimes m}. \tag{17}$$

As shown in Fig. 4(c) and (d), the amplitude $\sin\theta$ of high-weight eigenstates $|p\rangle$ is amplified to $\sin 4\theta$. As a result, the probability of the high-weight eigenstates being observed in the measurement increases.

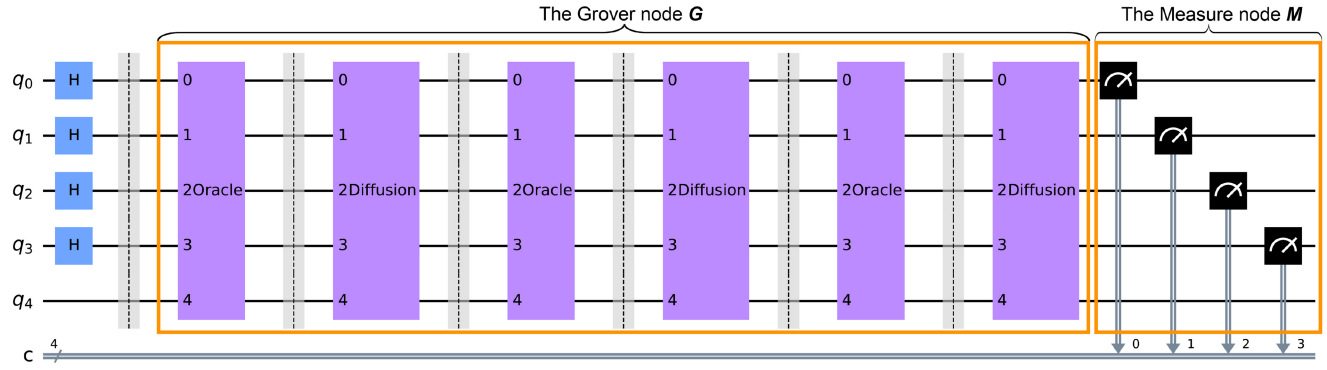



Fig. 6. Quantum circuit of a target agent's estimation process. Four qubits (q0–q3) represent the size of a quantum particle swarm. $c$ denotes the classical bits and q4 is an auxiliary qubit. Oracle and Diffusion are the corresponding quantum operators (reflection operators).

*4) Quantum States Collapse:* The amplitude of $f(\boldsymbol{\mathcal{P}})$ will be updated after meeting the Grover node. The quantum particle with high weight in this superposition state has the largest amplitude and is more likely to be measured with the Measure node $\boldsymbol{M}$. In this article, the projection operator [39] is integrated into $\boldsymbol{M}$ and used to project the high-weight quantum state onto the eigenstate. So the Measure node $\boldsymbol{M}$ is shown as follows:

$$\boldsymbol{M} = |p\rangle\langle p|. \tag{18}$$

When the initial quantum state meets the Measure node, it will collapse to $|p\rangle$ with amplitude $\sin\theta$ as follows:

$$\boldsymbol{M}f(\boldsymbol{\mathcal{P}}) = |p\rangle\langle p|\left(\sin\theta|p\rangle + \cos\theta\left|p^{\perp}\right\rangle\right) = \sin\theta|p\rangle. \tag{19}$$

When the Grover node make the amplitude close to 1, the projection result is the high-weight state approximately. The expected value is calculated according to the projection result, and the new position information is estimated.

Fig. 5 shows how the quantum message is delivered on the local FG, corresponding to the four stages mentioned above.

### D. Complexity Analysis

For better illustration, a quantum circuit of a target agent's estimation process is shown as Fig. 6 to provide a clearer representation. For simplicity, this case only uses four qubits (q0–q3) to construct the quantum particle space. There will be $2^4 = 16$ quantum particles to represent the target agent's state after the Hadamard gate $H$ operation (the blue part in Fig. 6). The mathematical implementation of the Hadamard gate is shown as (11). Then, the Grover node $\boldsymbol{G}$ (the purple part in Fig. 6) amplifies the amplitude of the high-weight particles. The mathematical implementation of the Grover node is shown as (16). The Measure node $\boldsymbol{M}$, the mathematical implementation of which is shown as (18), collapses the quantum state (the black part in Fig. 6).

Here is an analysis about the computational time and delay. In terms of initialization cost, the quantum particle swarm needs to be reinitialized after each measurement, which involves using quantum gate operations such as Hadamard gates to construct new quantum states. This process is relatively efficient in quantum computing. The propagation and update of quantum states are mainly realized by quantum search algorithms. The calculation time is related to the size of the solution space and the number of solutions. This is also the main source of theoretical delay of the proposed algorithm. The implementation of the measurement operation introduces additional delays because the quantum state needs to be processed and converted into classical information after each measurement. This process is relatively efficient in quantum computing. Then, we briefly analyze proposed method's complexity. Grover's search has an ideal number of iterations $N_{\text{optimal}}$ to produce the maximum probability of measuring a valid output. If the problem has $p$ possible terms and $q$ of them are solutions to the problem, then $N_{\text{optimal}} \approx (\pi/4)\sqrt{(p/q)}$ (refer to [40]). For example, $N_{\text{optimal}}$ in Fig. 6 is $N_{\text{optimal}} \approx (\pi/4)\sqrt{16} \approx 3$. So the Grover node in Fig. 6 consists of three $R_{|\psi_0\rangle}R_{|p^{\perp}\rangle}$ operations, where $R_{|p^{\perp}\rangle}$ is named "Oracle" and $R_{|\psi_0\rangle}$ is named "Diffusion." In CT-QBP, the most critical parameters that affect the computation time are the quantum particle space size $N_s$ and the number of target agents $N$. There is only one solution from $N_s$ items for an agent. Then, the complexity of CT-QBP is at most $O(N \times \sqrt{N_s})$. Correspondingly, classical BP algorithms require $O(N \times N_p)$, where $N_p$ represents the classical particle space size. When the classical particle space ($N_p$) and the quantum particle space ($N_s$) are the same size, the computational complexity is reduced from linear $O(n)$ to sublinear $O(\sqrt{n})$ by leveraging the CT-QBP, resulting in a significant improvement in efficiency for large-scale problems.

## V. Numerical Simulation and Analysis

To evaluate the proposed CT-QBP, we conducted several sets of experiments on the cooperative tracking problem, comparing them with three other improved target tracking methods: 1) the UCR-PF method [11]; 2) the AKKF-BP algorithm [17]; and 3) the error constraint-enhanced quantum particle filter (EC-QPF) method [41]. The first two methods (UCR-PF and AKKF-BP) are existing classical advanced methods, and the third method (EC-QPF) is existing quantum-inspired method. The experimental setup utilized a computer system running on the Windows 10 operating system, equipped with a 4-core i5 CPU and 8 GB of memory. Each simulation process commenced with the target agent starting from its initial position and then proceeding to

walk for a duration of 60 steps. In this section, we first focused on two typical experimental scenarios to verify the relationship between algorithm performance and collaborative network size: 1) a cooperative tracking simulation experiment without fixed nodes ($N = 4, M = 0$) and 2) a cooperative tracking simulation experiment with fixed nodes ($N = 4, M = 4$). The target agents keep random walk in these two experimental scenarios. After that we simulated the scene of change in the number of targets to verify the relationship between algorithm performance and number of targets. Then, in order to verify the relationship between the algorithm performance and the agent's mobility pattern, we simulated the scene of the agent performing circular motion. Finally, the performance of the algorithm under different sensor noise conditions was verified.

### A. Accuracy Verification Experiment ($N = 4, M = 0$)

In this experiment, we set the number of target agents to 4 and the number of fixed nodes to 0. The initial positions of the target agents are (20, 20), (40, 20), (40, 40), and (20, 40). Initially, each target agent starts from its respective initial position and traverses randomly for 60 steps. The movement trajectories of the four target agents are recorded as the actual trajectories. Additionally, we capture the inertial measurement information at each step, with each target agent measuring distances from all other target agents except itself. The root mean square error between the true and predicted positions is calculated at each step to evaluate algorithm performance. This error serves as the evaluation criterion for algorithm performance. The formula for calculating the error at each step is as follows:

$$e_i = \sqrt{\frac{1}{N}\sum_{\mathrm{n}=1}^{N}\left(\left(\hat{x}_{i,n} - x_{i,n}\right)^2 + \left(\hat{y}_{i,n} - y_{i,n}\right)^2\right)} \tag{20}$$

where $i$ is the current step, and $N$ is the number of the target agent. The real position of the target agent $n$ at step $i$ is denoted as $(x_{i,n}, y_{i,n})$, while the estimated position of the target agent $n$ at step $i$ is denoted as $(\hat{x}_{i,n}, \hat{y}_{i,n})$.

As depicted in Fig. 7(a), the red line represents the results of CT-QBP, while the orange, blue, and green lines represent the results of AKKF-BP, UCR-PF, and EC-QPF, respectively. The estimation error exhibits a gradually increasing trend as the target agents move over time, thus confirming the cumulative error and drifting issues associated with IMUs. UCR-PF utilizes a constrained optimization filtering algorithm, which, although slowing down the error accumulation process, leads to a gradual reduction in positioning accuracy. On the other hand, AKKF-BP achieves relatively high accuracy through the BP algorithm, yet the error trend displays fluctuations. The quantum inspired algorithm EC-QPF performs similarly to AKKF-BP. In contrast, CT-QBP not only achieves the highest positioning accuracy but also maintains a stable error trend. This is attributed to CT-QBP's integration of internal and mutual measurements to enhance positioning accuracy, along with its utilization of quantum characteristics to mitigate error accumulation during the state transition process. Fig. 7(b) shows the estimated trajectory of each algorithm visually. Moreover, as illustrated in Fig. 7(c), we repeat the aforementioned experimental process 20 times and calculate the mean square error. CT-QBP exhibits the least error fluctuation and the highest positioning accuracy, underscoring the robustness and effectiveness of the algorithm during execution.

### B. Accuracy Verification Experiment ($N = 4, M = 4$)

In this experiment, we selected four target agents and four fixed nodes. The initial positions of the target agents are (20, 20), (40, 20), (40, 40), and (20, 40), while the initial positions of the fixed nodes are (0, 0), (60, 0), (60, 60), and (0, 60). Initially, each target agent commenced from its respective initial position and moved randomly for 60 steps. The trajectories of the four target agents were recorded as the actual trajectories. Additionally, we recorded the inertial measurement information at each step, with each target agent measuring distances from all fixed nodes and other target agents except itself. The root-mean-square error between the true and predicted positions was calculated at each step.

As illustrated in Fig. 8(a), CT-QBP maintained the highest accuracy, consistent with the experimental conclusion described in the previous section. This shows that the method proposed in this article is applicable to both scenarios with and without fixed nodes. Fig. 8(b) shows the estimated trajectory of each algorithm visually. Furthermore, in 20 repetitions of the experiment, as depicted in Fig. 8(c), CT-QBP exhibited the least error fluctuation and the highest positioning accuracy. For the UCR-PF or AKKF-BP algorithms, the accuracy improved upon adding fixed nodes compared to cases without fixed nodes. We attribute this improvement to the fixed nodes providing more reliable information compared to the target nodes, given their stationary nature. After adding fixed nodes, the accuracy of EC-QPF is slightly better than that of AKKF-BP. This shows that quantum inspired method EC-QPF is suitable for tracking tasks that provide reliable information, while classical BP method AKKF-BP is suitable for collaborative tracking tasks without fixed nodes. Regardless of the presence of fixed nodes in the cooperative network, the proposed algorithm's accuracy and stability remained largely unaffected. This resilience is attributed to the superposition states of quantum particles counteracting the uncertainty generated in the estimation process.

### C. Target Number Verification Experiment ($N = 3/6/12$)

In order to verify the performance of the proposed algorithm under different target number, we set the number of target agents to 3, 6, and 12, respectively, to compare the tracking accuracy, while the number of fixed nodes is set to 0. The initial positions of the target agents are random. Each target agent starts from its respective initial position and traverses randomly for 60 steps.

Fig. 9(a)–(c) shows the results when the number of target agents is 3, 6, and 12, respectively. As illustrated in Fig. 9, CT-QBP maintained the highest accuracy, consistent with the experimental conclusion described in the previous section. This proves that it maintains excellent performance even in different target number. However, the accuracy of other

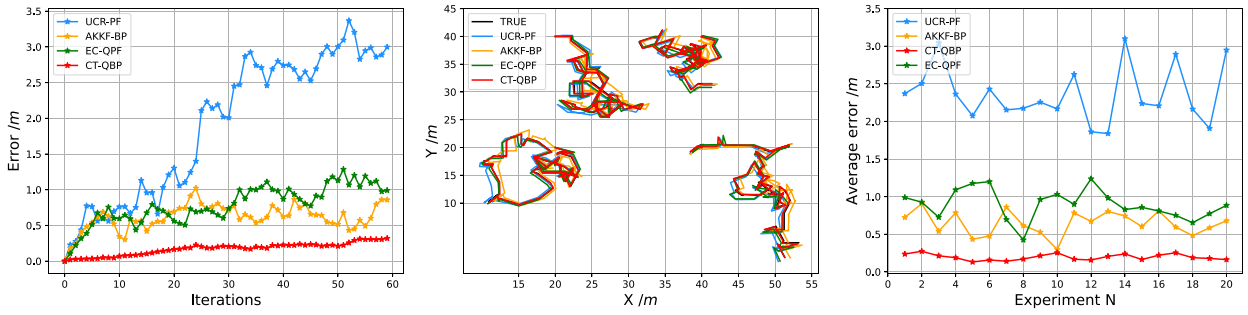



Fig. 7. (a) Estimation error trend ($N = 4$, $M = 0$). (b) Tracking results ($N = 4$, $M = 0$). (c) Average error trend under 20 repeated experiments ($N = 4$, $M = 0$).

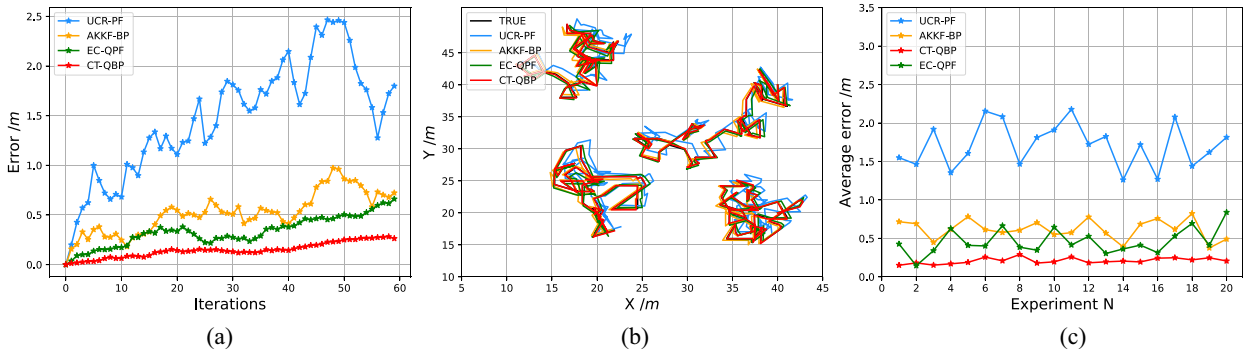



Fig. 8. (a) Estimation error trend ($N = 4$, $M = 4$). (b) Tracking results ($N = 4$, $M = 4$). (c) Average error trend under 20 repeated experiments ($N = 4$, $M = 4$).

algorithms can only be improved as the number of target agents increases.

### D. Mobility Pattern Verification Experiment

In order to verify the performance of the proposed algorithm under different motion modes, the target agents keep circular motion, which is different from the previous experimental settings. In this experiment, we set the number of target agents to 4 and the number of fixed nodes to 0. The initial positions of the target agents are (20, 20), (40, 20), (40, 40), and (20, 40). Each target agent starts from its respective initial position and traverses circularly for 60 steps. The trajectory and estimated results of circular motion are shown in Fig. 10(b).

As illustrated in Fig. 10(a), CT-QBP maintained the highest accuracy, consistent with the experimental conclusion described in the previous subsection. This proves that it maintains excellent performance even in different mobility pattern. However, classical tracking algorithm UCR-PF is affected by mobility pattern. In circular motion mode, its accuracy is obviously improved compared to random motion mode. Furthermore, in 20 repetitions of the experiment, as depicted in Fig. 10(c), CT-QBP exhibited the least error fluctuation and the highest positioning accuracy.

### E. Sensors Noise Verification Experiments

To demonstrate the robustness of the CT-QBP, we conducted experiments by altering the variance of internal measurements (step length information and heading angle information) and mutual measurements (distance information) separately. This section presents the robustness performance of the four algorithms in the experimental scenarios outlined in the previous sections, depicted in the same result plot. This approach allows for easy investigation into which algorithm is the optimal choice when sensor accuracies, represented by varying variances, differ. The experimental results presented in this section represent the average errors from 20 experiments. Dashed lines indicate the algorithm's application in a cooperative localization environment without fixed nodes, while solid lines represent cases with fixed nodes. From Fig. 11(a)–(c), the following conclusions can be drawn.

1) The CT-QBP algorithm demonstrates superior performance in both environments, with and without fixed nodes. It maintains optimal performance even with increased single variance. This superiority is attributed to the CT-QBP's utilization of the quantum superposition property to mitigate uncertainty associated with message fusion in BP, thereby enhancing the accuracy and stability of MCL.
2) Comparing Fig. 11(a) and (b) with Fig. 11(c), it becomes evident that the algorithm's performance is more susceptible to the uncertainty of mutual measurements. This susceptibility arises because the uncertainty of internal measurements can be offset by the uncertainty of quantum states, while mutual measurements, serving as the basis for the quantum factor to determine state weight, indirectly affect estimation results.
3) Fig. 11(c) illustrates that with an increase in distance variance, the accuracy of the CT-QBP algorithm with fixed nodes surpasses and falls below that of

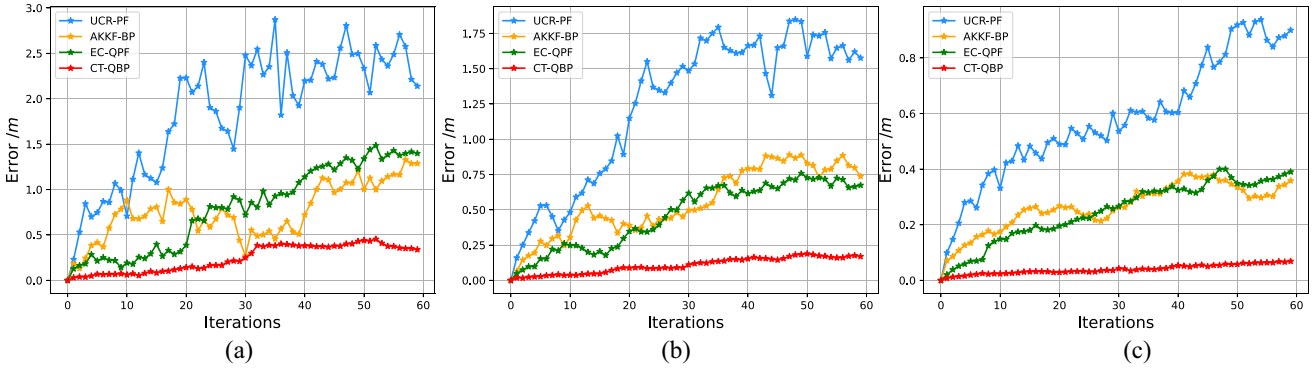



Fig. 9. (a) Estimation error trend ($N$ = 3). (b) Estimation error trend ($N$ = 6). (c) Estimation error trend ($N$ = 12).

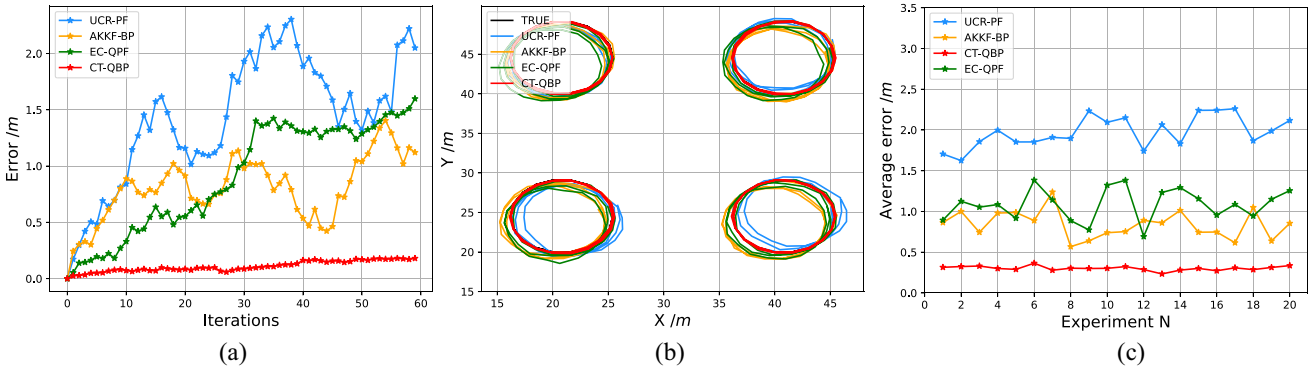



Fig. 10. (a) Estimation error trend (walk circularly). (b) Tracking results (walk circularly). (c) Average error trend under 20 repeated experiments (walk circularly).

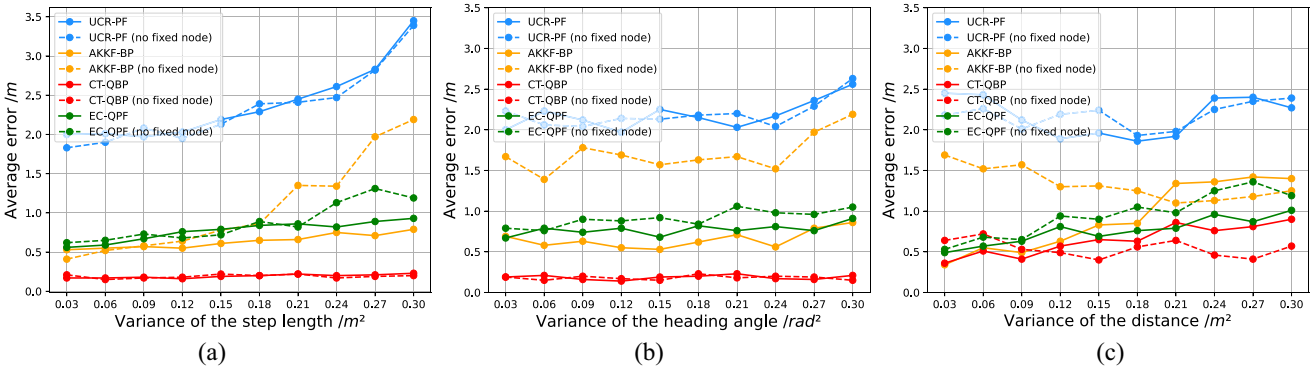



Fig. 11. (a) Trend of algorithm error for various step length variance cases. (b) Trend of algorithm error for various heading angle variance cases. (c) Trend of algorithm error for various distance variance cases.

the algorithm without fixed nodes. This discrepancy occurs because when distance information uncertainty is low, the fixed node provides more accurate distance information to the target agent than the information between target agents. As distance information uncertainty increases, the uncertainty between target agents' distances and their position uncertainty counterbalance each other.

## VI. Conclusion

In this article, we proposed a novel cooperative target tracking algorithm, namely, the quantum-inspired BP algorithm (CT-QBP), to address the limitations of existing localization algorithms in multitarget cooperative tracking. The algorithm leverages the quantum superposition property to mitigate uncertainty associated with message fusion in BP, thereby enhancing the accuracy and stability of MCL. We adopted the quantum Monte Carlo idea to simulate message distribution quantumly, where quantum particles represent the superposition of multiple states simultaneously. This not only resolves intractable integrations in message updating on the FG but also ensures the algorithm's independence from the number of particles. FG function nodes were quantized using quantum unitary transformations and closed box operations to propagate quantum messages. This innovation circumvents the

incapacity of traditional FG function nodes to process quantum messages and introduces a novel approach to implementing quantum function nodes. Experimental results demonstrated the algorithm's superior accuracy and robustness.

This work represents the first endeavor to integrate quantum computing with FGs and BP to tackle the multiobject cooperative localization problem. With the ongoing advancement and refinement of quantum technologies, there is widespread anticipation that quantum computing will broaden its applications, fostering further innovation and progress in human society's development.

## REFERENCES


[1] S. M. Esmailifar and F. Saghafi, "Cooperative localization of marine targets by uavs," *Mech. Syst. Signal Process.*, vol. 87, pp. 23–42, Mar. 2017. [Online]. Available: http://dx.doi.org/https://doi.org/10.1016/j.ymssp.2016.08.027

[2] O. Shorinwa, J. Yu, T. Halsted, A. Koufos, and M. Schwager, "Distributed multi-target tracking for autonomous vehicle fleets," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 3495–3501.

[3] B. Sumantri, N. Uchiyama, and S. Sano, "Least square based sliding mode control for a quad-rotor helicopter and energy saving by chattering reduction," *Mech. Syst. Signal Process.*, vol. 66, pp. 769–784, Jan. 2016.

[4] Y. Sung, A. K. Budhiraja, R. K. Williams, and P. Tokekar, "Distributed assignment with limited communication for multi-robot multi-target tracking," *Auton. Robots*, vol. 44, no. 1, pp. 57–73, 2020.

[5] M. Corah and N. Michael, "Scalable distributed planning for multi-robot, multi-target tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021, pp. 437–444.

[6] X. Bo, A. A. Razzaqi, X. Wang, and G. Farid, "Optimal geometric configuration of sensors for received signal strength based cooperative localization of submerged AUVs," *Ocean Eng.*, vol. 214, Oct. 2020, Art. no. 107785.

[7] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, and N. Zerhouni, "Particle filter-based prognostics: Review, discussion and perspectives," *Mech. Syst. Signal Process.*, vols. 72–73, pp. 2–31, May 2016.

[8] S. Martínez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[9] P. Zhan, D. W. Casbeer, and A. L. Swindlehurst, "Adaptive mobile sensor positioning for multi-static target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 1, pp. 120–132, Jan. 2010.

[10] G. Ivey and E. Johnson, "Investigation of methods for simultaneous localization and mapping using vision sensors," in *Proc. AIAA Guid., Naviga., Control Conf. Exhibit*, 2006, p. 6578.

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

[12] D. Gaglione, P. Braca, and G. Soldi, "Belief propagation based ais/radar data fusion for multi-target tracking," in *Proc. 21st Int. Conf. Inf. Fusion (FUSION)*, 2018, pp. 2143–2150.

[13] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[14] F. Meyer, P. Braca, P. Willett, and F. Hlawatsch," Tracking an unknown number of targets using multiple sensors: A belief propagation method," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, 2016, pp. 719–726.

[15] N. Wu, W. Yuan, H. Wang, and J. Kuang, "TOA-based passive localization of multiple targets with inaccurate receivers based on belief propagation on factor graph," *Digit. Signal Process.*, vol. 49, pp. 14–23, Feb. 2016. [Online]. Available: http://dx.doi.org/https://doi.org/10.1016/j.dsp.2015.10.013

[16] P. Sharma, A.-A. Saucan, D. J. Bucci, and P. K. Varshney, "Decentralized gaussian filters for cooperative self-localization and multi-target tracking," *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5896–5911, Nov. 2019. [Online]. Available: http://dx.doi.org/10.1109/TSP.2019.2946017

[17] M. Sun, M. E. Davies, I. K. Proudler, and J. R. Hopgood, "Adaptive kernel Kalman filter based belief propagation algorithm for maneuvering multi-target tracking," *IEEE Signal Process. Lett.*, vol. 29, pp. 1452–1456, 2022. [Online]. Available: http://dx.doi.org/10.1109/LSP.2022.3184534

[18] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Data-driven factor graphs for deep symbol detection," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2020, pp. 2682–2687.

[19] H. Wang, Y. Zhang, J. Zhang, T. Li, and L. Peng, "A factor graph model for unsupervised feature selection," *Inf. Sci.*, vol. 480, pp. 144–159, Apr. 2019.

[20] Z. Zhang, F. Wu, and W. S. Lee, "Factor graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 8577–8587.

[21] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2434–2444, Aug. 2021.

[22] Z. Wang, S. Y. M. Lee, S. Li, and G. Zhou, "Emotion analysis in code-switching text with joint factor graph model," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 3, pp. 469–480, Mar. 2017.

[23] L. Mutimbu and A. Robles-Kelly, "A factor graph evidence combining approach to image defogging," *Pattern Recognit.*, vol. 82, pp. 56–67, Oct. 2018.

[24] C. Tang, L. Zhang, Y. Zhang, and H. Song, "Factor graph-assisted distributed cooperative positioning algorithm in the GNSS system," *Sensors*, vol. 18, no. 11, p. 3748, 2018.

[25] G. Zhang, H.-F. Ng, W. Wen, and L.-T. Hsu, "3D mapping database aided GNSS based collaborative positioning using factor graph optimization," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6175–6187, Oct. 2021.

[26] W. Xiwei, X. Bing, W. Cihang, G. Yiming, and L. Lingwei, "Factor graph based navigation and positioning for control system design: A review," *Chin. J. Aeronaut.*, vol. 35, no. 5, pp. 25–39, 2022.

[27] S. J. Rennie, J. R. Hershey, and P. A. Olsen, "Hierarchical variational loopy belief propagation for multi-talker speech recognition," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, 2009, pp. 176–181.

[28] D. Burkett and D. Klein, "Fast inference in phrase extraction models with belief propagation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. Human Lang. Technol.*, 2012, pp. 29–38.

[29] T. Liu and X. Chen, "Deep learning-based belief propagation algorithm over non-binary finite fields," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2020, pp. 164–169.

[30] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky, "Visual hand tracking using nonparametric belief propagation," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, pp. 189–189.

[31] M. Cheng, M. Qiu, X. Shi, J. Huang, and W. Lin, "One-shot text field labeling using attention and belief propagation for structure information extraction," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 340–348.

[32] S. Van de Velde, G. Arora, L. Vallozzi, H. Rogier, and H. Steendam, "Cooperative hybrid localization using Gaussian processes and belief propagation," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, 2015, pp. 785–790.

[33] V. Savic and S. Zazo, "Cooperative localization in mobile networks using nonparametric variants of belief propagation," *Ad Hoc Netw.*, vol. 11, no. 1, pp. 138–150, 2013.

[34] A. Steane, "Quantum computing," *Rep. Progr. Phys.*, vol. 61, no. 2, p. 117, 1998.

[35] D. Dong and I. R. Petersen, "Quantum control theory and applications: A survey," *IET Control Theory Appl.*, vol. 4, no. 12, pp. 2651–2671, 2010.

[36] G.-L. Long, "Grover algorithm with zero theoretical failure rate," *Phys. Rev. A*, vol. 64, no. 2, 2001, Art. no. 22307.

[37] D. E. Bernal, A. Ajagekar, S. M. Harwood, S. T. Stober, D. Trenev, and F. You, "Perspectives of quantum computing for chemical engineering," *AIChE J.*, vol. 68, no. 6, 2022, Art. no. e17651.

[38] K. Zhang and V. E. Korepin, "Depth optimization of quantum search algorithms beyond grover's algorithm," *Phys. Rev. A*, vol. 101, no. 3, 2020, Art. no. 32346.

[39] R. H. Preston, "Applying Grover's algorithm to hash functions: A software perspective," *IEEE Trans. Quantum Eng.*, vol. 3, pp. 1–10, Jan. 20223.

[40] G. M. Vinod and A. Shaji, "Finding solutions to the integer case constraint satisfiability problem using Grover's algorithm," *IEEE Trans. Quantum Eng.*, vol. 2, pp. 1–13, Oct. 2021. [Online]. Available: http://dx.doi.org/10.1109/TQE.2021.3120449

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

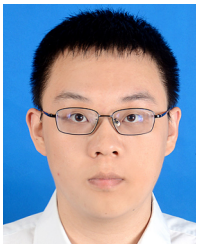
**Jiawang Wan** received the B.E. degree from the University of Science and Technology Beijing, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree.

His research interests include wireless localization, swarm intelligence, and 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), Beijing, China, in 2012, 2015, and 2019, respectively.

He was a Visiting Scholar with the Université libre de Bruxelles, Brussels, Belgium, and Nanyang Technological University, Singapore. He is currently an Associate Professor with the School of Computer and Communication Engineering, USTB. He was supported by the Postdoctoral Innovative Talent Support Program from Chinese government in 2019. His current research interests include swarm intelligence, multirobots network, quantum machine learning, distributed security, and Internet of Things.

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



**Yuchen Shi** received the B.E. and M.S. degrees from the University of Science and Technology Beijing (USTB), Beijing, China, in 2021 and 2024, respectively.

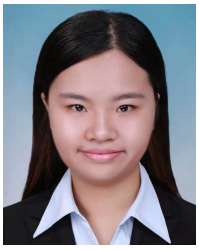
He is currently working as a Research Assistant with USTB. His research interests include multiagent systems, reinforcement learning, and Internet of Things.

**Weizhao Chen** received the M.S. degree from the University of Science and Technology Beijing, Beijing, China, in 2024.

His research interests include multiagent systems and Internet of Things.

**Fangwen Ye** received the B.E. degree from the University of Science and Technology Beijing, Beijing, China, in 2022, where he is currently pursuing the master's degree.

His research interests include multiagent systems and Internet of Things.



**Ran Wang** (Graduate Student Member, IEEE) received the B.E. degree from Beijing Information Science and Technology University, Beijing, China, in 2013, and the M.S. degree from the University of Science and Technology Beijing, Beijing, in 2016, where she is currently pursuing the Doctoral degree with the Microarchitecture and Integrated Circuits Laboratory.

Her research interests include multiagent systems, distributed security, and Internet of Things.

**Xiaotong Zhang** (Senior Member, IEEE) received the M.S. and Ph.D. degrees from the University of Science and Technology Beijing, Beijing, China, in 1997 and 2000, respectively.

He is a Professor with the Department of Computer Science and Technology, University of Science and Technology Beijing. His research includes wireless sensor networks and computer architecture.