New Closed-form Joint Localization and Synchronization using Sequential TOAs in a Multi-agent System

Ningyan Guo, Sihao Zhao, Xiao-Ping Zhang, Fellow, IEEE, Zheng Yao, Xiaowei Cui, and Mingquan Lu

Abstract—In a multi-agent system (MAS) comprised of parent nodes (PNs) and child nodes (CNs), a relative spatiotemporal coordinate is established by the PNs with known positions. It is an essential technique for the moving CNs to resolve the joint localization and synchronization (JLAS) problem in the MAS. Existing methods using sequential time-of-arrival (TOA) measurements from the PNs’ broadcast signals either require a good initial guess or have high computational complexity. In this paper, we propose a new closed-form JLAS approach, namely CFJLAS, which achieves optimal solution without initialization, and has low computational complexity. We first linearize the relation between the estimated parameter and the sequential TOA measurement by squaring and differencing the TOA measurement equations. By devising two intermediate variables, we are able to simplify the problem to finding the solution of a quadratic equation set. Finally, we apply a weighted least squares (WLS) step using the residuals of all the measurements to optimize the estimation. We derive the Cramér-Rao lower bound (CRLB), analyze the estimation error, and show that the estimation accuracy of the CFJLAS reaches CRLB under small noise condition. The complexity of the CFJLAS is studied and compared with the iterative method. Simulations in the 2D scene verify that the estimation accuracy of the new CFJLAS method in position, velocity, clock offset, and clock skew all reaches CRLB. Compared with the conventional iterative method, which requires a good initial guess to converge to the correct estimation and has growing complexity with more iterations, the proposed new CFJLAS method does not require initialization, always obtains the optimal solution and has constant low computational complexity.

Index Terms—sequential time-of-arrival (TOA), joint localization and synchronization (JLAS), closed-form method, multi-agent system (MAS).

I. INTRODUCTION

Ağ intelligent multi-agent system (MAS) utilizes a large group of small, inexpensive moving nodes, namely parent nodes (PNs) and child nodes (CNs), to perform cooperated operations and movements in harsh environments [1]. Compared with a single node’s limited capability, an MAS performs better in scalability, robustness and strong adaptability in harsh environments. Hence, an MAS presents prospectively wider applications, such as emergency rescue, environmental exploration, planet exploration, aerial surveillance, and target detection and track [2]–[5].

To enable the navigation and control for CNs in the absence of external spatiotemporal coordinate, joint localization and synchronization (JLAS) in a relative spatiotemporal coordinate is an important enabling technology. To achieve JLAS for CNs, PNs transmit signals at known places and the CNs capture the signals to obtain measurements, such as time-of-arrival (TOA) and time-of-flight (TOF). The TOF measurements require perfect synchronization between the PN and CN, which is costly to implement. The TOA measurements do not require synchronization between the PN and CN, and is one of the most widely adopted measurements for JLAS due to its simplicity and high accuracy [6]–[11].

Code division and frequency division are among the most widely used TOA schemes. Two typical examples of these two schemes are the Global Positioning System (GPS), which adopts the code division scheme and the GLONASS, which uses the frequency division [12], [13]. However, the code division scheme suffers from the near-far effect especially in a small region where a CN is near the PN [14], and the frequency division scheme occupies a relative wide frequency bandwidth [15] that increases the complexity of the radio frequency front-end design, especially for a small and low-cost CN.

Unlike code division or frequency division schemes, a time division (TD) MAS generating sequential TOA measurements is free of the near-far effect or large bandwidth occupancy. A CN in a TD MAS obtains sequential one-way TOA measurements to achieve JLAS on its own side, without any transmission back to the PNs. Such a TD broadcast scheme reduces the reception channels of a CN, supports an unlimited number of CNs, and offers higher safety for CNs from being detected. Hence, more and more research works on the TD scheme have appeared in recent years [16]–[24].

Localization based on TD broadcast sequential TOA has been studied in recent years. Dwivedi et al. [18] develop a solution for cooperative localization by using scheduled wireless transmissions. They formulate two practical estimators, namely sequential WLS-MDS and maximum likelihood (ML), for localization and derive the Cramér-Rao lower bound (CRLB) to evaluate the performance of these two estimators. Zachariah et al. [20] propose a self-localization method for a receiver node using given sequential measurements based on an iterative maximum a posterior (MAP) estimator that
considers the uncertainty of the anchor nodes and the turn-around delay of the asynchronous transceivers. They also propose a sequential approximate maximum a posterior estimator to resolve the localization problem based on sequential TOAs [25]. Carroll et al. [21] use a sequential transmission protocol to resolve an on-demand asynchronous localization problem for an underwater acoustic network, and verify that passive nodes can achieve low-error positioning through simulation. Yan et al. [22] investigate a localization method for passive nodes based on sequential messages. Simulation results show that the proposed iterative least squares (LS) method can improve the localization accuracy compared with the synchronous localization algorithm. However, all these methods only resolve the node’s localization problem and do not address the clock synchronization issue.

For JLAS using sequential TOA measurements in the MAS, several iterative methods are proposed. Shi et al. [17] present a distributed state estimation method in the CNs, which utilize the TD broadcast inter-node information to resolve JLAS problem. Their method enables accurate, high-frequency and real-time estimation and support an unlimited number of nodes. However, their iterative method has high computational complexity and does not apply to a moving CN. Shi et al. [24] extend the previous work of [17], and utilize a two-step weighted least squares (WLS) method to jointly estimate the position, velocity and clock parameters of the CNs in the presence of the uncertainties of parent nodes. However, this method requires re-parameterize the localization problem into a higher-dimensional state space, and thus more measurements are needed, and also has high computational complexity due to the second iterative step. Zhao et al. [23] develop maximum likelihood methods and iterative algorithms to estimate the localization and synchronization parameters. However, this method requires good initial guess and has a high computational cost due to iteration.

Unlike the iterative method mentioned above, the closed-form JLAS estimator requires no initial guess and has low and constant computational complexity. There are extensive studies on closed-form methods for concurrent TOA system. Bancroft originally introduces the closed-form approach in solving the Global Positioning System (GPS) equations [26]. Compared with iterative methods, Bancroft’s closed-form algorithm significantly reduces computational complexity without the initial solution guess. A few extensions of Bancroft’s algorithm to the case with noisy time-difference-of-arrival (TDOA) measurements are proposed in [27], [28]. Chan and Ho [29] propose a TSWLS estimator that achieves CRLB at small noise level. Closed-form localization methods based on the multidimensional scaling technique that utilizes a squared distance matrix are proposed in [30], [31]. Zhu and Ding [32] generalize Bancroft’s algorithm to the case with more sensors and noisy measurements. Other approximately efficient and closed-form methods are developed in [33], [34] for joint synchronization and source localization task. Wang et al. [35] use a closed-form method to resolve the JLAS problem when the known sensor positions and clock biases are subject to random errors. Zhao et al. [36] propose a closed-form localization method using TOA measurements from two non-synchronized systems. However, the above studies all assume the TOA measurements are concurrent and thus are not applicable to the MAS with broadcast sequential TOA messages.

In this paper, we propose a new closed-form method, namely CFJLAS, to resolve the JLAS problem of the dense moving CNs using sequential TOA measurements in a wireless broadcast MAS. We first convert the nonlinear problem into a linear one by squaring and differencing the sequential TOA measurement equations. We devise two intermediate variables to linearly express the parameters to be estimated, and a quadratic equation set with respect to the intermediate variables is constructed. Then, we find the solution of the intermediate variables by solving the quadratic equations analytically. We then apply the least squares (LS) method to roughly estimate positions and the clock parameters of the moving CNs from the results of the intermediate variables. Finally, we apply a weighted least squares (WLS) step using the residuals of all the measurements to obtain the final optimal estimation. We derive the CRLB of the estimation error for the moving CN and show that the estimated results from the new CFJLAS reach CRLB when the measurement noise is small. Simulation of a 2D moving scene is conducted to verify the performance of the new CFJLAS. Numerical results show that the accuracies for the position, velocity, clock offset and clock skew from the proposed CFJLAS all reach the CRLBs. Compared with the iterative method, which requires accurate initialization and has increasing complexity with the number of iterations, the new method always gives the correct results without requiring an initial guess, and has a constant low computational complexity.

The rest of the paper is organized as follows. In Section II, the sequential TOA-based JLAS problem model for the moving CNs is formulated. A new closed-form JLAS method is proposed in detail in Section III. Then, we derive the CRLB, analyze the theoretical error and complexity of the new method in Section IV. Simulations are conducted to evaluate the performance of this new CFJLAS method in Section V. Finally, Section VI concludes the paper.

Main notations used in this paper are summarized in Table I.

II. PROBLEM STATEMENT

We consider a distributed MAS comprised of M PNs with known positions and a number of moving CNs, as depicted in Fig. 1. It operates in a wireless broadcast manner using a TDMA scheme. All PNs are synchronized to a system clock. They sequentially broadcast the on-air packets according to its pre-scheduled launch time slots, e.g., PN #1, PN #2 and so on. PNs set up a relative spatiotemporal coordinate in advance. CNs receive the on-air packets to obtain the sequential TOA measurements and achieve JLAS in this relative spatiotemporal coordinate. Therefore, we aim to resolve the moving CNs’ JLAS problem based on the observed sequential TOA measurements in a wireless broadcast network.

Without loss of generality, we take one moving CN as an example to present the JLAS problem. Other moving CNs in
We model that the moving CN’s velocity remains constant in the communication rate is high enough, one TDMA round of the moving CN. The sequential TOA measurements in one TDMA round from all PNs are short, usually at millisecond level. We aim to find an accurate estimation of the moving CN’s position and clock parameters using all the sequential TOA measurements given by (2). To solve this problem, the parameters \( p, v, \beta, \omega \) need to be estimated. Existing studies use the iterative method, such as [23], to solve this estimation problem. It requires an accurate initial guess to obtain good estimation and has high computational complexity, which grows with the increasing number of iterations. We will propose a new distributed closed-form method to resolve this problem in the next section.

### III. A New Closed-form JLAS Method

In this section, we develop a new closed-form JLAS method, namely CFJLAS. First, we linearize the nonlinear problem. By employing two intermediate variables, we reduce the problem to solving a quadratic equation set. Then, we find the solution of the intermediate variables, based on which a raw estimation of the moving CN’s position and clock parameter is obtained. Finally, we apply a WLS step to obtain the final optimal estimation. The above steps are presented in detail in the following sub-sections.

#### A. Linearization

The sequential TOA measurement given by (2) is nonlinear. To convert it to a linear relation, we re-organize and square the two sides of (2), and then come to

\[
(\hat{\tau}_i + \beta_i - \beta - \omega t_i - \varepsilon_i)^2 = \|p + vt_i - p_i\|^2
\]

where all the variables have the same unit of meter.

We propose a new distributed closed-form method to resolve this problem in the next section.
We define $\hat{\alpha}_i \triangleq \tau_i + \beta_i$, expand and re-organize (3), and come to

\[
2p_i T p + 2t_i p_i T v - 2\hat{\alpha}_i \beta - 2t_i \hat{\alpha}_i \omega + (\beta^2 - ||p||^2)
\]

\[
= ||p_i||^2 - \hat{\alpha}_i^2 - t_i^2 (\omega^2 - ||v||^2) - 2t_i (\beta \omega - p^T v) + \eta_i + \varepsilon_i^2,
\]

where $\eta_i = 2(\hat{\alpha}_i - \beta - \omega t_i) \varepsilon_i$.

To remove the term $\beta^2 - ||p||^2$, we can conduct subtraction between equation (4) with arbitrary i and equations (4) with other i values. Without loss of generality, we substitute $i = 1$ into (4). We ignore the squared error term $\varepsilon_i^2$ under the small noise condition and obtain

\[
2p_1^T p + 2t_1 p_1^T v - 2\hat{\alpha}_1 \beta - 2t_1 \hat{\alpha}_1 \omega + (\beta^2 - ||p||^2)
\]

\[
= ||p_1||^2 - \hat{\alpha}_1^2 - t_1^2 (\omega^2 - ||v||^2) - 2t_1 (\beta \omega - p^T v) + \eta_1,
\]

(5)

By subtracting (5) from (4), we remove the term $\beta^2 - ||p||^2$. We put all the linear unknown parameters on the left and have

\[
2 (p_i^T - p_1^T) p + 2 (t_i p_i^T - t_1 p_1^T) v + 2 (\hat{\alpha}_1 - \hat{\alpha}_i) \beta + 2 (t_1 \hat{\alpha}_1 - t_i \hat{\alpha}_i) \omega
\]

\[
= ||p_i||^2 - ||p_1||^2 - (\hat{\alpha}_i^2 - \hat{\alpha}_1^2) + (t_i^2 - t_1^2) (\omega^2 - ||v||^2) + 2 (t_1 - t_i) (\beta \omega - p^T v) + \eta_i - \eta_1,
\]

\[
i = 2, \cdots, M,
\]

(6)

We devise two intermediate variable as

\[
\lambda_1 = \omega^2 - ||v||^2,
\]

\[
\lambda_2 = \beta \omega - p^T v.
\]

(7)

With these two intermediate variables, we are able to simplify the problem to the task of finding the solution of the two intermediate variables. This new problem can be solved analytically as will be shown in the next subsection.

We denote the unknown parameters by $\hat{\theta} = [p^T, v^T, \beta, \omega]^T$, and then rewrite (6) into the collective form as

\[
A \hat{\theta} = y + G [\lambda_1, \lambda_2]^T + \eta_d,
\]

(8)

where

\[
A =
\begin{bmatrix}
p_i^T - p_1^T & t_2 p_i^T - t_1 p_1^T & \hat{\alpha}_1 - \hat{\alpha}_i & t_1 \hat{\alpha}_1 - t_2 \hat{\alpha}_2 \\
p_i^T - p_1^T & t_2 p_i^T - t_1 p_1^T & \hat{\alpha}_1 - \hat{\alpha}_i & t_1 \hat{\alpha}_1 - t_2 \hat{\alpha}_2 \\
& \vdots & \vdots & \vdots \\
p_M^T - p_1^T & t_2 p_M^T - t_1 p_1^T & \hat{\alpha}_1 - \hat{\alpha}_M & t_1 \hat{\alpha}_1 - t_2 \hat{\alpha}_M
\end{bmatrix},
\]

(9)

\[
G = \begin{bmatrix}
(t_1^2 - t_2^2) 2(t_1 - t_2) \\
\vdots \\
(t_1^2 - t_M^2) 2(t_1 - t_M)
\end{bmatrix},
\]

(10)

\[
y = \left[ ||p_2||^2 - ||p_1||^2 - (\hat{\alpha}_2^2 - \hat{\alpha}_1^2) \right] \\
\vdots \\
[||p_M||^2 - ||p_1||^2 - (\hat{\alpha}_M^2 - \hat{\alpha}_1^2)]
\]

\[
\eta_d = [\eta_2 - \eta_1, \cdots, \eta_M - \eta_1]^T.
\]

We obtain the linear relation of the moving CN’s unknown parameter $\theta$ and the two intermediate variables $\lambda_1$ and $\lambda_2$ as given by (8). $A$, $y$ and $G$ are known, and $\eta_d$ is treated as the noise vector. If we can find the solution of these two intermediate variables, the position, velocity, clock offset, and clock skew of the moving CN can be estimated using this linear relation (8).

**B. Solution for Intermediate Variables**

By observing (8), under the condition of small noise, we can roughly estimate $\hat{\theta}$ with the intermediate variables $\lambda_1$ and $\lambda_2$ by applying an LS method as given by

\[
\hat{\theta} = (A^T A)^{-1} A^T \left( y + G [\lambda_1, \lambda_2]^T \right),
\]

(13)

or simply

\[
\hat{\theta} = g + U [\lambda_1, \lambda_2]^T,
\]

(14)

where $\hat{\theta} = [\hat{p}^T, \hat{v}^T, \hat{\beta}, \hat{\omega}]^T$ is the raw estimate of the parameter $\theta$, and

\[
g = (A^T A)^{-1} A^T y,
\]

(15)

\[
U = (A^T A)^{-1} A^T G.
\]

(16)

To find the solution of the intermediate variables $\lambda_1$ and $\lambda_2$, we design the following two matrices

\[
H_1 = \begin{bmatrix}
O_{K \times K} & O_{K \times K} & O_{K \times 2} \\
O_{K \times K} & -I_K & O_{K \times 2} \\
O_{2 \times K} & O_{2 \times K} & 0
\end{bmatrix},
\]

(17)

\[
H_2 = \begin{bmatrix}
O_{K \times K} & -I_K & O_{K \times 2} \\
-I_K & O_{K \times K} & O_{K \times 2} \\
O_{2 \times K} & O_{2 \times K} & 1
\end{bmatrix}.
\]

(18)

To construct equations with respect to the intermediate variables, we use these two matrices $H_1$ and $H_2$ along with (7) to convert the raw estimate parameter $\hat{\theta}$ to the linear expression of $\lambda_1$ and $\lambda_2$ as

\[
\hat{\theta}^T H_1 \hat{\theta} = (g + U [\lambda_1, \lambda_2])^T H_1 (g + U [\lambda_1, \lambda_2])
\]

\[
= \omega^2 - ||v||^2 \approx \lambda_1,
\]

(19)

\[
\hat{\theta}^T H_2 \hat{\theta} = (g + U [\lambda_1, \lambda_2])^T H_2 (g + U [\lambda_1, \lambda_2])
\]

\[
= 2 (\hat{\beta} \omega - \hat{p}^T \hat{v}) \approx 2 \lambda_2.
\]

(20)

After re-organizing (19) and (20), we obtain two quadratic equations with respect to $\lambda_1$ and $\lambda_2$ as

\[
a_1 \lambda_1^2 + b_1 \lambda_1 \lambda_2 + c_1 \lambda_2^2 + d_1 \lambda_1 + e_1 \lambda_2 + f_1 = 0,
\]

(21)

\[
a_2 \lambda_1^2 + b_2 \lambda_1 \lambda_2 + c_2 \lambda_2^2 + d_2 \lambda_1 + e_2 \lambda_2 + f_2 = 0,
\]

(22)

where

\[
a_1 = [U]_{:,1}^T H_1 [U]_{:,1}, \quad b_1 = 2[U]_{:,1}^T H_1 g - 1,
\]

(23)

\[
c_1 = [U]_{:,2}^T H_1 [U]_{:,2}, \quad d_1 = 2[U]_{:,1} H_1 g - 1,
\]

\[
e_1 = 2[U]_{:,2} H_1 g,
\]

\[
a_2 = [U]_{:,1}^T H_2 [U]_{:,1}, \quad b_2 = 2[U]_{:,1}^T H_2 [U]_{:,2},
\]

\[
c_2 = [U]_{:,2}^T H_2 [U]_{:,2}, \quad d_2 = 2[U]_{:,2} H_2 g - 1,
\]

\[
e_2 = 2[U]_{:,2} H_2 g.
\]
thereby obtain a linear equation as the Taylor series expansion and retain the first order term, and parameter \( \theta \) and \( \epsilon \). Then, by measurements to refine this raw estimation.

\[
\ln \frac{\text{form of } \hat{\theta}}{\theta} = \frac{1}{2} \epsilon^T \hat{C}^{-1} \epsilon + \frac{1}{2} \epsilon^T \hat{C}^{-1} \theta - \frac{1}{2} \theta^T \hat{C}^{-1} \theta + \frac{1}{2} \theta^T \hat{C}^{-1} \theta \]

where \( \hat{C} \) is the covariance matrix for the measurement noise and

\[
\hat{C} = \text{diag} \left( \sigma_1^2, \ldots, \sigma_M^2 \right),
\]

and

\[
[r]_i = \hat{\alpha}_i - \| \hat{p} + \hat{v}t_i - p_i \| - \hat{\beta} - \hat{\omega}t_i, i = 1, \ldots, M. \quad (24)
\]

When the sequential TOA measurement noise variance \( \sigma_i \) is identical, the selection strategy (23) can be simplified to the form of \( \min_\theta r^T \hat{C}^{-1} r \) to save computation.

C. Bias Compensation based on WLS

The raw parameter estimate \( \hat{\theta} \) obtained from the previous step needs further refinement to become an optimal estimation. We apply a WLS step using the residuals of all the TOA measurements to refine this raw estimation.

We first express (2) in the collective form as

\[
\hat{\tau} = h(\theta) + \epsilon, \quad (25)
\]

where \( \hat{\tau} = [\hat{\tau}_1, \ldots, \hat{\tau}_M]^T \), \( h(\theta) \) is a function of \( \theta \) as given by

\[
[h(\theta)]_i = \| \hat{p} + \hat{v}t_i - p_i \| + \hat{\beta} + \hat{\omega}t_i - \beta_i, i = 1, \ldots, M,
\]

and \( \epsilon = [\epsilon_1, \ldots, \epsilon_M]^T \).

Note that the raw estimate \( \hat{\theta} \) is not far from the true parameter \( \theta \) under the small noise condition. Thus, we apply the Taylor series expansion and retain the first order term, and thereby obtain a linear equation as

\[
\hat{\tau} = h(\hat{\theta}) + \left( \frac{\partial h(\theta)}{\partial \theta} |_{\theta=\hat{\theta}} \right) (\theta - \hat{\theta}) + \epsilon. \quad (26)
\]

We define the design matrix

\[
\hat{J} = \frac{\partial h(\theta)}{\partial \theta} |_{\theta=\hat{\theta}},
\]

and its \( i \)-th row is given by

\[
[\hat{J}]_{i,:} = \left[ \frac{\partial h(\theta)}{\partial \theta} |_{\theta=\hat{\theta}} \right]_{i,:} = \left[ -\hat{\beta}_i^T, -\hat{\beta}_i^T, 1, t_i \right], \quad (27)
\]

\[
\hat{J}_i = \frac{p_i - \hat{p} - \hat{v}t_i}{\| p_i - \hat{p} - \hat{v}t_i \|}, i = 1, \ldots, M. \quad (28)
\]

We apply the WLS method to (26) and obtain the refinement vector, denoted by \( \Delta \hat{\theta} \), as

\[
\Delta \hat{\theta} = \left( \hat{J}^T \hat{W} \hat{J} \right)^{-1} \hat{J}^T \hat{W} \left( \hat{\tau} - h(\hat{\theta}) \right), \quad (29)
\]

where the weighting matrix

\[
\hat{W} = \hat{C}^{-1} = \left( \mathbb{E} [\epsilon \epsilon^T] \right)^{-1}. \quad (30)
\]

Then, the final parameter estimation, denoted by \( \hat{\theta}_{\text{est}} = [\hat{p}_{\text{est}}^T, \hat{v}_{\text{est}}^T, \hat{\beta}_{\text{est}}, \hat{\omega}_{\text{est}}]^T \), is

\[
\hat{\theta}_{\text{est}} = \hat{\theta} + \Delta \hat{\theta} \quad (31)
\]

The procedure of the new CFJLAS method is summarized in Algorithm 1.

**Algorithm 1 Closed-form JLAS (CFJLAS)**

1: Input noisy sequential measurements \( \tilde{\tau}_i \), and PN positions \( p_i \), and clock offsets \( \beta_i, i = 1, \ldots, M \).
2: Linearization: Square the TOA measurement equations, and form matrices \( A, G \) and vector \( y \) based on (8) to (11).
3: Solution of intermediate variables: Solve quadratic equations (21) and (22) and select the root based on (23) to obtain raw parameter estimate \( \hat{\theta} \).
4: WLS refinement: Compute refined parameter \( \hat{\theta}_{\text{est}} \) using (29) and (31).
5: Output the parameter estimate result \( \hat{\theta}_{\text{est}} \).

We can see from the steps of the new CFJLAS method that it first finds the solution of the two intermediate variables. Then, it computes the parameters using the intermediate variables. Compared with the method proposed in [24], which estimates newly employed variables along with the parameters, the new CFJLAS requires fewer measurements by nature since it does not need to estimate the intermediate variables along with the parameters.

IV. PERFORMANCE ANALYSIS

It is of significant interest to evaluate the achievable estimation performance of this new CFJLAS approach. In this section, we first derive the CRLB, which is a lower bound on the achievable estimation error variance to quantify the estimation performance. After that, a theoretical error analysis is conducted to evaluate the estimation accuracy of the proposed CFJLAS compared with the CRLB. Then, we analyze the computational complexity in comparison with the iterative method.

A. CRLB Derivation

CRLB is the lower bound of the estimation accuracy of an unbiased estimator. It is written as

\[
\text{CRLB} = \mathcal{F}^{-1}(\theta), \quad (32)
\]

where \( \mathcal{F} \) is the Fisher information matrix (FIM). The likelihood function, denoted by \( p(\hat{\tau} | \theta) \), is

\[
p(\hat{\tau} | \theta) = \exp \left( \frac{-1}{2} (\hat{\tau} - h(\theta))^T \mathcal{W} (\hat{\tau} - h(\theta)) \right) \frac{1}{(2\pi)^{\frac{M}{2}} | \mathcal{W} |^{-\frac{1}{2}}}. \quad (33)
\]
The FIM is then given by
\[
\mathcal{F} = -\mathbb{E} \left[ \frac{\partial^2 \ln p(\hat{\theta})}{\partial \theta \partial \theta^T} \right] = \left( \frac{\partial h(\theta)}{\partial \theta} \right)^T W \frac{\partial h(\theta)}{\partial \theta}
\]

\[= J^T W J. \tag{34} \]

where
\[
[J]_{i,:} = \left[ \frac{\partial h(\theta)}{\partial \theta} \right]_{t_i} = [-I_l^T, -I_l^T t_i, 1, t_i], \tag{35} \]

\[
l_i = \frac{p_i - p - vt_i}{\|p_i - p - vt_i\|}, i = 1, \ldots, M. \tag{36} \]

B. Theoretical Error Analysis

In this subsection, we derive the theoretical error of the new CFJLAS, and show its optimality.

Introducing (25), we re-write (29) as
\[
\Delta \hat{\theta} = \left( J^T W \tilde{J} \right)^{-1} J^T W \left( \hat{\tau} - h(\tilde{\hat{\theta}}) \right)
\]

\[= \left( J^T W \tilde{J} \right)^{-1} J^T W \left( \hat{\tau} - h(\hat{\theta}) + h(\hat{\theta}) - h(\tilde{\hat{\theta}}) \right)
\]

\[= \left( J^T W \tilde{J} \right)^{-1} J^T W \tilde{\epsilon} + \left( J^T W \tilde{J} \right)^{-1} J^T W \left( h(\theta) - h(\tilde{\hat{\theta}}) \right). \tag{37} \]

According to (25) and (26), we have
\[
h_1(\theta) = h(\hat{\theta}) + \left( \frac{\partial h(\theta)}{\partial \theta} \right)_{\theta=\hat{\theta}} (\theta - \hat{\theta}). \tag{38} \]

By substituting (38) into (37), we come to
\[
\Delta \hat{\theta} = \left( J^T W \tilde{J} \right)^{-1} J^T W \tilde{\epsilon} + \left( J^T W \tilde{J} \right)^{-1} J^T W \left( \hat{\theta} - \tilde{\hat{\theta}} \right)
\]

\[= \left( J^T W \tilde{J} \right)^{-1} J^T W \tilde{\epsilon} + \hat{\theta} - \tilde{\hat{\theta}}. \tag{39} \]

Based on (31), we re-organize (39) as
\[
\tilde{\hat{\theta}}_{\text{est}} - \hat{\theta} = \left( J^T W \tilde{J} \right)^{-1} J^T W \tilde{\epsilon}. \tag{40} \]

C. Complexity Analysis

Following [37], a floating point addition, multiplication or square root operation for real numbers can be done in one flop. In this subsection, we estimate the complexities of the CFJLAS and the iterative method by counting the flops of the major operations.

We denote the total complexity of the CFJLAS by \(D\). As part of \(D\), the complexity caused by the step of solving the equations (21) and (22) is fixed, not related to the position dimension \(K\) or the number of PNs \(M\). Other complexities lie in finding the proper roots in (23), raw estimate in (13) and WLS refinement in (29). The detailed complexity analysis is given in Appendix B. We count the total number of flops as approximated by

\[D \approx 32K^3 + 16K^2M + 104K^2 + 62KM + 148K + 65M + 697, \tag{42} \]

where \(K\) is the dimension of the position, and \(M\) is the number of PNs.

For the iterative method, the total complexity is determined by the complexity in each iteration, denoted by \(L\), and the number of iterations, denoted by \(n\). The major operations in each iteration are comprised of matrix multiplication and matrix inversion. Each iteration has the same complexity. Thus, the total complexity of the iterative method is \(nL\). The detailed analysis is presented in Appendix B. The number of flops for \(L\) is approximated by

\[L \approx 16K^3 + 8K^2M + 56K^2 + 22KM + 64K + 16M + 24. \tag{43} \]

We can see that the propose new CFJLAS method has constant complexity \(D\), while the iterative method has increasing complexity \(nL\), when the iteration time \(n\) grows. We substitute \(K = 2\) and \(M = 8\) into (42) and (43) and obtain that \(D = 3689\) and \(L = 1240\). We can see that \(L\) and \(D/3\) are approximately the same. Therefore, we can expect that the new CFJLAS method has lower computational complexity when the number iteration exceeds 3 for the iterative method. This will be shown by simulation in Section V.

V. NUMERICAL SIMULATION

In this section, numerical simulations are carried out to evaluate the performance of the new CFJLAS method. The iterative ML method [23] is selected for comparison. The computational platform running the following simulations is Matlab R2019b on a PC with Intel Core i5-8400 CPU @2.8GHz and 32G RAM.

A. Performance Metrics

The mean square error (MSE) is used to evaluate the performance in the following numerical simulations. The MSEs
of the position $p$, velocity $v$, clock offset $\beta$ and clock skew $\omega$ are given by

$$MSE_p = \frac{1}{N} \sum_{i=1}^{N} \| p - \hat{p}_{est,i} \|^2,$$

$$MSE_v = \frac{1}{N} \sum_{i=1}^{N} \| v - \hat{v}_{est,i} \|^2,$$

$$MSE_{\beta} = \frac{1}{N} \sum_{i=1}^{N} \| \beta - \hat{\beta}_{est,i} \|^2,$$

$$MSE_{\omega} = \frac{1}{N} \sum_{i=1}^{N} \| \omega - \hat{\omega}_{est,i} \|^2,$$

(44)

where $N$ is the total number of Monte-Carlo runs, the variables with "∼" overhead are the estimated parameters from the proposed CFJLAS method.

CRLB is used as a benchmark to evaluate the lower bound of the accuracy. In the simulation scene, the theoretical lower bound of the position, the velocity, the clock offset and clock skew are computed by

$$\text{error}_p = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \text{CRLB}_{i,j},$$

$$\text{error}_v = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=K+1}^{2K} \text{CRLB}_{i,j},$$

$$\text{error}_\beta = \frac{1}{N} \sum_{i=1}^{N} \text{CRLB}_{2K+1,2K+1},$$

$$\text{error}_\omega = \frac{1}{N} \sum_{i=1}^{N} \text{CRLB}_{2K+2,2K+2},$$

(45)

where $[\cdot]_{i,j}$ is the entry at the $i$-th row and the $j$-th column of a matrix.

B. Simulation Settings

We create a 2D simulation scene with a MAS comprised of 8 PNs and 1 CN. To resemble the real world, 8 PNs and 1 CN are moving in a formation, as shown in Fig.2. PNs #1, #2 and #8 are at the back (left side in the figure) of the formation. PNs #3, #4, #5, #6 and #7 form a triangle in the front (right side in the figure) of the formation. One CN is initially placed near the center of this formation. All the nodes move forward along the $x$-axis, with the PNs’ speed being 2 m/s and the CN’s speed 5 m/s. We allocate 8 time slots to the 8 PNs with an interval of 5 ms between successive slots. The initial clock offset and initial clock skew of CN for each simulation run are set as random variables, drawn from uniform distributions, i.e., $\beta \sim \mathcal{U}(-10^{-5}, 10^{-5})$ s, and $\omega \sim \mathcal{U}(-20, 20)$ part per million (ppm).

For the iterative method, the maximum iteration time is set to 10 and the convergence threshold is set to 0.01 m, i.e., the iteration stops if the increment of the $(k + 1)$-th iteration $\| [\hat{p}_{est}(k + 1)^T, \hat{\beta}_{est}(k + 1)]^T - [\hat{p}_{est}(k) - \hat{\beta}_{est}(k)]^T \| < 0.01$. In each iteration, a matrix inversion is conducted to estimate the parameter increment. Before inversion, we compute the reciprocal condition number ($rcond$) to judge if the matrix is singular or not. Specifically, we set the threshold of the reciprocal condition number to $10^{-15}$, i.e., if $rcond < 10^{-15}$, the matrix is considered singular and the iterative method stops.

C. Estimation Performance

We set the TOA measurement noise $\sigma$ varying from $0.1$ m to $3$ m with a step of $0.29$ m. At each step, we conduct 100,000 Monte-Carlo simulations.

Fig. 3 illustrates the estimation error results with varying measurement noise. The theoretical CRLBs are computed based on (45), and the MSEs of the estimation errors are the average of 100,000 Monte-Carlo runs based on (44). Take Fig. 3 (a) as an example, the black solid line shows the theoretical position lower bound from CRLB. The blue dash line with circle markers and the magenta dash line with triangle markers show the positioning accuracy of the iterative method and the proposed CFJLAS method, respectively. For the iterative method, to obtain the optimal maximum likelihood (ML) result, the initial guess of the CN’s parameters is set to the ground truth. It can be seen that the positioning accuracy of the proposed CFJLAS method reaches CRLB under small noise condition. The position error of the proposed method is comparable with the iterative method initialized with the ground truth.

Other estimation errors for the velocity, clock offset and clock drift are shown in Fig. 3 (b), (c) and (d), respectively. The results of the new CFJLAS method are comparable to that of the iterative method. All the estimation errors reach their respective CRLBs. The results verify that the proposed method can optimally estimate the position, velocity, clock offset and the clock skew.

Note that the results of the iterative method shown in Fig. 3 is in the ideal case with ground truth initialization. In practice, we do not know the true values of the parameters to be estimated and thus cannot initialize the iterative method.
with the ground truth. To compare the performance of the two methods in this practical case, we initialize the iterative method with a random position, which is the sum of the true CN coordinate and a random Gaussian noise with a 200 m STD. We run 100,000 Monte-Carlo simulations at each measurement noise step. Other settings remain the same. We note from the simulation settings that there are three cases for the iterative method to stop, i.e., (a) convergence, i.e., the norm of the parameter estimation increment in the current iteration is smaller than 0.01 m, (b) the matrix to be inverted is singular, i.e., \( rcond < 10^{-15} \), (c) the iteration count exceeds 10.

We first consider the convergence case (case (a)) only and illustrate the estimation errors in Fig. 4.

It is shown in Fig. 4 that the iterative method has much larger estimation error than that of the CFJLAS since it fails to converge to the global minimum to generate the correct result. Errors for other parameters have the similar pattern. This result demonstrates the superiority of the new CFJLAS over the iterative method.

We further investigate the failure of the iterative method. In the 100,000 simulation runs for each measurement noise step, we count the number of results in the above three cases. In addition, we use a threshold of \( 9 \text{CRLB} \), which equals the 3-\( \sigma \) or 99.7% of a Gaussian distribution, to judge if the localization result is correct or not, i.e., in the case that the estimated position MSE is smaller than \( 9 \text{CRLB} \), the estimation result is judged correct. The results of all the above four cases are listed in Table II.

As can be seen from Table II, the numbers of results in the correct estimate case (row 2, case (0)) are smaller but close to that of the convergence case (row 3, case (a)). This gap between the two cases indicates that the some results of the convergence case (a) do not converge to the correct solution and thus cause the large errors shown in Fig. 4. Besides the convergence case, results also fall into (b) singular matrix and (c) iteration count exceeded cases. All the three cases constitute the total number of simulation runs.

We use the same threshold of \( 9 \text{CRLB} \) to judge the correctness for both the CFJLAS method and the iterative method. The correctness rates of the two methods at each measurement noise step are shown in Fig. 5. We can see that the correctness rate of the CFJLAS method is close to 1, showing its good estimation performance. It is much higher than the correctness rate of the iterative method, which is lower than 88%. This result verifies that the CFJLAS method does not depend on the initial guess and can always obtain a high accuracy estimation. However, the iterative method is dependent on the initial guess, and it has a probability to fail in obtaining the correct solution when the initial guess is not close to the true value. This result shows that the new CFJLAS method has better estimation performance than the iterative method.

D. Computational Complexity

To evaluate the computational complexity, we compare the new CFJLAS method and the iterative method. We note that the iterative method requires more iterations or more computational complexity when the initial guess deviates from the true parameters. However, the complexity of the new CFJLAS is stable as analyzed in Section IV-C.

We simulate the case with inaccurate initialization for the iterative method. We fix the measurement noise \( \sigma = 2 \text{ m} \), and vary the error STD of the initial position for the iterative method from 10 to 200 m. At each step, we run 100,000 Monte-Carlo simulations for each of the two methods.

The computation time of both methods are depicted in Fig. 6 (a). The average iteration count of the iterative method from 100,000 Monte-Carlo runs is shown in Fig. 6 (b). We can see from the figure that the iteration count of the iterative method increases when the initial guess has larger error. With inaccurate initialization, the computational complexity of the
TABLE II
NUMBER OF RESULTS FOR ALL CASES OF THE ITERATIVE METHOD

| Measurement noise $\sigma$ (m) | 0.10 | 0.39 | 0.68 | 0.97 | 1.26 | 1.55 | 1.84 | 2.13 | 2.42 | 2.71 | 3.00 |
|---------------------------------|------|------|------|------|------|------|------|------|------|------|------|
| (0) Correct Estimate (MSE < 9CRLB) | 86883 | 86745 | 86846 | 86804 | 86884 | 86807 | 86603 | 87035 | 86876 | 86958 | 86944 |
| (a) Convergence (increment norm < 0.01 m) | 86984 | 86875 | 86965 | 86869 | 86972 | 86906 | 86716 | 87130 | 86952 | 87024 | 87025 |
| (b) Singular Matrix ($\text{rcond} < 10^{-15}$) | 4511 | 4507 | 4436 | 4503 | 4615 | 4437 | 4557 | 4471 | 4442 | 4478 | 4358 |
| (c) Iteration Count Exceeded ($\geq 10$) | 8505 | 8618 | 8599 | 8628 | 8413 | 8657 | 8727 | 8399 | 8606 | 8498 | 8617 |

Note: The number of results in case (0) is close to that in case (a). The gap between them shows that the iterative method may produce wrong solutions even if it converges. The numbers in cases (a), (b) and (c) have a sum of 100,000, showing that they cover all the cases for the iterative method.

Fig. 5. Correctness rate of CFJLAS and iterative method. The threshold to judge the correctness is set to 9CRLB, which equals to the $3-\sigma$ of a Gaussian distribution. The iterative method is initialized with 200 m STD. 100,000 Monte-Carlo runs are done for each noise step. The proposed CFJLAS has a correctness rate of about 99.7%, while the iterative method only around 87%. Unlike the iterative method, which may fail with inaccurate initial guess, the CFJLAS does not depend on initialization and can always obtain a correct estimation.

Fig. 6. (a) Computation time (CFJLAS & Iterative) vs. initial position error. (b) Average iteration count (Iterative) vs. initial position error. The measurement noise $\sigma$ is fixed at 2 m. The computation time and the average iteration count are the sum and average of 100,000 simulations, respectively. The iteration count and computation time of the iterative method grow when the initial guess deviates from the true value. The computation time of the CFJLAS is constant and is smaller than that of the iterative method.

The solution to a quadratic equation set such as (21) and (22) can be found in [36]. For the convenience of interested readers, we put the method here in this appendix. Note that the notations used herein are only effective within this appendix.
We replace the unknowns with \(x\) and \(y\), respectively, and rewrite the two quadratic equations of (21) and (22) as
\[
\begin{align*}
a_1x^2 + b_1xy + c_1y^2 + d_1x + e_1y + f_1 &= 0, \quad (46) \\
a_2x^2 + b_2xy + c_2y^2 + d_2x + e_2y + f_2 &= 0. \quad (47)
\end{align*}
\]
We first remove the \(y^2\) term by multiplying \(c_1\) and \(c_2\) to (47) and (46), respectively, and then subtracting the resulting equations. After re-organizing, we obtain
\[
(t_1x + t_2)y = t_3x^2 + t_4x + t_5, \quad (48)
\]
where
\[
\begin{align*}
t_1 &= b_1c_2 - b_2c_1, \\
t_2 &= e_1c_2 - e_2c_1, \\
t_3 &= -a_1c_2 + a_2c_1, \\
t_4 &= -d_1c_2 + d_2c_1, \\
t_5 &= -f_1c_2 + f_2c_1.
\end{align*}
\]
Here are two cases. One is \(t_1x + t_2 = 0\) and the other is \(t_1x + t_2 \neq 0\).

Case 1: \(t_1x + t_2 = 0\)

If \(t_1 = 0\), then \(t_2\) must equal to zero. In this sub-case, the problem reduces to solving the equation of
\[
t_3x^2 + t_4x + t_5 = 0. \quad (49)
\]
After substituting the root of \(x\) from (49) into (47), the root of \(y\) can be found.

If \(t_1 \neq 0\), then we need to test if \(-t_2/t_1\) is the root of \(x\) by substituting it into (49). If it satisfies (49), then the root of \(y\) can be found by substituting \(x\) into (47). Otherwise, there is no solution.

Case 2: \(t_1x + t_2 \neq 0\)

In this case, we have
\[
y = (t_3x^2 + t_4x + t_5)/(t_1x + t_2). \quad (50)
\]
By substituting (50) into (47), we come to a quartic equation of \(x\) as
\[
\alpha x^4 + \beta x^3 + \gamma x^2 + \lambda x + \mu = 0, \quad (51)
\]
where
\[
\begin{align*}
\alpha &= a_1t_1^2 + b_1t_1t_3 + c_1t_3^2, \\
\beta &= d_1t_1^2 + 2a_1t_1t_2 + b_1t_1t_4 + b_1t_2t_3 + 2c_1t_3t_4 + e_1t_1t_3, \\
\gamma &= c_1(t_3^2 + 2t_3t_5) + a_1t_2^2 + f_1t_1^2 + b_1t_1t_5 + b_1t_2t_4 + 2d_1t_2t_3 + e_1t_4t_3 + e_1t_2t_3, \\
\lambda &= d_1t_2^2 + b_1t_2t_5 + 2c_1t_4t_5 + e_1t_1t_5 + e_1t_4t_3 + 2f_1t_1t_2, \\
\mu &= f_1t_2^2 + e_1t_2t_5 + c_1t_5.
\end{align*}
\]
Solution of the quartic equation (51) can be found in mathematical literature such as [38], [39]. We simply write the solution as follows in this section without derivation for those interested readers.

There are at most four roots for this equation, either real or complex values. The general form of the roots is given by
\[
\begin{align*}
x(1), x(2) &= -\frac{\beta}{4\alpha} - s \pm \frac{1}{2} \sqrt{-4s^2 - 2p + \frac{q_0}{s}}, \\
x(3), x(4) &= -\frac{\beta}{4\alpha} + s \pm \frac{1}{2} \sqrt{-4s^2 - 2p - \frac{q_0}{s}}. \\& (52)
\end{align*}
\]
with the variables expressed as follows,
\[
\begin{align*}
s &= \frac{1}{2} \sqrt{-\frac{2}{3}p + \frac{1}{3\alpha}(q_1 + \Delta_0),} \\
p &= \frac{8\alpha \gamma - 3\beta^2}{8\alpha^2}, \\
q_0 &= \frac{\beta^3 - 4\alpha \beta \gamma + 8\alpha^2 \lambda}{8\alpha^3}, \\
\Delta &= -\frac{\Delta_1^2 - 4\Delta_3}{27}, \\
\Delta_0 &= \gamma^2 - 3\beta \lambda + 12\alpha \mu, \\
\Delta_1 &= 2\gamma^3 - 9\beta \gamma \lambda + 27\beta^2 \mu + 27\alpha \lambda^2 - 72\alpha \gamma \mu. \quad (53)
\end{align*}
\]

**Appendix B**

**Complexity Analysis for CFJLAS and Iterative Method**

We assume that a single addition, multiplication or square root operation for real numbers can be done in one flop [37]. By counting the flops of the major operations in the CFJLAS and the iterative method, their complexities can be estimated.

**A. CFJLAS Method**

For the CFJLAS method, we first investigate the complexity of the raw estimate (13) and the WLS compensation (29). Following [40], we use \(2n^2\), where \(n\) is the number of rows (or columns) of a square matrix, to approximate the number of flops required for a matrix inverse operation. The number of flops required by (13) is
\[
\begin{align*}
2(2K + 2)^2(M - 1) + 2(2K + 2)^3 + \\
2(2K + 2)^2 + 2(2K)^2 + (M - 1) + 5M - 5
\end{align*}
= 16K^3 + 8K^2M + 48K^2 + 20KM + 48K + 17M + 15. \quad (54)
\]
where \(K\) is the dimension of the position and \(M\) is the number of PNs.

Considering that \(W\) is a diagonal matrix, the number of flops required by (29) is
\[
\begin{align*}
2(2K + 2)^2M + 2(2K + 2)^3 + 2(2K + 2)M + \\
2(2K + 2)^2 + (2K + 2)^2M + 2M
\end{align*}
= 16K^3 + 8K^2M + 56K^2 + 22KM + 64K + 16M + 24. \quad (55)
\]

We then look into the complexity in solution for the intermediate variables given by Section III-B. The coefficients of the equations (21) and (22) are obtained by the equations below them. We take \(a_1\) as an example. By observing \(H_1\), we can see that there are \(K + 1\) entries with value of 1. Therefore, \(a_1 = [U]_{1}\), \(H_1[U]_{1,:}\) takes \((K + 1)\) multiplications and \(K\) additions, \((2K + 1)\) flops in total. All the six coefficients with subscript “1” are similar and take \(6(2K + 1)\) flops for \(a_1 \sim f_1\). We also note that \(H_2\) has \((2K + 2)\) entries of one and all other entries of zero in it. Thus, the total number of flops for computing \(a_2 \sim f_2\) is \(6(4K + 3)\). We then have
\[
6(6K + 4) \quad (56)
\]
flops for computing the coefficients of the equation set.

We then count the flops of computing the coefficients in (48) and obtain 15. The number of flops required by computing the
coefficients in (51) is 81. For finding the roots of \( x \) in (52), there may be operations of complex numbers. We consider there are 2 flops for a complex addition and 6 flops for a complex multiplication or square root. Therefore, solution for the roots of \( x \) in (52) is \( 4 \times 38 = 152 \) flops. Computing the variables in (53) requires 278 flops. Then, computing the roots of \( y \) in (50) requires \( 4 \times 28 = 112 \) flops. Therefore, finding the roots of the quadratic equations (21) and (22) need
\[
15 + 81 + 152 + 278 + 112 = 638 \tag{57}
\]
flops.

Root selection given by (23) needs to compute the residual as given by (24). Computing \( \hat{\mathbf{p}} + \hat{v}t_i - \mathbf{p}_i \) needs \( K \) multiplications and \( 2K \) additions. The norm operation needs \( K \) multiplications, \( (K - 1) \) additions and 1 square root. Thus, there are 5 flops for one distance. We need 3 additions and 1 multiplications to compute one residual \( r_j^2 \). Considering \( M \) measurements and at most 4 roots from the equation set, we need
\[
4M(5K + 4) \tag{58}
\]
flops to compute all the residuals. The root selection in (23) has \( 2M \) multiplications and \( (2M - 1) \) additions. For 4 roots, there are
\[
4(4M - 1) \tag{59}
\]
flops in total.

As a result, we obtain the total number of flops by summing (54) to (59) as
\[
D \approx 32K^3 + 16K^2M + 104K^2 + 62KM + 148K + 65M + 697 \tag{60}
\]

B. Iterative Method

The complexity in a single iteration, denoted by \( L \), is mainly comprised of matrix multiplication and matrix inversion. Following the iterative method that estimates position, velocity, clock offset and drift in [23], the matrix multiplication and matrix inverse operations are similar to that of (29). Therefore, it has the same expression as (55), i.e.,
\[
L \approx 16K^3 + 8K^2M + 56K^2 + 22KM + 64K + 16M + 24. \tag{61}
\]

In most of the time, the iterative method requires more than one iteration to obtain the correct result. If there are \( n \) iterations, then the total complexity is \( nL \).

REFERENCES

[1] P. Beaumont and B. Chaib-draa, “Multiagent coordination techniques for complex environments: The case of a fleet of combat ships,” IEEE Trans. Syst., Man, Cybern. C, vol. 37, no. 3, pp. 373–385, 2007.

[2] D. Panasetsky, P. Etingov, and N. Voropai, “Multi-agent approach to emergency control of power system,” in 2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies. IEEE, 2008, pp. 2157–2161.

[3] A. F. G. Ferreira, D. M. A. Fernandes, A. P. Catarino, and J. L. Monteiro, “Localization and positioning systems for emergency responders: A survey,” IEEE Commun. Surveys Tuts., vol. 19, no. 4, pp. 2836–2870, 2017.

[4] R. W. Beard, T. W. McLain, M. A. Goodrich, and E. P. Anderson, “Coordinated target assignment and intercept for unmanned air vehicles,” IEEE Trans. Robot., vol. 18, no. 6, pp. 911–922, 2002.

[5] M. Maimone, Y. Cheng, and L. Mathies, “Two years of visual odometry on the mars exploration rovers,” Journal of Field Robotics, vol. 24, no. 3, pp. 169–186, 2007.

[6] B. Denis, J.-B. Pierrot, and C. Abou-Rjeily, “Joint distributed synchronization and positioning in UWB ad hoc networks using TOA,” IEEE Trans. Microw. Theory Techn., vol. 54, no. 4, pp. 1896–1911, 2006.

[7] R. M. Vaghefi and R. M. Buehner, “Cooperative joint synchronization and localization in wireless sensor networks,” IEEE Trans. Signal Process., vol. 65, no. 14, pp. 3615–3627, 2017.

[8] A. Ahmad, E. Serpentin, H. Nounou, and M. Nounou, “Joint node localization and time-varying clock synchronization in wireless sensor networks,” IEEE Trans. Wireless Commun., vol. 12, no. 10, pp. 5322–5333, 2013.

[9] B. Ettlinger, F. Meyer, F. Hlawatsch, A. Springer, and H. Wymeersch, “Cooperative simultaneous localization and synchronization in mobile agent networks,” IEEE Trans. Signal Process., vol. 65, no. 14, pp. 3587–3602, 2017.

[10] J.-A. Luo, X.-P. Zhang, and Z. Wang, “A new passive source localization method using AOA-GROA-TDOA in wireless sensor array networks and its Cranmer-Rao bound analysis,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013, pp. 4031–4035.

[11] J. Zheng and Y.-C. Wu, “Joint time synchronization and localization of an unknown node in wireless sensor networks,” IEEE Trans. Signal Process., vol. 58, no. 3, pp. 1309–1320, 2010.

[12] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more. Springer Science & Business Media, 2007.

[13] P. Misra and P. Enge, Global Positioning System: signals, measurements and performance second edition. Ganga-Jamuna Press, 2006.

[14] A. V. Picois and N. Samama, “Near-far interference mitigation for pseudolites using double transmission,” IEEE Trans. Aerosp. Electron. Syst., vol. 50, no. 4, pp. 2929–2941, 2014.

[15] H. G. Myung, J. Lim, and D. J. Goodman, “Single carrier FDMA for uplink wireless transmission,” IEEE Veh. Technol. Mag., vol. 1, no. 3, pp. 30–38, 2006.

[16] S. Dwivedi, A. De Angelis, and P. Händel, “Scheduled UWB pulse transmissions for cooperative localization,” in 2012 IEEE International Conference on Ultra-Wideband. IEEE, 2012, pp. 6–10.

[17] Q. Shi, X. Cui, S. Zhao, S. Xu, and M. Lu, “BLAS: Broadcast relative localization and clock synchronization for dynamic dense multi-agent systems,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 5, pp. 3822–3839, 2020.

[18] S. Dwivedi, D. Zachariah, A. De Angelis, and P. Händel, “Cooperative decentralized localization using scheduled wireless transmissions,” IEEE Commun. Lett., vol. 17, no. 6, pp. 1240–1243, 2013.

[19] M. Hamer and R. D’Andrea, “Self-calibrating ultra-wideband network supporting multi-robot localization,” IEEE Access, vol. 6, pp. 22292–22304, 2018.

[20] D. Zachariah, A. De Angelis, S. Dwivedi, and P. Händel, “Self-localization of asynchronous wireless nodes with parameter uncertainties,” IEEE Signal Process. Lett., vol. 20, no. 6, pp. 551–554, 2013.

[21] P. Carroll, K. Mahmood, S. Zhou, H. Zhou, X. Xu, and J.-H. Cui, “On-demand asynchronous localization for underwater sensor networks,” IEEE Trans. Signal Process., vol. 62, no. 13, pp. 3337–3348, 2014.

[22] J. Yan, X. Zhang, X. Luo, Y. Wang, C. Chen, and X. Guan, “Asynchronous localization with mobility prediction for underwater acoustic sensor networks,” IEEE Trans. Veh. Technol., vol. 67, no. 3, pp. 2543–2556, 2017.

[23] S. Zhao, X.-P. Zhang, X. Cui, and M. Lu, “Optimal localization with sequential pseudorange measurements for moving users in a time division broadcast positioning system,” arXiv preprint arXiv:2101.01636, 2020.

[24] Q. Shi, X. Cui, S. Zhao, and M. Lu, “Sequential TOA-Based moving target localization in multi-agent networks,” IEEE Commun. Lett., vol. 24, no. 8, pp. 1719–1723, 2020.

[25] D. Zachariah, A. De Angelis, S. Dwivedi, and P. Händel, “Scheduled-based sequential localization in asynchronous wireless networks,” EURASIP Journal on Advances in Signal Processing, vol. 2014, no. 1, p. 16, 2014.

[26] F. B. Bancroft, “An algebraic solution of the GPS equations,” IEEE Trans. Aerosp. Electron. Syst., vol. 1, no. 1, pp. 56–59, 1985.

[27] H. Schau and A. Robinson, “Passive source localization employing intersecting spherical surfaces from time-of-arrival differences,” IEEE Trans.
Trans. Acoust., Speech, Signal Process., vol. 35, no. 8, pp. 1223–1225, 1987.
[28] J. Smith and J. Abel, “Closed-form least-squares source location estimation from range-difference measurements,” IEEE Trans. Acoust., Speech, Signal Process., vol. 35, no. 12, pp. 1661–1669, 1987.
[29] Y.-T. Chan and K. Ho, “A simple and efficient estimator for hyperbolic location,” IEEE Trans. Signal Process., vol. 42, no. 8, pp. 1905–1915, 1994.
[30] K. W. Cheung, H.-C. So, W.-K. Ma, and Y.-T. Chan, “Least squares algorithms for time-of-arrival-based mobile location,” IEEE Trans. Signal Process., vol. 55, no. 10, pp. 5103–5107, 2007.
[31] S. Zhu and Z. Ding, “Joint synchronization and localization using toas: A linearization based WLS solution,” IEEE J. Sel. Areas Commun., vol. 28, no. 7, pp. 1017–1025, 2010.
[32] J. Huang, Y. Xue, and L. Yang, "An efficient closed-form solution for joint synchronization and localization using TOA," Future Generation Computer Systems, vol. 29, no. 3, pp. 776–781, 2013.
[33] M. Sun and L. Yang, “On the joint time synchronization and source localization using TOA measurements,” Int. J. Distrib. Sens. Netw., vol. 9, no. 2, p. 794805, 2013.
[34] Y. Wang, J. Huang, L. Yang, and Y. Xue, “TOA-based joint synchronization and source localization with random errors in sensor positions and sensor clock biases,” Ad Hoc Netw., vol. 27, pp. 99–111, 2015.
[35] S. Zhao, X.-P. Zhang, X. Cui, and M. Lu, “A closed-form localization method utilizing pseudorange measurements from two non-synchronized positioning systems,” IEEE Internet Things J., vol. 8, no. 2, pp. 1082–1094, 2021.
[36] G. Golub and C. Van Loan, Matrix Computations, 4th Ed. The Johns Hopkins University Press, 2013.
[37] S. L. Shmakov, “A universal method of solving quartic equations,” International Journal of Pure and Applied Mathematics, vol. 71, no. 2, pp. 251–259, 2011.
[38] M. Riccardi, “Solution of cubic and quartic equations,” Formalized Mathematics, vol. 17, no. 2, pp. 117–122, 2009.
[39] E. S. Quintana, G. Quintana, X. Sun, and R. van de Geijn, “A note on parallel matrix inversion,” SIAM Journal on Scientific Computing, vol. 22, no. 5, pp. 1762–1771, 2001.