

# An iterative method for the distance constraints in a multi-sensor positioning system

Tao Liu, Jian Kuang, and Xiaoji Niu

**Abstract**—The distance constraint can enhance the state estimation performance of a multi-sensor positioning system. However, the existing methods encounter problems such as low state estimation accuracy and high computational complexity. This study proposes an iterative constraint algorithm that effectively solves the distance constraint problem in a multi-sensor positioning system. The proposed algorithm linearizes the distance constraint in each iteration to obtain an approximate linear constraint model. Then, it re-estimates the approximate system state by using the estimation projection algorithm. In the last iteration, the proposed algorithm uses the approximate state estimate as input to estimate a more accurate system state until the iteration is terminated. Three simulations are provided to demonstrate the effectiveness and superiority of the proposed algorithm.

**Index Terms**—Multi-sensor system, distance constraint, state estimation, iterative estimation.

## I. INTRODUCTION

THE growing demand for location-based services warrants the acquisition of accurate and reliable position information. The multi-sensor system has distinct advantages over the single-sensor system in terms of positioning accuracy and reliability [1, 2], fault or interference tolerance, and more [3]. The multi-sensor system has the ability to implement the complementary advantages of the different types of sub-sensors. Moreover, some periodic and objective characteristic signals in the different nodes of a mobile carrier can further be used to enhance system positioning performance. In a multi-sensor system, some inherent constraints are contained among the sub-sensors arising from the physical platform, mathematical and physical attributes, and environmental characteristics. For example, non-holonomic constraints [4] in land-based vehicle positioning, normalized constraints [5, 6] in quaternion-based attitude estimation, and joint kinematics constraints [7, 8] between body segments in pedestrian positioning.

In multi-sensor systems, distance constraint is a typical inherent constraint. As shown in Fig. 1, three low-cost inertial sensors are mounted on the left and right wheels and back of the vehicle or robot in a wheel-based navigation system [9, 10]. The determinate and constant distance between the three sensors can improve the inertial-based dead-reckoning performance. The distance between adjacent joints (e.g., the

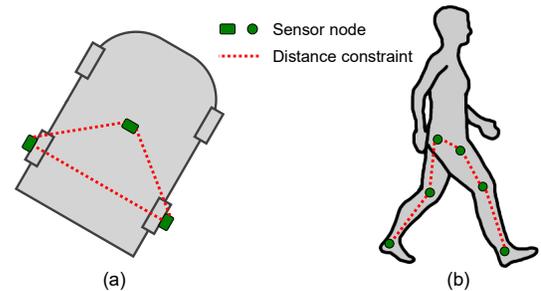


Fig. 1. Examples of distance constraint in the multi-sensor system. (a) Wheel-based navigation system. (b) Wearable body sensor network (WBSN).

knee and hip joints) in the human body is constant [8]. The constant distance constraint can be used to enhance the performance of wearable body sensor networks (WBSN) in various applications such as motion monitoring [7, 11, 12], disease treatment and rehabilitation [13, 14], and daily activity analysis [15]. Therefore, exploiting the potential of distance constraints to improve the positioning performance of multi-sensor systems is crucial. In this study, we consider the implementation of an accurate and consistent state estimation based on the distance constraints of a multi-sensor positioning system in the Kalman filter framework.

The classical method of distance constraint in a multi-sensor system is to use the distance to directly construct the position correlation between any two sub-sensors. For example, in WBSN, the distance constraint between feet and knee is simplified to the height difference, which can be used to establish the positional relationship between the two [8, 11]. Unfortunately, the distance constraint in this approach does not help improve state estimation performance. An effective strategy is to translate the distance constraint into the lever arm in inertial navigation algorithms [4, 10]. In this case, the attitude estimate using inertial sensor observations and the corresponding lever arm are combined to effectively improve positioning performance. However, this distance constraint strategy is not universal, as it can only be applied to multi-sensor systems with inertial sensors.

Furthermore, the distance constraint is nonlinear. Therefore, the pseudo-observation approach is a classical strategy for addressing distance constraints [16, 17]. This approach combines the distance constraints and original measurement equation to develop an augmented model. Moreover, the optimal state estimation is implemented using a nonlinear filter. Typical used nonlinear filters include extended Kalman filter (EKF), unscented Kalman filter (UKF), cubature Kalman filter (CKF), and particle filter (PF). EKF transforms the nonlinear model into a linear model using the first-order Taylor expansion and

Manuscript received July 12, 2022; revised XX XX, 202X; accepted XX XX, 202X. Date of publication XX XX, 202X. This work was supported by the National Natural Science Foundation of China under Grant 41904019 and Grant 41974024. (Corresponding author: Jian Kuang.)

T. Liu, J. Kuang and X. Niu are with the GNSS Research Center, Wuhan University, Wuhan, Hubei, CO 430072 PR China. X. Niu is also with the Artificial Intelligence Institute, Wuhan University and the Hubei Luojia Laboratory. (E-mails: liu\_tao@whu.edu.cn; kuang@whu.edu.cn; xjniu@whu.edu.cn).

implements recursive updates of the system state and covariance matrix using the KF [16, 18]. UKF employs an unscented transformation to approximate the posterior probability density function of a nonlinear model [19–22]. CKF uses a third-degree spherical–radial cubature rule to approximate the posterior mean and variance of the nonlinear model [23, 24]. Moreover, PF is an optimal recursive Bayesian filter based on a sequential Monte Carlo simulation [25–27]. In general, the pseudo-observation methods cannot achieve a balance between state estimation accuracy and computational complexity. When using EKF, UKF, and CKF, the state estimation accuracy of the pseudo-observation methods is significantly limited owing to the nonlinear property in the distance-constrained problem. The pseudo-observation approach requires a huge computational effort to achieve accurate state estimation performance when PF is used. Furthermore, as the number of sub-sensors increases, more distance constraint information becomes available, which causes a sharp increase in the computational burden of the pseudo-observation approach.

The distance constraint can be reduced to a quadratic constraint model when a multi-sensor positioning system contains only two sub-sensors. The constraint model is  $\mathbf{x}^T \mathbf{M} \mathbf{x} = \mathbf{d}$ , where  $\mathbf{x}$  is the system state,  $\mathbf{M}$  is the coefficient matrix, and  $\mathbf{d}$  is the distance constraint vector. The quadratic constraint problem can be considered a conditional extremum problem of minimum state variance under the quadratic equality constraint condition [28, 29], which can be solved using the Lagrange multiplier technique. Furthermore, singular value decomposition (SVD) and Cholesky factorization methods can be used to transform the conditional extremum into a solution of a polynomial equation, with the Lagrange multiplier as the only independent variable. The quadratic constraint method can provide an accurate analytical solution for the system state vector. However, this method cannot obtain an analytical solution for the system covariance matrix. Although some studies have obtained an approximate covariance matrix [20, 30], inconsistencies still exist between the estimated covariance matrix and state vector. This inconsistency leads to the poor numerical stability of the quadratic constraint method. Furthermore, the quadratic constraint method is only suitable for positioning systems using only two sub-sensors. Therefore, it cannot be generalized to a multi-sensor system with multiple distance constraints (three or more sub-sensors).

To solve the problem of multiple distance constraints in the multi-sensor positioning system, an iterative distance constraint algorithm is proposed in this study. First, an unconstrained centralized filter is used to obtain the initial state estimate and corresponding covariance matrix. Subsequently, an iterative algorithm is presented to solve the multiple distance constraints in the multi-sensor system. In each iteration of the proposed algorithm, the multiple distance constraints are approximated as a linear constraint optimization problem. The classical estimation projection method is employed to acquire an approximate state estimate in a single iteration. Finally, the optimal estimate is determined from alternative estimates obtained through continuous iterative updates. Specifically, the contributions of the study are summarized as follows:

(1) We separated the distance constraint and unconstrained

filter and formulate the distance constraint problem as a general optimization model. From the algorithm architecture level, the computational complexity was reduced and the interference of the number of sub-sensors was isolated.

- (2) We presented an iterative strategy that permits the updated system state to continuously approximate the distance constraint model. The nonlinear effect of the distance constraints was weakened such that accurate system state estimation can be obtained.
- (3) We established the estimated state error in each iteration as a linear function related to the unconstrained state error, which ensures that the estimated covariance matrix is consistent with the system state.

The remainder of this paper is organized as follows. The problem formulation is given in Section II. Section III reviews the unconstrained filter, pseudo-observation, and quadratic constraint algorithms. The proposed iterative distance constraint algorithm is derived in Section IV. Simulation results are presented in Section V. Section VI concludes this paper.

## II. PROBLEM FORMULATION

Consider a multi-sensor system composed of  $s$  sub-sensors, each of which is indexed by an integer  $i \in \{1, 2, \dots, s\}$ . The dynamic model of the  $i$ th sub-sensor is represented by a nonlinear discrete-time system as follows:

$$\begin{cases} \mathbf{x}_{k+1}^i = \mathbf{f}_i(\mathbf{x}_k^i) + \mathbf{w}_k^i \\ \mathbf{z}_k^i = \mathbf{h}_i(\mathbf{x}_k^i) + \mathbf{v}_k^i \end{cases} \quad (1)$$

where  $k$  is the time index,  $\mathbf{x}_k^i \in \mathbb{R}^{n_i}$  is the state vector,  $\mathbf{f}_i(\cdot)$  is the known state transition function,  $\mathbf{z}_k^i \in \mathbb{R}^{m_i}$  is the measurement vector,  $\mathbf{h}_i(\cdot)$  is the known measurement function. The process noise  $\mathbf{w}_k^i \in \mathbb{R}^{n_i}$  and measurement noise  $\mathbf{v}_k^i \in \mathbb{R}^{m_i}$  are assumed to be zero-mean white Gaussian noise with covariance matrices  $\mathbf{Q}_k^i$  and  $\mathbf{R}_k^i$ . Moreover,  $\mathbf{x}_k^i$ ,  $\mathbf{w}_k^i$ , and  $\mathbf{v}_k^i$  are assumed to be uncorrelated to one another. For a linear discrete-time system, the state transition function  $\mathbf{f}_i(\cdot)$  and measurement function  $\mathbf{h}_i(\mathbf{x}_k^i)$  can be modeled as  $\mathbf{F}_k^i \mathbf{x}_k^i$  and  $\mathbf{H}_k^i \mathbf{x}_k^i$ ,  $\mathbf{F}_k^i \in \mathbb{R}^{n_i \times n_i}$  and  $\mathbf{H}_k^i \in \mathbb{R}^{m_i \times n_i}$  are the known state transition and measurement matrices.

We used a centralized filter to simultaneously estimate the state of multiple sub-sensors. The multi-sensor dynamical system can be integrated into a state-space model as follows:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{w}_k \\ \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \end{cases} \quad (2)$$

where  $\mathbf{\Lambda}_k = \text{coa}(\mathbf{\Lambda}_k^1, \dots, \mathbf{\Lambda}_k^p)$ ,  $\mathbf{\Lambda} \in \{\mathbf{x}, \mathbf{w}, \mathbf{z}, \mathbf{v}, \mathbf{f}(\cdot), \mathbf{h}(\cdot)\}$ , and  $p \in \{s, m\}$ . For a linear discrete-time system,  $\mathbf{\Gamma}_k = \text{cob}(\mathbf{\Gamma}_k^1, \dots, \mathbf{\Gamma}_k^p)$ , and  $\mathbf{\Gamma} \in \{\mathbf{F}, \mathbf{H}\}$ . The details of the vector  $\mathbf{\Lambda}$  and matrix  $\mathbf{\Gamma}$  are given as follows:

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}^1 \\ \vdots \\ \mathbf{\Lambda}^p \end{bmatrix}; \quad \mathbf{\Gamma} = \begin{bmatrix} \mathbf{\Gamma}^1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \mathbf{\Gamma}^p \end{bmatrix} \quad (3)$$

Evidently, the noise  $\mathbf{w}_k$  and  $\mathbf{v}_k$  in the state-space model (2) still obey the Gaussian noise distribution with zero mean

vectors and covariance matrices  $\mathbf{Q}_k = \text{cob}(\mathbf{Q}_k^1, \dots, \mathbf{Q}_k^s)$  and  $\mathbf{R}_k = \text{cob}(\mathbf{R}_k^1, \dots, \mathbf{R}_k^m)$ .

The distance constraint between two sub-sensors in a multi-sensor system can be written as follows:

$$d_{i,j} = \sqrt{(x_k^i - x_k^j)^2 + (y_k^i - y_k^j)^2 + (z_k^i - z_k^j)^2} \quad (4)$$

where  $i$  represent the  $i$ th sub-sensor system,  $k$  is the time index,  $(x, y, z)$  is the three-dimensional (3D) position,  $d_{i,j}$  is the known distance constraint, and  $d_{i,j} \geq 0$ . The distance constraint model can be further written as follows:

$$\zeta_{i,j}(\mathbf{x}_k) = d_{i,j}^2 \quad (5)$$

$$\zeta_{i,j}(\mathbf{x}_k) = (\mathbf{D}_i \mathbf{x}_k^i - \mathbf{D}_j \mathbf{x}_k^j)^\top (\mathbf{D}_i \mathbf{x}_k^i - \mathbf{D}_j \mathbf{x}_k^j) \quad (6)$$

where  $\mathbf{D}_i$  is a matrix to extract the position from the  $i$ th sub-sensor state. If the system state of each sub-sensor is established by position and velocity, the matrix  $\mathbf{D}$  can be expressed as follows:

$$\mathbf{D}_i = \mathbf{D}_j = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

Without loss of generality, the multiple distance constraints in a multi-sensor system only exist between two adjacent sub-sensors. Thus, the distance constraint model of the multi-sensor system is written as follows:

$$\zeta(\mathbf{x}_k) = \mathbf{d} \quad (8)$$

$$\mathbf{d} \triangleq [d_{1,2}^2, \dots, d_{s-1,s}^2, d_{s,1}^2]^\top \quad (9)$$

$$\zeta(\mathbf{x}_k) \triangleq [\zeta_{1,2}(\mathbf{x}_k), \dots, \zeta_{s-1,s}(\mathbf{x}_k), \zeta_{s,1}(\mathbf{x}_k)]^\top \quad (10)$$

### III. CLASSICAL DISTANCE CONSTRAINT METHODS

In this section, we review the standard unconstrained filter methods, and subsequently introduce the pseudo-observation and quadratic constraint algorithms.

#### A. Unconstrained filters

We first summarized the state estimation without considering the distance constraint in the KF framework for the multi-sensor system (2). The standard KF can achieve optimal state estimation in a linear Gaussian system [17, 31]. Under the known initial state  $\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$  and its covariance matrix  $\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^\top]$ , KF can recursively estimate the state vector and covariance matrix through the following process.

Prediction:

$$\begin{cases} \hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1} \\ \mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^\top + \mathbf{Q}_{k-1} \end{cases} \quad (11)$$

Update:

$$\begin{cases} \mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \\ \mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \end{cases} \quad (12)$$

where  $\hat{\mathbf{x}}_k$  and  $\mathbf{P}_k$  are the state estimate and corresponding covariance matrix at time  $k$ , respectively;  $\hat{\mathbf{x}}_{k|k-1}$  and  $\mathbf{P}_{k|k-1}$

are the predicted state and corresponding covariance matrix, respectively; and  $\mathbf{K}_k$  is the gain matrix.

For a nonlinear Gaussian system, EKF, UKF, CKF, and PF are the classical filters. For nonlinear Gaussian systems in most positioning applications (e.g., wireless positioning and inertial integrated navigation), UKF has almost the same state estimation accuracy and algorithmic complexity as CKF. Compared to EKF, UKF has a significant advantage in state estimation accuracy. Furthermore, UKF can achieve approximately the same state estimation accuracy as the PF and has a significant computational advantage [22, 32]. Therefore, UKF was chosen to solve the nonlinear estimation problem for the multi-sensor system (2). UKF procedure is given as follows:

Prediction:

$$\begin{cases} \chi_{k-1} = \left\{ \hat{\mathbf{x}}_{k-1}, \hat{\mathbf{x}}_{k-1} \pm (\sqrt{(n+\lambda)\mathbf{P}_{k-1}})_j \right\} \\ \xi_{k|k-1} = \mathbf{f}(\chi_{k-1}); \quad \hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} \mathbf{W}_i^m \xi_{k|k-1}^i \\ \mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} \mathbf{W}_i^c (\chi_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1})(\diamond)^\top + \mathbf{Q}_{k-1} \end{cases} \quad (13)$$

Update:

$$\begin{cases} \gamma_{k|k-1} = \mathbf{h}(\chi_{k-1}); \quad \hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n} \mathbf{W}_i^m \gamma_{k|k-1}^i \\ \mathbf{P}_k^{zz} = \sum_{i=0}^{2n} \mathbf{W}_i^c (\gamma_{k|k-1}^i - \hat{\mathbf{z}}_{k|k-1})(\diamond)^\top + \mathbf{R}_k \\ \mathbf{P}_k^{xz} = \sum_{i=0}^{2n} \mathbf{W}_i^c (\chi_{k|k-1}^i - \hat{\mathbf{x}}_{k|k-1})(\gamma_{k|k-1}^i - \hat{\mathbf{z}}_{k|k-1})^\top \\ \mathbf{K}_k = \mathbf{P}_k^{xz} (\mathbf{P}_k^{zz})^{-1} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}) \\ \mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{k|k-1} \mathbf{K}_k^\top \end{cases} \quad (14)$$

where  $\chi_{k-1}$  is the sigma points;  $\xi_{k|k-1}$  and  $\gamma_{k|k-1}$  are the updated sample points related to the system state and measurement, respectively;  $\hat{\mathbf{x}}_{k|k-1}$  and  $\mathbf{P}_{k|k-1}^{zz}$  are the predicted state and corresponding covariance matrix, respectively;  $\hat{\mathbf{z}}_{k|k-1}$  and  $\mathbf{P}_k^{zz}$  are the predicted measurement and corresponding covariance matrix, respectively;  $\mathbf{P}_k^{xz}$  is the covariance matrix between the state and measurement;  $\mathbf{K}_k$  is the filter gain matrix;  $\hat{\mathbf{x}}_k$  and  $\mathbf{P}_k$  are the state estimate and corresponding covariance matrix, respectively;  $\lambda$  is a scaling parameter;  $\mathbf{W}_i^m$  and  $\mathbf{W}_i^c$  are the weight coefficients;  $(\mathbf{A})_j$  represents  $\mathbf{A}(1:n, j)$ ,  $j = 1, \dots, n$ ; and  $(\mathbf{A})(\diamond)^\top$  stands for  $(\mathbf{A})(\mathbf{A})^\top$ .

#### B. Pseudo-observation method

The pseudo-observation method aims to combine the distance constraint model (8) with the original measurement equation to form an augmented model [22, 28, 32]. Then, the distance constraint in the multi-sensor system (2) can be converted into a common nonlinear filter problem. The augmented measurement model is given as follows:

$$\tilde{\mathbf{z}}_k = \tilde{\mathbf{h}}(\mathbf{x}_k) + \tilde{\mathbf{v}}_k \quad (15)$$

$$\tilde{\mathbf{z}}_k \triangleq \begin{bmatrix} \mathbf{z}_k \\ \mathbf{d} \end{bmatrix}; \quad \tilde{\mathbf{h}}(\mathbf{x}_k) \triangleq \begin{bmatrix} \mathbf{h}(\mathbf{x}_k) \\ \zeta(\mathbf{x}_k) \end{bmatrix}; \quad \tilde{\mathbf{v}}_k \triangleq \begin{bmatrix} \mathbf{v}_k \\ \mathbf{0}_{s \times s} \end{bmatrix} \quad (16)$$

where the noise  $\tilde{\mathbf{v}}_k$  still is a zero-mean white Gaussian noise, and the corresponding covariance matrix  $\tilde{\mathbf{R}}_k = \text{cob}\{\mathbf{R}_k, \mathbf{0}_{s \times s}\}$ .

After the augmented measurement model is constructed, the nonlinear filter algorithm can implement the problem of distance constraint in the multi-sensor system. UKF can

handle this problem owing to its accuracy and computational complexity advantages.

However, the pseudo-observation algorithm has some shortcomings. As the number of sub-sensors increases, more distance constraint information becomes available, resulting in a sharp increase in the computational requirements of the pseudo-observation approach. UKF has a limited improvement in state estimation accuracy under distance constraints, while the higher accuracy nonlinear filter (e.g., PF) requires more complex algorithms and enormous computational effort.

### C. Quadratic constraint algorithm

If the system (2) contains two sub-sensors, the distance constraint model (8) can be reduced as follows:

$$\mathbf{x}_k^T \mathbf{M} \mathbf{x}_k - d_{1,2}^2 = 0 \quad (17)$$

$$\mathbf{x}_k = [(\mathbf{x}_k^1)^T \ (\mathbf{x}_k^2)^T]^T, \quad \mathbf{M} = [\mathbf{D}_1 \ -\mathbf{D}_2]^T [\mathbf{D}_1 \ -\mathbf{D}_2] \quad (18)$$

The state estimation of the distance constraint can be formulated as the following optimization problem:

$$\check{\mathbf{x}}_k = \arg \min_{\check{\mathbf{x}}_k} (\check{\mathbf{x}}_k - \hat{\mathbf{x}}_k)^T \mathbf{W}_k (\check{\mathbf{x}}_k - \hat{\mathbf{x}}_k) \quad (19)$$

$$\text{s.t.} \quad \check{\mathbf{x}}_k^T \mathbf{M} \check{\mathbf{x}}_k - d_{1,2}^2 = 0 \quad (20)$$

where  $\check{\mathbf{x}}_k$  is the constrained estimate,  $\hat{\mathbf{x}}_k$  is the state estimate of the unconstrained filter, and  $\mathbf{W}_k = \mathbf{P}_k^{-1}$  is a weighting matrix. By using the Lagrange technique [28], we obtained the following state estimate with the distance constraint:

$$\tilde{\mathbf{x}}_k = (\mathbf{W}_k + \lambda \mathbf{M})^{-1} (\mathbf{W}_k \hat{\mathbf{x}}_k) \quad (21)$$

where  $\mathbf{W}_k$  is a real symmetric positive definite matrix, which can be decomposed by the Cholesky decomposition as  $\mathbf{W}_k = \mathbf{G}^T \mathbf{G}$ . Moreover, the singular value decomposition (SVD) is used to decompose the matrix  $\mathbf{L} \mathbf{G}^{-1}$  as follows:

$$\mathbf{L} \mathbf{G}^{-1} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad (22)$$

where  $\mathbf{L} = [\mathbf{D}_1 \ -\mathbf{D}_2]$ ;  $\mathbf{U}$  and  $\mathbf{V}$  are unitary matrices,  $\mathbf{U} \mathbf{U}^T = \mathbf{I}$ ,  $\mathbf{V} \mathbf{V}^T = \mathbf{I}$ , and  $\mathbf{I}$  is identity matrix;  $\mathbf{\Sigma}$  stands for  $\text{diag}\{\sigma_1, \dots, \sigma_m\}$ . Thus, the state  $\tilde{\mathbf{x}}_k$  can be written as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_k &= (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{L}^T \mathbf{L})^{-1} (\mathbf{W}_k \hat{\mathbf{x}}_k) \\ &= \mathbf{G}^{-1} \mathbf{V} (\mathbf{I} + \lambda \mathbf{\Sigma}^T \mathbf{\Sigma})^{-1} (\mathbf{V}^T \mathbf{G}^{-T} \mathbf{W}_k \hat{\mathbf{x}}_k) \end{aligned} \quad (23)$$

Define  $\mathbf{e}_{\tilde{\mathbf{x}}} = \mathbf{V}^T \mathbf{G}^{-T} \mathbf{W}_k \hat{\mathbf{x}}_k = \{e_1, \dots, e_n\}$ , we have

$$\begin{aligned} \tilde{\mathbf{x}}_k^T \mathbf{M} \tilde{\mathbf{x}}_k &= \mathbf{e}_{\tilde{\mathbf{x}}}^T (\mathbf{I} + \lambda \mathbf{\Sigma}^T \mathbf{\Sigma})^{-1} (\mathbf{\Sigma}^T \mathbf{\Sigma}) (\mathbf{I} + \lambda \mathbf{\Sigma}^T \mathbf{\Sigma})^{-1} \mathbf{e}_{\tilde{\mathbf{x}}} \\ &= \sum_{i=1}^m \frac{\sigma_i^2 e_i^2}{(1 + \lambda \sigma_i^2)^2} \end{aligned} \quad (24)$$

By substituting (24) into (17), we have:

$$\sum_{i=1}^m \frac{\sigma_i^2 e_i^2}{(1 + \lambda \sigma_i^2)^2} - d_{1,2}^2 = 0 \quad (25)$$

where  $m=3$  for 3D positioning application.

Therefore, the distance constraint problem in the multi-sensor system with two sub-sensors is transformed into the solution of  $\lambda$ . We solved the analytical solution of the derived equation with one unknown variable  $\lambda$ , to obtain an accurate root. We can derive a maximum of six roots for the

3D positioning scenario. The optimal Lagrangian multiplier  $\lambda^*$  satisfies the solution with the minimum cost function  $(\tilde{\mathbf{x}}_k - \hat{\mathbf{x}}_k)^T \mathbf{W}_k (\tilde{\mathbf{x}}_k - \hat{\mathbf{x}}_k)$  among all real roots. Therefore, the constrained state estimate  $\check{\mathbf{x}}_k$  can be derived as follows:

$$\check{\mathbf{x}}_k = (\mathbf{W}_k + \lambda^* \mathbf{M})^{-1} (\mathbf{W}_k \hat{\mathbf{x}}_k) \quad (26)$$

### D. Covariance matrix update in quadratic constraint method

Covariance matrix estimation has the same priority as state estimation in practical applications. The constrained covariance matrix can be approximated as follows[20, 30]:

$$\check{\mathbf{P}}_k \approx \mathbf{\Delta} \mathbf{p} \mathbf{P}_k (\mathbf{\Delta} \mathbf{p})^T \quad (27)$$

where  $\mathbf{\Delta} \mathbf{p}$  is the Jacobi matrix of the projection function  $\mathbf{p}(\mathbf{x})$  with respect to  $\mathbf{x}$  around  $\hat{\mathbf{x}}_k$ , and is defined as follows:

$$\mathbf{p}(\mathbf{x}) = (\mathbf{W}_k + \lambda^* \mathbf{M})^{-1} (\mathbf{W}_k \mathbf{x}) = \mathbf{\Lambda}(\lambda^*) \mathbf{x} \quad (28)$$

$$\mathbf{\Delta} \mathbf{p} = \left. \frac{\partial \mathbf{p}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \quad (29)$$

Define  $\mathbf{\Lambda} = \mathbf{\Lambda}(\lambda^*)$ . Because  $\left. \frac{\partial \mathbf{x}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} = \mathbf{I}$ , we have

$$\mathbf{\Delta} \mathbf{p} = \left. \frac{\partial \mathbf{\Lambda}}{\partial \lambda^*} \frac{\partial \lambda^*}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \mathbf{x} + \mathbf{\Lambda}(\lambda^*) \quad (30)$$

Define  $\mathbf{\Pi}(\lambda^*) = (\mathbf{W}_k + \lambda^* \mathbf{M})^{-1}$ , we obtain

$$\frac{\partial \mathbf{\Lambda}}{\partial \lambda^*} = \frac{\partial (\mathbf{W}_k + \lambda^* \mathbf{M})^{-1}}{\partial \lambda^*} \mathbf{W}_k = -\mathbf{\Pi}(\lambda^*) \mathbf{M} \mathbf{\Lambda}(\lambda^*) \quad (31)$$

The optimal Lagrangian multiplier  $\lambda^*$  can be represented as the root of the nonlinear implicit function as follows:

$$\lambda^* \mapsto \left\{ \lambda \in \mathbb{R} \mid \mathbf{f}(\mathbf{\Lambda}(\lambda), \hat{\mathbf{x}}_k) = 0 \right\} \quad (32)$$

$$\mathbf{f}(\mathbf{\Lambda}(\lambda), \hat{\mathbf{x}}_k) = (\mathbf{\Lambda}(\lambda) \hat{\mathbf{x}}_k)^T \mathbf{M} \mathbf{\Lambda}(\lambda) \hat{\mathbf{x}}_k - d_{1,2}^2 \quad (33)$$

According to the Dini classical implicit function theorem [33], we derive the derivative of the function  $\lambda^*$  with respect to  $\mathbf{x}$  around  $\hat{\mathbf{x}}_k$  as follows:

$$\left. \frac{\partial \lambda^*}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} = - \left( \left. \frac{\partial \mathbf{f}}{\partial \lambda} \right|_{\lambda=\lambda^*} \right)^{-1} \left( \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \right) \quad (34)$$

where

$$\begin{aligned} \left. \frac{\partial \mathbf{f}}{\partial \lambda} \right|_{\lambda=\lambda^*} &= \hat{\mathbf{x}}_k^T \left( \left. \frac{\partial \mathbf{\Lambda}(\lambda)^T}{\partial \lambda} \right|_{\lambda=\lambda^*} \right) \mathbf{M} \mathbf{\Lambda}(\lambda^*) \hat{\mathbf{x}}_k \\ &\quad + \hat{\mathbf{x}}_k^T \mathbf{\Lambda}(\lambda^*)^T \mathbf{M} \left( \left. \frac{\partial \mathbf{\Lambda}(\lambda)}{\partial \lambda} \right|_{\lambda=\lambda^*} \right) \hat{\mathbf{x}}_k \\ &= -2 \mathbf{p}^T(\hat{\mathbf{x}}_k) \mathbf{M} \mathbf{\Pi}(\lambda^*) \mathbf{M} \mathbf{p}(\hat{\mathbf{x}}_k) \end{aligned} \quad (35)$$

$$\left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} = 2 \hat{\mathbf{x}}_k^T \mathbf{\Lambda}^T(\lambda^*) \mathbf{M} \mathbf{\Lambda}(\lambda^*) = 2 \mathbf{p}^T(\hat{\mathbf{x}}_k) \mathbf{M} \mathbf{\Lambda}(\lambda^*) \quad (36)$$

By substituting (35) and (36) into (34), we obtain

$$\left. \frac{\partial \lambda^*}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} = \frac{\mathbf{p}^T(\hat{\mathbf{x}}_k) \mathbf{M} \mathbf{\Lambda}(\lambda^*)}{\mathbf{p}^T(\hat{\mathbf{x}}_k) \mathbf{M} \mathbf{\Pi}(\lambda^*) \mathbf{M} \mathbf{p}(\hat{\mathbf{x}}_k)} \quad (37)$$

By substituting (31) and (37) into (30), we obtain

$$\mathbf{\Delta} \mathbf{p} = \left( \mathbf{I} - \frac{\mathbf{\Pi}(\lambda^*) \mathbf{M} \mathbf{p}(\hat{\mathbf{x}}_k) \mathbf{p}^T(\hat{\mathbf{x}}_k) \mathbf{M}}{\mathbf{p}^T(\hat{\mathbf{x}}_k) \mathbf{M} \mathbf{\Pi}(\lambda^*) \mathbf{M} \mathbf{p}(\hat{\mathbf{x}}_k)} \right) \mathbf{\Lambda}(\lambda^*) \quad (38)$$

The constrained covariance matrix can be obtained by substituting (38) into (27). Notably, the anomaly  $\mathbf{\Delta} \mathbf{p} \geq \mathbf{I}$



We obtained a series of alternative state estimates  $\check{\mathbf{x}}_{k,l}^*$  and corresponding covariance matrices through continuous iteration updates. The determination criterion for the final constrained state estimate  $\check{\mathbf{x}}_k$  is given as follows:

$$\check{\mathbf{x}}_k = \arg \min_{\check{\mathbf{x}}_{k,l}^*} (\check{\mathbf{x}}_{k,l}^* - \hat{\mathbf{x}}_k)^\top \mathbf{W}_k (\check{\mathbf{x}}_{k,l}^* - \hat{\mathbf{x}}_k) \quad (54)$$

We can conclude from (49) and (53) that the constrained state estimate  $\check{\mathbf{x}}_{k,l}^*$  and the corresponding covariance matrix  $\check{\mathbf{P}}_{k,l}$  are consistent in each iteration update process. The detailed implementation of one time step of the proposed algorithm for the distance constraint optimization is given in Algorithm 1.

Particularly, the proposed algorithm assumes that the state estimate generated by an unconstrained filter is close to the true state vector. If the unconstrained filter converges, the approximate system state estimate  $\hat{\mathbf{x}}_k$  and corresponding weight matrix  $\mathbf{W}_k$  can be obtained; hence, the assumption of the proposed algorithm is generally satisfied. Fortunately, in the practical application of multi-sensor positioning, the unconstrained filter is usually convergent rather than divergent, otherwise unconstrained filter is considered to be an abnormal working state.

---

**Algorithm 1** One time step of the proposed algorithm for distance constraint optimization in a multi-sensor system

---

**input:** unconstrained filter estimate by Section III.A:  $\hat{\mathbf{x}}_k$ ,  $\mathbf{P}_k$ ; the distance constraint vector  $\mathbf{d}$ ; and the maximum iterations  $l_m$ .  
**output:** constrained state estimate and covariance matrix:  $\check{\mathbf{x}}_k$ ,  $\check{\mathbf{P}}_k$   
1: set:  $\check{\mathbf{x}}_{k,0}^* = \hat{\mathbf{x}}_k$ ,  $\mathbf{W}_k = \mathbf{P}_k^{-1}$ ,  $l = 1$ .  
2: **while**  $l \leq l_m$  **do**  
3:   calculate  $\mathbf{A}_l$  and  $\mathbf{b}_l$  by  $\check{\mathbf{x}}_{k,l}^*$  and (50, 51).  
4:   calculate the iterative constrained state estimate  $\check{\mathbf{x}}_{k,l+1}^*$  by (49).  
5:   update  $l = l + 1$ .  
6: **end while**  
7: obtain  $\check{\mathbf{x}}_k$  by (54), and  $\check{\mathbf{P}}_k = (\mathbf{I} - \mathbf{J}_l) \mathbf{P}_k (\mathbf{I} - \mathbf{J}_l)^\top$  by (53).

---

We define the computational complexity  $\mathcal{O}$  as the number of basic mathematical operations (i.e., addition, subtraction, multiplication, and division). We used the most straightforward matrix multiplication and classic Gauss-Jordan elimination methods [35] to analyze the computational complexity of the algorithm. For an  $n \times m$  matrix  $\mathbf{A}$ , an  $m \times p$  matrix  $\mathbf{B}$ , and an  $n \times n$  matrix  $\mathbf{C}$ , we can obtain  $\mathcal{O}(\mathbf{AB}) = n(2m - 1)p$  and  $\mathcal{O}(\mathbf{C}^{-1}) = n^3$  using the classic Gauss-Jordan elimination.

We assume that the system state of each sub-sensor only contains the position and velocity vectors, and that its dimension is  $n$ . Therefore, the transfer matrix  $\mathbf{D}$  of each sub-sensor is the same, with the dimension of  $0.5n \times n$ . Then, we derive the computational complexity of the updated variables in the iterative process as follows:

$$\mathcal{O}(\mathbf{S}_{i,j}^{1,l}) = \mathcal{O}(\mathbf{S}_{i,j}^{2,l}) = 3n^2 + n, \quad \mathcal{O}(\mathbf{b}_{i,j}^l) = n^3 + 2n^2 + 7n - 1 \quad (55)$$

Hence,

$$\mathcal{O}(\mathbf{A}_l) = 2s(3n^2 + n), \quad \mathcal{O}(\mathbf{b}_l) = s(n^3 + 2n^2 + 7n - 1) \quad (56)$$

After updating  $\mathbf{A}_l$  and  $\mathbf{b}_l$ , we define  $\mathbf{C}_1 = \mathbf{A}_l^\top$ ,  $\mathbf{C}_2 = \mathbf{W}_k^{-1} \mathbf{C}_1$ ,  $\mathbf{C}_3 = \mathbf{A}_l \mathbf{C}_2$ ,  $\mathbf{C}_4 = \mathbf{C}_3^{-1}$ ,  $\mathbf{C}_5 = \mathbf{A}_l \hat{\mathbf{x}}_k - \mathbf{b}_l$ ,  $\mathcal{O}(\mathbf{C}_6) = \mathcal{O}(\mathbf{C}_4 \mathbf{C}_5)$ , and  $\mathcal{O}(\mathbf{C}_7) = \mathcal{O}(\mathbf{C}_2 \mathbf{C}_6)$ . Thus, we derive

$$\begin{cases} \mathcal{O}(\mathbf{C}_1) = s(sn); & \mathcal{O}(\mathbf{C}_2) = (sn)(2sn - 1)(s) \\ \mathcal{O}(\mathbf{C}_3) = s^2(2sn - 1); & \mathcal{O}(\mathbf{C}_4) = s^3 \\ \mathcal{O}(\mathbf{C}_5) = 2s(sn); & \mathcal{O}(\mathbf{C}_6) = s(2s - 1) \\ \mathcal{O}(\mathbf{C}_7) = sn(2s - 1) \end{cases} \quad (57)$$

Therefore, the computational complexity of the proposed iterative algorithm in one iteration in Eq. (49) is as follows:

$$\begin{aligned} \mathcal{O}(\check{\mathbf{x}}_{k,l}^*) &= \mathcal{O}(\mathbf{A}_l) + \mathcal{O}(\mathbf{b}_l) + \mathcal{O}(\mathbf{C}_1) + \mathcal{O}(\mathbf{C}_2) + \mathcal{O}(\mathbf{C}_3) \\ &\quad + \mathcal{O}(\mathbf{C}_4) + \mathcal{O}(\mathbf{C}_5) + \mathcal{O}(\mathbf{C}_6) + \mathcal{O}(\mathbf{C}_7) + (sn) \\ &= (2n^2 + 2n + 1)s^3 + (4n + 1)s^2 - s \end{aligned} \quad (58)$$

We have

$$\mathcal{O}(\mathbf{J}) = 2(n^2 + n)s^3 - (n^2 + n)s; \quad \mathcal{O}(\mathbf{I} - \mathbf{J}) = (sn)^2 \quad (59)$$

Because  $\mathcal{O}((\mathbf{I} - \mathbf{J})^\top) = \mathcal{O}(\mathbf{I} - \mathbf{J})$ , the computational complexity of the constrained covariance matrix (53) is given as follows:

$$\begin{aligned} \mathcal{O}(\check{\mathbf{P}}) &= \mathcal{O}(\mathbf{J}) + \mathcal{O}(\mathbf{I} - \mathbf{J}) + \mathcal{O}((\mathbf{I} - \mathbf{J})^\top) + 2(sn)^2(2sn - 1) \\ &= (4n^3 + 2n^2 + 2n)s^3 - (n^2 + n)s \end{aligned} \quad (60)$$

By setting the number of iterations  $l$  to a constant value, we express the computational complexity of the proposed iterative constraint algorithm as follows:

$$\mathcal{O}_{\text{Proposed}} = l \times \mathcal{O}(\check{\mathbf{x}}_{k,l}^*) + \mathcal{O}(\check{\mathbf{P}}) \quad (61)$$

For the unconstrained KF, we assume that the matrices  $\mathbf{F}_k \in \mathbb{R}^{sn \times sn}$ ,  $\mathbf{Q}_k \in \mathbb{R}^{sn \times sn}$ ,  $\mathbf{H}_k \in \mathbb{R}^{sm \times sn}$  and  $\mathbf{R}_k \in \mathbb{R}^{sm \times sm}$  in Eqs. (11, 12) are deterministic and time-invariant. Hence,

$$\begin{cases} \mathcal{O}(\hat{\mathbf{x}}_{k|k-1}) = sn(2sn - 1) \\ \mathcal{O}(\mathbf{P}_{k|k-1}) = 4(sn)^3 - (sn)^2 \\ \mathcal{O}(\mathbf{K}_k) = sn(2sn - 1)sm + 2sn(sm)^2 + (sm)^3 \\ \mathcal{O}(\hat{\mathbf{x}}_k) = 4(sn)(sm) \\ \mathcal{O}(\mathbf{P}_k) = (sn)^2(2sm + 2sn - 1) \end{cases} \quad (62)$$

Therefore, the computational complexity of the unconstrained KF algorithm is expressed as follows:

$$\begin{aligned} \mathcal{O}_{\text{KF}} &= \mathcal{O}(\hat{\mathbf{x}}_{k|k-1}) + \mathcal{O}(\mathbf{P}_{k|k-1}) + \mathcal{O}(\mathbf{K}_k) + \mathcal{O}(\hat{\mathbf{x}}_k) + \mathcal{O}(\mathbf{P}_k) \\ &= 6(sn)^3 + 4(sm)(sn)^2 + (2sm + 3m - 1)s + (sm)^3 \end{aligned} \quad (63)$$

We assume that the system state of each sub-sensor is the same, containing only position and velocity, and position as the system measurement vector. Thus,  $n = 4$  and  $m = 2$  for 2D positioning applications, and  $n = 6$  and  $m = 3$  for 3D applications. Figure 2 shows the ratio of the computational complexity between the proposed iterative constraint algorithm (61) and unconstrained KF (61). After performing the unconstrained filter by Section III. A, the computational complexity of the proposed algorithm is comparable to that of the unconstrained KF. Even for a multi-sensor system with ten sub-sensors, the computational complexity ratio between the proposed algorithm and KF does not exceed 1.6 in 2D localization applications.

Note that the complexity of the proposed algorithm does not strictly correspond to the actual run time. The algorithm should

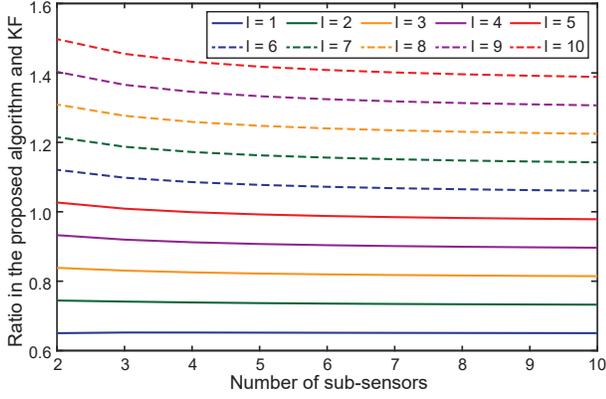


Fig. 2. Computational complexity ratio between the proposed algorithm (61) and KF (63) in 2D positioning applications,  $l$  is the number of iterations.

construct and update matrix  $\mathbf{A}_l$  and vector  $\mathbf{b}_l$  based on the last iterative state estimate, and perform the iteration termination discriminant function, which will consume a longer run time than the computational complexity. Furthermore, the storage space and run time required to update  $\mathbf{A}_l$  and  $\mathbf{b}_l$  with the number of sub-sensors may be multiples rather than a linear relationship in the multi-sensor system.

## V. NUMERICAL SIMULATION

### A. Simulation Setup

To verify the effectiveness of the proposed algorithm and demonstrate its performance, we simulated various experimental tests for 2D positioning applications. The state vector of each subsystem in the multi-sensor system is constructed by 2D position  $(x_k^i, y_k^i)$  and velocity  $(v_{x,k}^i, v_{y,k}^i)$ , that is,  $\mathbf{x}_k^i = [x_k^i \ y_k^i \ v_{x,k}^i \ v_{y,k}^i]^T$ ,  $i \in \{1, \dots, s\}$ . The state transition matrix  $\mathbf{F}_k^i$  and process noise matrix  $\mathbf{Q}_k^i$  of each subsystem [36] are given as follows:

$$\mathbf{F}_k^i = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \mathbf{Q}_k^i = q^i \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix} \quad (64)$$

where  $T = 1$  s is the sampling interval, and  $q^i$  is power spectral density (PSD) of the process noise  $\mathbf{w}_k^i$ .

We described linear and nonlinear measurement problems in three scenarios. The positioning performance of two and multiple sub-sensors was also considered. Scenario 1 is used to demonstrate the effectiveness and superiority of the proposed algorithm compared to other methods in terms of improving state estimation performance. Scenario 2 aims to verify the general applicability of the proposed algorithm in multi-sensor positioning systems with multiple sub-sensors. Additionally, Scenario 3 is used to demonstrate the effectiveness of the proposed algorithm for nonlinear positioning systems.

Using KF and UKF, the proposed constraint algorithm performed state estimation without considering distance constraints for the linear and nonlinear measurement problems, respectively. The proposed algorithm was compared with the following methods for the linear measurement system: 1)

unconstrained KF in Sec. III. A; 2) UKF-based pseudo-observation method in Sec. III. B (UPSE); 3) quadratic constraint algorithm in Sec. III. C and D (QUAD). In addition, we compared the performance of the following four algorithms in nonlinear measurement simulations: 1) the unconstrained UKF in Sec. III. A; 2) UPSE; 3) the proposed algorithm. All methods were programmed on MATLAB 2020b and executed on a computer with AMD Ryzen 7 4800H CPU @2.90 GHz and 16GB memory.

The cumulative distribution function (CDF) of the positioning error, the root mean squared error (RMSE), and the averaged RMSE (ARMSE) were selected as the metrics to evaluate the positioning performance. Moreover, the run time was also an indicator to evaluate the operating efficiency (or complexity) of the different algorithms. The positioning error  $\epsilon$ , RMSE, and ARMSE are defined as follows:

$$\begin{cases} \epsilon_k^s \triangleq \sqrt{(\hat{\mathbf{p}}_{x,k}^s - \mathbf{p}_{x,k}^s)^2 + (\hat{\mathbf{p}}_{y,k}^s - \mathbf{p}_{y,k}^s)^2} \\ \text{RMSE}_k \triangleq \sqrt{\frac{1}{n} \sum_{s=1}^n (\epsilon_k^s)^2} \\ \text{ARMSE} \triangleq \sqrt{\frac{1}{m \times n} \sum_{k=1}^m \sum_{s=1}^n (\epsilon_k^s)^2} \end{cases} \quad (65)$$

where  $(\hat{\mathbf{p}}_{x,k}^s, \hat{\mathbf{p}}_{y,k}^s)$  and  $(\mathbf{p}_{x,k}^s, \mathbf{p}_{y,k}^s)$  are the estimated and true states (i.e., position or velocity) at  $s$ th Monte Carlo run and time index  $k$ ,  $n = 1000$  is the number of Monte Carlo simulations, and  $m$  is the total number of time epochs.

### B. Scenario 1

In this simulation, we consider a multi-sensor positioning problem for a land vehicle. Two sensors were mounted on top of the vehicle, and their positions were measured using global positioning system techniques. The true movement trajectories of the two sensors are shown in Fig. 3. The vehicle maintained a constant velocity motion, and  $\|(v_{x,k}^i, v_{y,k}^i)\| = 5$  m/s. Furthermore, the distance between the two sensors was constant as the vehicle moved and is given by

$$d_{1,2} = 10 \text{ m} \quad (66)$$

The measurement matrices of the two sensors are given by

$$\mathbf{H}_k^1 = \mathbf{H}_k^2 = [\mathbf{I}_2 \ \mathbf{0}_2] \quad (67)$$

where  $\mathbf{I}_2$  and  $\mathbf{0}_2$  are the identity and zero matrices with  $2 \times 2$  dimension, respectively. The noise covariance matrices were set to  $\mathbf{R}_k^1 = \mathbf{R}_k^2 = \sigma^2 \mathbf{I}_2$ , and  $\sigma = 5$  m. The system process noise PSDs of the two sensors are set as  $q^1 = q^2 = 1 \text{ m}^2/\text{s}^3$ . The initial state  $\hat{\mathbf{x}}_0$  and covariance matrix  $\mathbf{P}_0^i$  were set as follows:

$$\hat{\mathbf{x}}_0^1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \hat{\mathbf{x}}_0^2 = \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \mathbf{P}_0^1 = \mathbf{P}_0^2 = \begin{bmatrix} 5^2 & 0 & 0 & 0 \\ 0 & 5^2 & 0 & 0 \\ 0 & 0 & 5^2 & 0 \\ 0 & 0 & 0 & 5^2 \end{bmatrix} \quad (68)$$

Figures 4 and 5 depict the CDF of the positioning error and RMSE of the position estimate using different algorithms through 1000 Monte Carlo simulations in Scenario 1, respectively. Table I summarizes the ARMSE of the position estimate, and run time of the four algorithms in 1000 Monte Carlo simulations. In addition, the system state uncertainty of the sub-sensor 1 in one simulation is shown in Fig. 6.

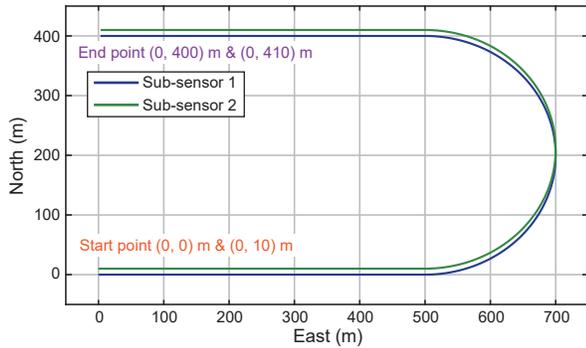


Fig. 3. True movement trajectory in multi-sensor positioning scenario 1.

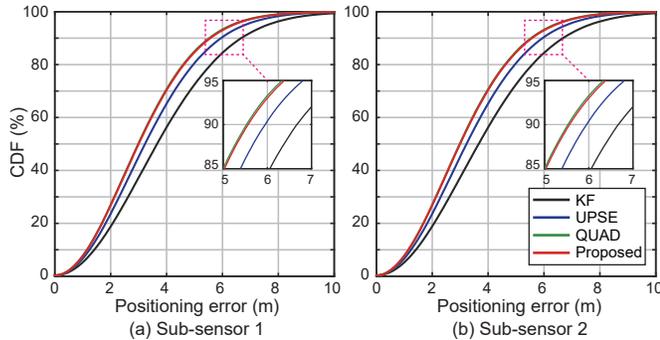


Fig. 4. CDF of positioning error through 1000 simulations in Scenario 1.

We can conclude from Figs. 4 and 5 that the distance constraint information significantly improves the positioning performance of unconstrained filters. The positioning accuracy of the proposed algorithm is significantly close to that of the QUAD method. As summarized in Tab. I, the average ARMSEs of the UPSE, QUAD, and the proposed algorithms were reduced by 11.2%, 17.5%, and 17.3%, respectively, compared to KF. The positioning accuracy difference between the proposed algorithm and QUAD was approximately 0.2%. In particular, the proposed algorithm has a significant runtime advantage over other distance constraint methods. Compared with UPSE and QUAD methods, the running time of the proposed algorithm for running 1000 Monte Carlo simulations is reduced by 43% and 60%, respectively.

TABLE I  
ARMSE AND RUN TIME OF DIFFERENT METHODS IN SCENARIO 1.

| ARMSE of position (m) | KF    | UPSE   | QUAD   | Proposed      |
|-----------------------|-------|--------|--------|---------------|
| Sub-sensor 1          | 4.394 | 3.903  | 3.626  | 3.635         |
| Sub-sensor 2          | 4.397 | 3.902  | 3.628  | 3.636         |
| Average               | 4.396 | 3.903  | 3.627  | <b>3.636</b>  |
| Run time (s)          | 3.433 | 49.731 | 71.037 | <b>28.290</b> |

As mentioned in the previous sections, covariance matrix estimation has the same priority as state estimation in practical applications. Fig. 6 shows the state uncertainty of sensor 1. The state uncertainty is defined as the square root of the diagonal of the covariance matrix. It can be seen that the state uncertainty obtained by the QUAD has some outliers because the state and covariance estimation are inconsistent. The abnormal phenomenon of the covariance matrix estimation does not occur in the proposed algorithm.

The probability density function (PDF) of the proposed

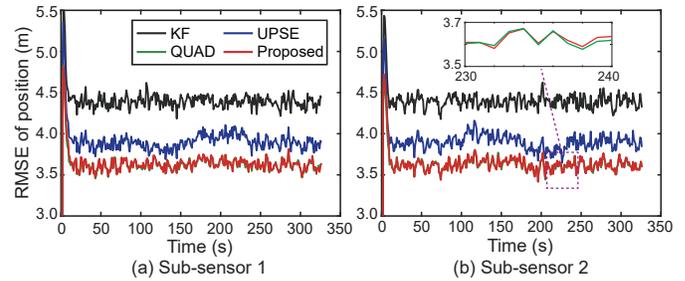


Fig. 5. RMSE of position estimate through 1000 simulations in Scenario 1.

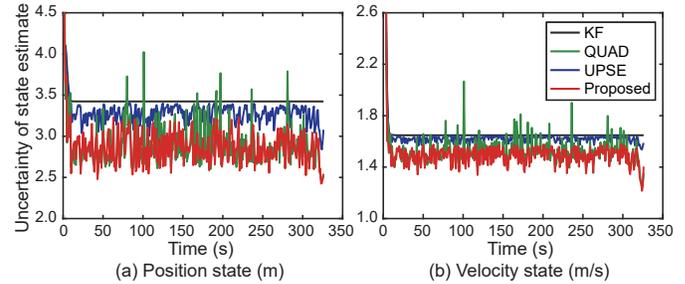


Fig. 6. State uncertainty of the sub-sensor 1 (1st simulation) in Scenario 1.

algorithm for different iteration numbers is shown in Fig. 7. Also, the RMSE of the position state using the proposed algorithm at different iterations is given in Fig. 7. It can be seen that the number of iterations of the proposed algorithm in single-epoch state estimation is distributed in the range of 1–4, accounting for more than 95%. In addition, the RMSE statistics of the estimated position state were very close when the iteration numbers were greater than or equal to 3.

The simulation results in positioning scenario 1 verify the effectiveness of the proposed algorithm and also demonstrate its advantages in terms of positioning performance, covariance matrix estimation, and running time, compared to the KF, UPSE, and QUAD methods.

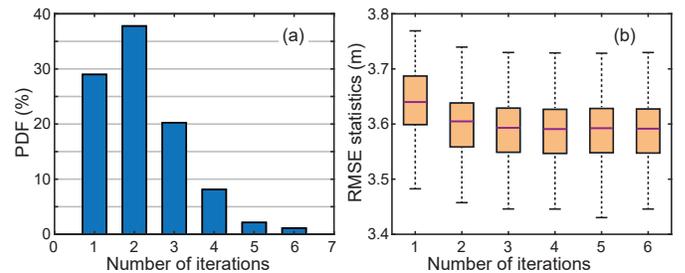


Fig. 7. Analysis of the proposed algorithm. (a) PDF of different iteration numbers. (b) RMSE statistics of the position estimate for different iterations.

### C. Scenario 2

In this simulation, we consider a multi-sensor positioning problem for a land vehicle with four sensors. Since the QUAD method cannot be extended to the multi-sensor positioning system with three or more sub-sensors, this simulation only compares the proposed algorithm with KF and UPSE methods. four sensors are mounted on a vehicle and their positions are measured using global positioning system techniques. As

shown in Fig. 8, the distances between the four sensors are constant as the vehicle moves and are given by

$$d_{1,2} = 6 \text{ m}; \quad d_{2,3} = 4 \text{ m}; \quad d_{3,4} = 6 \text{ m}; \quad d_{4,1} = 4 \text{ m} \quad (69)$$

The vehicle maintains a constant velocity motion, and  $\|(v_{x,k}^i, v_{y,k}^i)\| = 5 \text{ m/s}$ . The measurement matrices of the four sensors are given by

$$\mathbf{H}_k^1 = \mathbf{H}_k^2 = \mathbf{H}_k^3 = \mathbf{H}_k^4 = [\mathbf{I}_2 \quad \mathbf{0}_2] \quad (70)$$

where the noise covariance matrices are set as  $\mathbf{R}_k^1 = \mathbf{R}_k^2 = \mathbf{R}_k^3 = \mathbf{R}_k^4 = \sigma^2 \mathbf{I}_2$ , and  $\sigma = 5 \text{ m}$ . The system process noise PSDs of the four sub-sensors are set as  $q^1 = q^2 = q^3 = q^4 = 1 \text{ m}^2/\text{s}^3$ . The initial state  $\hat{\mathbf{x}}_0^i$  and covariance matrix  $\mathbf{P}_0^i$  are set as follows:

$$\left\{ \begin{array}{l} \hat{\mathbf{x}}_0^1 = [6 \quad 2 \quad 0 \quad 0]^T; \quad \hat{\mathbf{x}}_0^2 = [0 \quad 2 \quad 0 \quad 0]^T \\ \hat{\mathbf{x}}_0^3 = [0 \quad -2 \quad 0 \quad 0]^T; \quad \hat{\mathbf{x}}_0^4 = [6 \quad -2 \quad 0 \quad 0]^T \\ \mathbf{P}_0^1 = \mathbf{P}_0^2 = \mathbf{P}_0^3 = \mathbf{P}_0^4 = \begin{bmatrix} 5^2 & 0 & 0 & 0 \\ 0 & 5^2 & 0 & 0 \\ 0 & 0 & 5^2 & 0 \\ 0 & 0 & 0 & 5^2 \end{bmatrix} \end{array} \right. \quad (71)$$

Figures 9 and 10 show the RMSE of the position estimate and the velocity estimate using different algorithms, respectively. Table II summarizes the ARMSE of the position and velocity estimate, and run time of the different algorithms in 1000 Monte Carlo simulations.

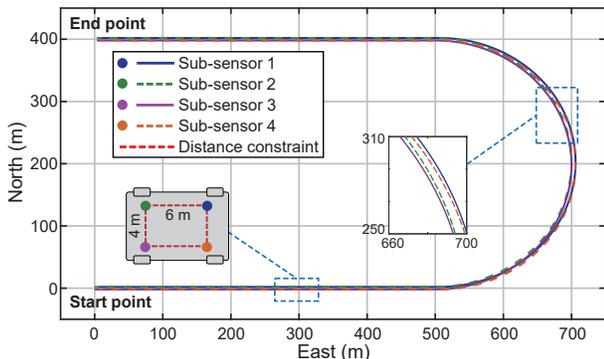


Fig. 8. True movement trajectory in multi-sensor positioning scenario 2.

TABLE II  
ARMSE AND RUN TIME OF DIFFERENT METHODS IN SCENARIO 2.

|                         | Sub-sensor | KF    | UPSE    | Proposed      |
|-------------------------|------------|-------|---------|---------------|
| ARMSE of position (m)   | 1          | 4.404 | 3.338   | 3.001         |
|                         | 2          | 4.399 | 3.371   | 3.019         |
|                         | 3          | 4.398 | 3.374   | 3.017         |
|                         | 4          | 4.413 | 3.336   | 3.011         |
|                         | Average    | 4.404 | 3.355   | <b>3.012</b>  |
| ARMSE of velocity (m/s) | 1          | 1.263 | 0.935   | 0.796         |
|                         | 2          | 1.261 | 0.931   | 0.804         |
|                         | 3          | 1.261 | 0.933   | 0.802         |
|                         | 4          | 1.264 | 0.932   | 0.795         |
|                         | Average    | 1.262 | 0.933   | <b>0.799</b>  |
| Run time (s)            |            | 4.971 | 133.457 | <b>55.197</b> |

As the distance constraint information increases in the multi-sensor system, the state estimation accuracy of the distance constraint algorithms also increases. Clearly, the proposed algorithm has higher accuracy in position and velocity

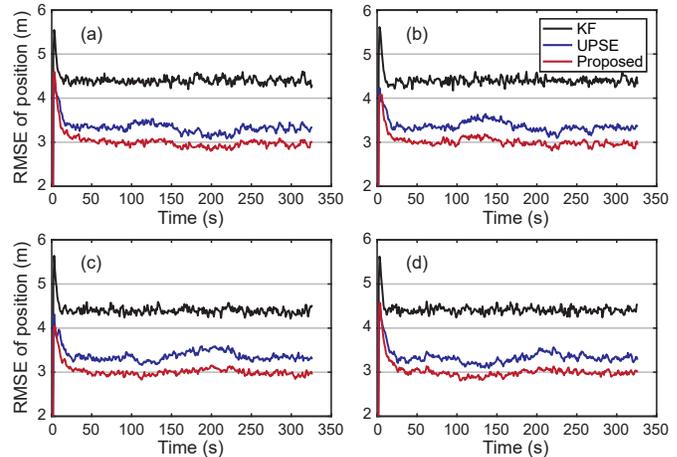


Fig. 9. RMSE of position estimate through 1000 simulations in Scenario 2. (a) Sub-sensor 1. (b) Sub-sensor 2. (c) Sub-sensor 3. (d) Sub-sensor 4.

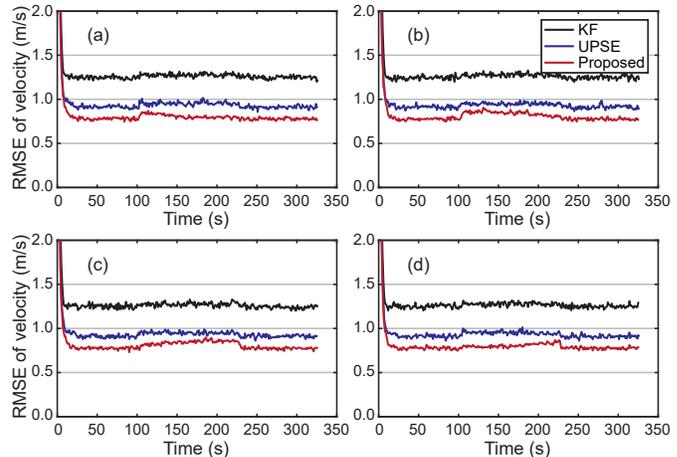


Fig. 10. RMSE of velocity estimate through 1000 simulations in Scenario 2. (a) Sub-sensor 1. (b) Sub-sensor 2. (c) Sub-sensor 3. (d) Sub-sensor 4.

estimation than KF and UPSE. The proposed algorithm reduces the average ARMSE in the position estimate by 31.6 % and 10.2 %, respectively; and the average ARMSE in the velocity estimate by 36.7 % and 14.4 %, respectively. Furthermore, the proposed algorithm has a significant operational efficiency advantage over the UPSE method. The run time of the proposed algorithm was reduced by 58.6 % compared to the UPSE method.

The effectiveness of the proposed algorithm and its advantages in terms of positioning performance and run time are further verified in Scenario 2.

#### D. Scenario 3

In this simulation, we consider a multi-sensor positioning problem for a mobile robot with four sensors. As shown in Figure 11, the four sensors are mounted on the mobile robot. Furthermore, the distance between the sensor and base station is measured by wireless positioning technology. The positions of the three base stations are  $(0, 0) \text{ m}$ ,  $(40, 0) \text{ m}$  and  $(20, 40) \text{ m}$ . Fig. 12 illustrates the measured distance between the four sub-sensors and the three base stations. The distance constraints between the four sensors are given by

$$d_{1,2} = 1 \text{ m}; \quad d_{2,3} = 1.5 \text{ m}; \quad d_{3,4} = 1 \text{ m}; \quad d_{4,1} = 1.5 \text{ m} \quad (72)$$

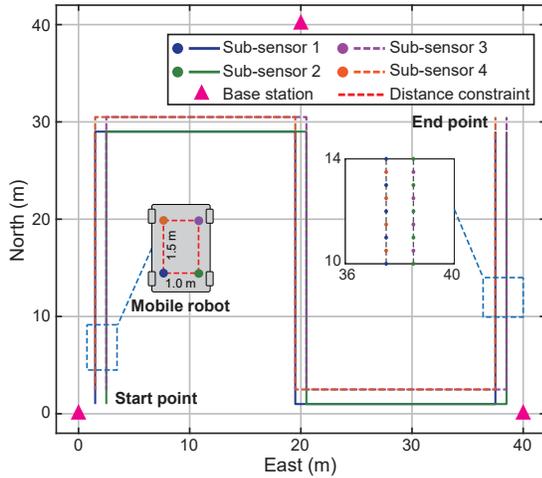


Fig. 11. True movement trajectory in multi-sensor positioning scenario 3.

The measurement model of each sub-sensor is non-linear rather than linear. The QUAD method cannot be extended to the multi-sensor positioning system with three or more sub-sensors. Therefore, the unconstrained UKF and UPSE are used as comparison methods to evaluate the proposed algorithm. The measurement function of each sensor is given as follows:

$$h_i(\mathbf{x}_k^i) = \begin{bmatrix} \sqrt{(x_k^i - r_{x,k}^1)^2 + (y_k^i - r_{y,k}^1)^2} \\ \sqrt{(x_k^i - r_{x,k}^2)^2 + (y_k^i - r_{y,k}^2)^2} \\ \sqrt{(x_k^i - r_{x,k}^3)^2 + (y_k^i - r_{y,k}^3)^2} \end{bmatrix} \quad (73)$$

where  $r_k^m$  is the position of the  $m$ th base station, and  $i \in \{1, 2, 3, 4\}$  represents the  $i$ th sub-sensor. The noise covariance matrices are set as  $\mathbf{R}_k^1 = \mathbf{R}_k^2 = \mathbf{R}_k^3 = \mathbf{R}_k^4 = \sigma^2 \mathbf{I}_2$ , and  $\sigma = 0.5$  m. The system process noise PSDs of the four sub-sensors are set as  $q^1 = q^2 = q^3 = q^4 = 0.1 \text{ m}^2/\text{s}^3$ . The initial state  $\hat{\mathbf{x}}_0^i$  and covariance matrix  $\mathbf{P}_0^i$  are set as follows:

$$\begin{cases} \hat{\mathbf{x}}_0^1 = [1.5 \ 1.0 \ 0 \ 0]^T; & \hat{\mathbf{x}}_0^2 = [2.5 \ 1.0 \ 0 \ 0]^T \\ \hat{\mathbf{x}}_0^3 = [2.5 \ 2.5 \ 0 \ 0]^T; & \hat{\mathbf{x}}_0^4 = [1.5 \ 2.5 \ 0 \ 0]^T \\ \mathbf{P}_0^1 = \mathbf{P}_0^2 = \mathbf{P}_0^3 = \mathbf{P}_0^4 = \begin{bmatrix} 1^2 & 0 & 0 & 0 \\ 0 & 1^2 & 0 & 0 \\ 0 & 0 & 1^2 & 0 \\ 0 & 0 & 0 & 1^2 \end{bmatrix} \end{cases} \quad (74)$$

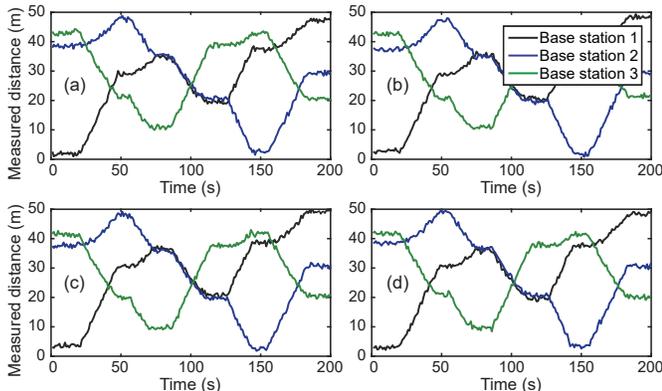


Fig. 12. Measured distance between the four sub-sensors and the three base stations in Scenario 3. (a) Sensor 1. (b) Sensor 2. (c) Sensor 3. (d) Sensor 4.

Figures 13 and 14 show the estimated trajectories and the CDF of the positioning error using different algorithms, respectively. Table III summarizes the RMS, maximum (MAX), and standard deviation (STD) of the positioning error statistics using different methods.

TABLE III  
POSITIONING ERROR STATISTICS OF DIFFERENT METHODS IN SCENARIO 3.

| Error (m) | Sub-sensor | UKF   | UPSE  | Proposed     |
|-----------|------------|-------|-------|--------------|
| RMS       | 1          | 0.516 | 0.411 | 0.333        |
|           | 2          | 0.511 | 0.423 | 0.343        |
|           | 3          | 0.522 | 0.407 | 0.343        |
|           | 4          | 0.518 | 0.397 | 0.334        |
|           | Average    | 0.517 | 0.410 | <b>0.338</b> |
| MAX       | 1          | 1.232 | 1.221 | 0.978        |
|           | 2          | 1.646 | 1.169 | 0.981        |
|           | 3          | 1.322 | 1.103 | 0.865        |
|           | 4          | 1.351 | 1.146 | 0.863        |
|           | Average    | 1.388 | 1.141 | <b>0.922</b> |
| STD       | 1          | 0.256 | 0.220 | 0.156        |
|           | 2          | 0.253 | 0.205 | 0.165        |
|           | 3          | 0.268 | 0.203 | 0.166        |
|           | 4          | 0.241 | 0.192 | 0.164        |
|           | Average    | 0.255 | 0.205 | <b>0.163</b> |

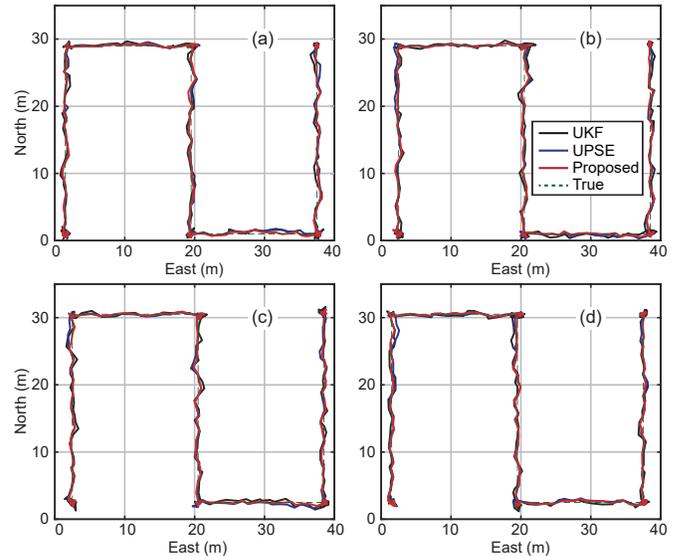


Fig. 13. Estimated trajectories using different methods in Scenario 3. (a) Sub-sensor 1. (b) Sub-sensor 2. (c) Sub-sensor 3. (d) Sub-sensor 4.

For the nonlinear measurement models, the distance constraint also improves the state estimation accuracy of the nonlinear filter algorithms. The proposed algorithm has higher positioning accuracy than UKF and UPSE and reduces the positioning error (average RMS) by 34.6 % and 17.8 %, respectively. Moreover, the proposed algorithm reduces the MAX and STD of the positioning error. For the nonlinear system, the effectiveness of the proposed algorithm in improving the positioning performance is demonstrated in Scenario 3. Further, the proposed algorithm can be implemented in most multi-sensor system applications. The simulation results also verify the universality of the proposed algorithm for multi-sensor positioning systems.

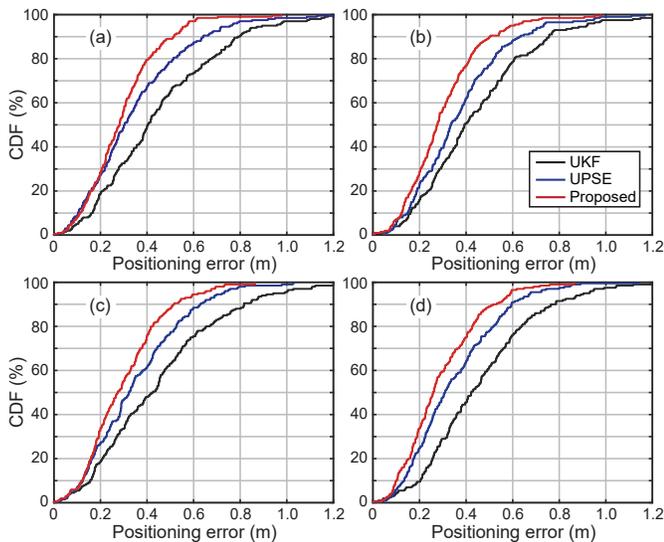


Fig. 14. CDF of the positioning error using different methods in Scenario 3. (a) Sub-sensor 1. (b) Sub-sensor 2. (c) Sub-sensor 3. (d) Sub-sensor 4.

## VI. CONCLUSION AND FUTURE WORK

This study proposed an iterative constraint algorithm to solve the distance constraint optimization problem of the multi-sensor system. The proposed algorithm was compared with the current representative of the UKF-based pseudo-observation and quadratic constraint methods using positioning simulations. The simulation results demonstrate the advantages of the proposed algorithm in terms of positioning performance, covariance matrix estimation, and operational efficiency (or computational complexity). The proposed algorithm is universally applicable to multi-sensor systems with multiple sub-sensors.

For future work, we will study the distance constraint optimization problem with noise, and then extend the proposed algorithm to distributed Kalman filter to adapt to a larger-scale multi-sensor system. Furthermore, we will apply the proposed algorithm to improve the positioning performance in real applications, such as the multi-sensor-based wheeled robot and pedestrian positioning system.

## REFERENCES

- [1] J. Rantakokko, J. Rydell, P. Strömbäck, P. Händel, J. Callmer, D. Törnqvist, F. Gustafsson, M. Jobs, and M. Grudén, "Accurate and reliable soldier and first responder indoor positioning: multi-sensor systems and cooperative localization," *IEEE Wireless Commun.*, vol. 18, no. 2, pp. 10–18, 2011.
- [2] Z. Zhou, Y. Li, J. Liu, and G. Li, "Equality constrained robust measurement fusion for adaptive Kalman-filter-based heterogeneous multi-sensor navigation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2146–2157, 2013.
- [3] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino, "Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges," *Inf. Fusion*, vol. 35, pp. 68–80, 2017.
- [4] Q. Zhang, Y. Hu, and X. Niu, "Required lever arm accuracy of non-holonomic constraint for land vehicle navigation," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8305–8316, 2020.
- [5] R. Zanetti, M. Majji, R. H. Bishop, and D. Mortari, "Norm-constrained Kalman filtering," *J. Guid. Control Dyn.*, vol. 32, no. 5, pp. 1458–1465, 2009.

- [6] J. R. Forbes, A. H. de Ruiter, and D. E. Zlotnik, "Continuous-time norm-constrained Kalman filtering," *Automatica*, vol. 50, no. 10, pp. 2546–2554, 2014.
- [7] B. Fasel, J. Spörri, J. Chardonens, J. Kröll, E. Müller, and K. Aminian, "Joint inertial sensor orientation drift reduction for highly dynamic movements," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 1, pp. 77–86, 2018.
- [8] L. Sy, M. Raitor, M. D. Rosario, H. Khamis, L. Kark, N. H. Lovell, and S. J. Redmond, "Estimating lower limb kinematics using a reduced wearable sensor count," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 4, pp. 1293–1304, 2021.
- [9] X. Niu, Y. Wu, and J. Kuang, "Wheel-ins: A wheel-mounted mems imu-based dead reckoning system," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 9814–9825, 2021.
- [10] Y. Wu, J. Kuang, and X. Niu, "Wheel-ins2: Multiple mems imu-based dead reckoning system with different configurations for wheeled robots," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3064–3077, 2023.
- [11] J. Li, X. Liu, Z. Wang, H. Zhao, T. Zhang, S. Qiu, X. Zhou, H. Cai, R. Ni, and A. Cangelosi, "Real-time human motion capture based on wearable inertial sensor networks," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8953–8966, 2022.
- [12] J. Li, Z. Wang, S. Qiu, H. Zhao, J. Wang, X. Shi, L. Liu, and N. Yang, "Study on horse-rider interaction based on body sensor network in competitive equestrian," *IEEE Trans. Affect. Comput.*, vol. 13, no. 1, pp. 553–567, 2022.
- [13] B. Oubre, J.-F. Daneault, K. Boyer, J. H. Kim, M. Jasim, P. Bonato, and S. I. Lee, "A simple low-cost wearable sensor for long-term ambulatory monitoring of knee joint kinematics," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 12, pp. 3483–3490, 2020.
- [14] L. S. Vargas-Valencia, F. B. A. Schneider, A. G. Leal-Junior, P. Caicedo-Rodríguez, W. A. Sierra-Arévalo, L. E. Rodríguez-Cheu, T. Bastos-Filho, and A. Frizzera-Neto, "Sleeve for knee angle monitoring: An imu-pof sensor fusion system," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 2, pp. 465–474, 2021.
- [15] T. R. Bennett, J. Wu, N. Kehtarnavaz, and R. Jafari, "Inertial measurement unit-based wearable computers for assisted living applications: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 28–35, 2016.
- [16] A. T. Alouani and W. D. Blair, "Use of a kinematic constraint in tracking constant speed, maneuvering targets," *IEEE Trans. Autom. Control*, vol. 38, no. 7, pp. 1107–1111, 1993.
- [17] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *IET Control Theory Appl.*, vol. 4, no. 8, pp. 1303–1318, 2010.
- [18] E. K. Babacan, L. Ozbek, and M. Efe, "Stability of the extended kalman filter when the states are constrained," *IEEE Trans. Automat. Control*, vol. 53, no. 11, pp. 2707–2711, 2008.
- [19] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
- [20] S. J. Julier and J. J. LaViola, "On Kalman filtering with nonlinear equality constraints," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2774–2784, 2007.
- [21] A. N. Vargas, H. M. T. Menegaz, J. Y. Ishihara, and L. Acho, "Unscented kalman filters for estimating the position of an automotive electronic throttle valve," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 4627–4632, 2016.
- [22] G. Zhou, K. Li, and T. Kirubarajan, "Constrained state estimation using noisy destination information," *Signal Process.*, vol. 166, p. 107226, 2020.
- [23] I. Arasaratnam and S. Haykin, "Cubature kalman filters," *IEEE Trans. Automat. Control*, vol. 54, no. 6, pp. 1254–1269, 2009.
- [24] B. Cui, X. Wei, X. Chen, J. Li, and L. Li, "On sigma-point update of cubature kalman filter for gnss/ins under gnss-challenged environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8671–8682, 2019.
- [25] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian

- bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.
- [26] G. M. Hoffmann and C. J. Tomlin, “Mobile sensor network control using mutual information methods and particle filters,” *IEEE Trans. Automat. Control*, vol. 55, no. 1, pp. 32–47, 2010.
- [27] G. Raja, S. Suresh, S. Anbalagan, A. Ganapathisubramaniyan, and N. Kumar, “Pfin: An efficient particle filter-based indoor navigation framework for uavs,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 4984–4992, 2021.
- [28] C. Yang and E. Blasch, “Kalman filtering with nonlinear state constraints,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 1, pp. 70–84, 2009.
- [29] D. Wang, M. Li, X. Huang, and J. Li, “Kalman filtering for a quadratic form state equality constraint,” *J. Guid. Control Dyn.*, vol. 37, no. 3, pp. 951–957, 2014.
- [30] I. Skog, J.-O. Nilsson, D. Zachariah, and P. Händel, “Fusing the information from two navigation systems using an upper bound on their maximum spatial separation,” in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, 2012, pp. 1–5.
- [31] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems, 2nd ed.* Boca Raton, FL, USA: CRC Press LLC, 2012.
- [32] G. Zhou, K. Li, X. Chen, L. Wu, and T. Kirubarajan, “State estimation with a destination constraint using pseudo-measurements,” *Signal Process.*, vol. 145, pp. 155–166, 2018.
- [33] A. L. Dontchev and R. T. Rockafellar, *Implicit functions and solution mappings, 2nd ed.* New York, NY, USA: Springer, 2009.
- [34] D. Simon and T. L. Chia, “Kalman filtering with state equality constraints,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 1, pp. 128–136, 2002.
- [35] F. L. Gall, “Powers of tensors and fast matrix multiplication,” in *Proc. 39th Int. Symp. Symbolic Algebraic Comput.*, 2014, pp. 296–303.
- [36] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software.* New York, NY, USA: Wiley-Interscience, 2001.