

Simultaneous Time Synchronization and Mutual Localization for Multi-robot System

Xiangyong Wen*, Yingjian Wang*, Xi Zheng, Kaiwei Wang, Chao Xu, Fei Gao

Abstract—Mutual localization stands as a foundational component within various domains of multi-robot systems. Nevertheless, in relative pose estimation, time synchronization is usually underappreciated and rarely addressed, although it significantly influences estimation accuracy. In this paper, we introduce time synchronization into mutual localization to recover the time offset and relative poses between robots simultaneously. Under a constant velocity assumption in a short time, we fuse time offset estimation with our previous bearing-based mutual localization by a novel error representation. Based on the error model, we formulate a joint optimization problem and utilize semi-definite relaxation (SDR) to furnish a lossless relaxation. By solving the relaxed problem, time synchronization and relative pose estimation can be achieved when time drift between robots is limited. To enhance the application range of time offset estimation, we further propose an iterative method to recover the time offset from coarse to fine. Comparisons between the proposed method and the existing ones through extensive simulation tests present prominent benefits of time synchronization on mutual localization. Moreover, real-world experiments are conducted to show the practicality and robustness.

I. INTRODUCTION

Multi-robot systems have gained more and more attention from industry and academia due to their strong environmental adaptability, efficient mission completion, and high fault tolerance in complex tasks, such as multi-robot exploration [1] and post-disaster rescue [2]. In the collaborative tasks of multi-robot systems, sharing coordinate frames plays a crucial role in providing localization consensus. Some works rely on external devices, such as GPS, motion capture, and UWB, to achieve coordinate frame sharing. The dependence on external devices limits the applications of these multi-robot systems in many real-world environments, such as denied or sheltered scenarios. Therefore, performing relative localization with only onboard sensors has recently become a focus of multi-robot research.

Among various relative localization approaches, methods [3]–[5] that leverage mutual observation measurements between robots to recover relative poses have attracted

This work was supported by the National Natural Science Foundation of China under grant no. 62322314.

Xiangyong Wen, Yingjian Wang, Chao Xu and Fei Gao are with the Institute of Cyber-Systems and Control, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China, and also with the Huzhou Institute, Zhejiang University, Huzhou 313000, China.

Kaiwei Wang is with the College of Optical Science and Engineering, Zhejiang University, Hangzhou, 310027, China.

Xi Zheng is with the Hong Kong Polytechnic University, Hong Kong, 100872, China.

Corresponding author: Fei Gao and Yingjian Wang. E-mails: {wenxiangyong, yj_wang, fgaoaa}@zju.edu.cn.

*Equal contributors.

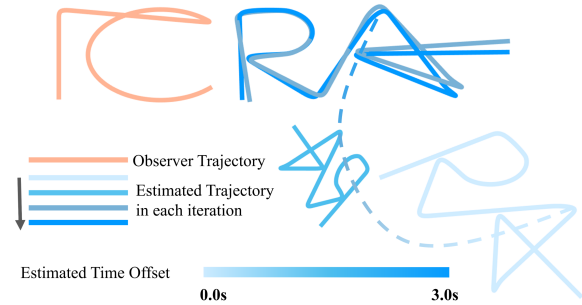


Fig. 1. The 'ICRA' trajectory recovering in an extremely large time offset situation. It can be seen that the trajectory of the letter 'RA' gradually approaches the ground truth 'IC' frame as the number of iterative optimizations increases.

much attention since they are independent of environment and require quite a low communication bandwidth. In our previous work [6], relative pose estimation with optimality guarantees was achieved by utilizing bearing measurements between robots. However, many drawbacks, such as time synchronization, still limit the practical application of relative localization methods. It denotes that data from different robots have different timestamps, typically from the unsynchronized clocks and sensors triggering delay. Such time offsets may lead to incorrect data associations, divergence in relative pose estimation, and even potential collapse of multi-robot systems. Some efforts have attempted to address this issue using network protocols like Network Time Protocol (NTP) [7] and Precision Time Protocol (PTP) [8]. These methods usually require a certain hardware and may have poor performance with high network latency. Besides, time synchronization has been extensively studied in multi-sensor fusion, where researchers aim to calibrate the time offset between sensors. For example, Qin [9] introduced time offset estimation between cameras and IMUs in visual-inertial odometry, improving estimation accuracy.

Inspired by Qin [9], this paper first introduces the online time offset estimation into mutual localization to solve time synchronization and relative pose estimation simultaneously. In most cases, the robot's velocity can be seen as constant if time is short. Based on this, we introduce time offset into the observation model, resulting in a novel error representation, including time offset and relative pose. Since the error representation can be reformulated into linear form, the joint state estimation can be formulated as a quadratically constrained quadratic problem (QCQP) by introducing auxiliary variables. Then, this problem is further relaxed into a convex programming by semi-definite relaxation, through which a

globally optimal solution can be obtained in polynomial time. Furthermore, considering that the constant velocity assumption will be invalid when the time offset is large, we design an iterative optimization framework, which recovers the time offset and relative pose from coarse to fine. Extensive simulation and real-world experiments verify our proposed method's outperforming accuracy and robustness in multi-robot localization scenarios.

In summary, our contributions are listed as follows:

- 1) We introduce time offset estimation into multi-robot relative localization, deriving a novel error representation and joint estimation problem.
- 2) We design an iterative optimization strategy, significantly enhancing the accuracy and range of time offset estimation.
- 3) We conduct extensive simulations and real-world experiments to validate the feasibility of the proposed method.

II. RELATED WORKS

A. Mutual Localization

The mainstream of mutual localization can be categorized into map-based methods and mutual observation-based methods. Map-based methods [10]–[13] take the environment features as the basis and realize mutual localization by environment landmarks sharing and loop closure detection. However, this approach occupies a large computational resource and is significantly affected by the surrounding environment. On the contrary, mutual observation-based approaches [3]–[5] focus only on relative measurements between robots, such as distance and bearing between robots. Stegagno [14] proposed a probabilistic multiple formation registration method for anonymous bearing-only measurements localization based on the particle filter. Zhou [3] summarized the mutual localization question into 14 minimal problems by the symmetry of range and bearing measurements and provided algebraic and numerical solutions to solve these minimal problems. Our previous work [6] formulated the mutual localization as a mixed integer quadratically constrained quadratic problem (MIQCQP) and relaxed this nonconvex problem into an SDP problem to get a certifiably global optimal solution. As far as we know, existing works did not discuss the time synchronization issue in mutual localization, although the time drift between robots will severely influence the performance of multi-robot systems.

B. Time synchronization

Researchers usually use network protocols to solve the time synchronization problem in multi-robot systems, such as NTP [7] and PTP [8]. Among that, NTP has a theoretical accuracy of 1 ms and does not rely on additional hardware. However, its accuracy is severely affected by network latency and bandwidth limitations. In multi-robot systems, frequent communication between robots may lead to discontinuous data streams and incorrect time offset estimation. The accuracy of PTP is up to 1 μ s but needs specialized and expensive hardware equipment. The limitation of weight and price

restricts its application, especially for low-cost multi-robot systems. On the other hand, researchers in communication networks considered system time offset as an unknown parameter and used estimation algorithms to solve time synchronization problems. Yang [15] employed an interactive multi-model Kalman filter that adaptively calculated clock skew based on environmental changes. In Shi's method [16], a maximum likelihood estimation with an innovative implementation method was presented to minimize the time delay. Like NTP, these methods rely on stable connections between agents, limiting the application in multi-robot systems.

In addition, multi-sensor fusion for individual robots faces a similar time synchronization problem. Furgale [17] released a Kalibr toolbox, widely used to calibrate the time offset and extrinsic parameters between the camera and IMU by continuous-time batch estimation. Qin [9] integrated time offset calibration into the simultaneous localization and mapping (SLAM) optimization framework and solved it in parallel with the robot state estimation problem. Inspired by the above works, we utilize robots' motion in this paper to recover the time offset and estimate the relative poses simultaneously. The tightly coupled optimization brings high accuracy and robustness in practice.

III. RELATIVE LOCALIZATION WITH TIME OFFSET ESTIMATION

We study the bearing-based relative localization, which uses only local odometry and inter-robot bearing measurements to recover the relative poses. In this section, we introduce time synchronization into relative localization to optimize the offset and relative pose simultaneously. First, we utilize uniform velocity assumption to derive a novel error format. Thanks to the new error representation, the time offset between trajectories can be optimized continuously. Furthermore, exploiting the linear form of error, we construct a least square problem. Then, by Shur Complement, we obtain a marginalized problem to recover time offset and relative rotations.

A. Novel Error Representation with Time Offset

We consider two robots: a observing robot \mathcal{A}_1 and a observed robot \mathcal{A}_2 . As shown in Fig.2, the corresponding local odometry frame are defined as \mathcal{L}_1 and \mathcal{L}_2 . $\mathcal{L}_n \mathbf{t}_{\tau_k} \in \mathbb{R}^3$ and $\mathcal{L}_n \mathbf{R}_{\tau_k} \in SO(3)$ denotes the translation and rotation in frame \mathcal{L}_n at \mathcal{A}_n 's local time τ_k . We emphasize the definition of local time, since if there is a time offset $\Delta\tau$ between robots, each robot will have a different local time even at the same "world" time. Similarly, ${}^{\mathcal{A}_1} \mathbf{b}_{\tau_k}^{\mathcal{A}_2}$ denotes the bearing measurement of \mathcal{A}_2 in the frame of \mathcal{A}_1 at \mathcal{A}_1 ' local time τ_k . Our goal is to estimate the relative frame transformation, ${}^{\mathcal{L}_1} \mathbf{t}$ and ${}^{\mathcal{L}_2} \mathbf{R}$, and time offset $\Delta\tau$, using each robot's local odometry and inter-robot bearing measurements.

In our previous work, the estimation error term can be written as:

$$\mathbf{e}_{\tau_k} = {}^{\mathcal{L}_1} \mathbf{R}_{\tau_k} \mathbf{b}_{\tau_k} D_{\tau_k} + {}^{\mathcal{L}_1} \mathbf{t}_{\tau_k} - ({}^{\mathcal{L}_2} \mathbf{R} {}^{\mathcal{L}_2} \mathbf{t}_{\tau_k} + {}^{\mathcal{L}_2} \mathbf{t}), \quad (1)$$

where $\mathcal{L}_2^1 \mathbf{R}$, $\mathcal{L}_2^1 \mathbf{t}$ and inter-robot distance D_{τ_k} are unknown variables to be solved. When the local time of two robots has been synchronized, the error makes sense. However, if there is any time offset $\Delta\tau$ between \mathcal{A}_1 and \mathcal{A}_2 , this error modeling is not precise anymore. That's because when \mathcal{A}_1 is with state $\mathcal{L}_1 \mathbf{R}_{\tau_k}$ and $\mathcal{L}_1 \mathbf{t}_{\tau_k}$, the position of \mathcal{A}_2 corresponding to the bearing $\mathcal{A}_1 \mathbf{b}_{\tau_k}^{\mathcal{A}_2}$ is not $\mathcal{L}_2 \mathbf{t}_{\tau_k}$ but $\mathcal{L}_2 \mathbf{t}_{(\tau_k + \Delta\tau)}$. Thus, the estimation error with time offset should be correlated as

$$\mathbf{e}_{\tau_k} = \mathcal{L}_1 \mathbf{R}_{\tau_k} \mathbf{b}_{\tau_k} D_{\tau_k} + \mathcal{L}_1 \mathbf{t}_{\tau_k} - (\mathcal{L}_2^1 \mathbf{R} \mathcal{L}_2^2 \mathbf{t}_{(\tau_k + \Delta\tau)} + \mathcal{L}_2^1 \mathbf{t}). \quad (2)$$

The correlation is simple but very significant to relative pose estimation. If the time offset is not considered, the bias will exist throughout the entire estimation process and affect the accuracy, which is verified in the subsequent experiments in Sec.V. Then, to optimize the time offset, it is necessary to model $\mathcal{L}_2 \mathbf{t}_{(\tau_k + \Delta\tau)}$ using $\Delta\tau$ precisely. Here, we borrow the idea that the pixel has constant 2D velocity on the image in sensor time calibration [9]. We consider that robots move with a constant velocity in a short time and thus obtain the approximation of $\mathcal{L}_2 \mathbf{t}_{(\tau_k + \Delta\tau)}$ as

$$\mathcal{L}_2 \mathbf{t}_{(\tau_k + \Delta\tau)} \approx \mathcal{L}_2 \mathbf{t}_{\tau_k} + \mathcal{L}_2 \mathbf{v}_{\tau_k} \Delta\tau, \quad (3)$$

where $\mathcal{L}_2 \mathbf{v}_{\tau_k}$ is the velocity of robot \mathcal{A}_2 , which can be obtained by local odometry. This constant velocity assumption is reliable since actual robots are always desired to move smoothly in navigation. In this way, we effectively introduce the time offset $\Delta\tau$ into the relative pose estimation and derive a novel error representation:

$$\mathbf{e}_{\tau_k} = \mathcal{L}_1 \mathbf{R}_{\tau_k} \mathbf{b}_{\tau_k} D_{\tau_k} + \mathcal{L}_1 \mathbf{t}_{\tau_k} - (\mathcal{L}_2^1 \mathbf{R} \mathcal{L}_2^2 \mathbf{t}_{\tau_k} + \mathcal{L}_2^1 \mathbf{R} \mathcal{L}_2^2 \mathbf{v}_{\tau_k} \Delta\tau + \mathcal{L}_2^1 \mathbf{t}). \quad (4)$$

The new error modeling jointly considers time offset and relative poses with the support of velocity estimation. It fully utilizes the robot's motion information.

B. Estimation Problem Formulation

We then construct a joint optimization program using the aforementioned error representation. Observing that $\Delta\tau$ is unknown, we consider to take $\mathcal{L}_2^1 \mathbf{R}$ and $\Delta\tau \mathcal{L}_2^1 \mathbf{R}$ as two independent variables. Thus, we introduce

$$\begin{aligned} \mathbf{r}_p &= \text{vec}(\mathcal{L}_2^1 \mathbf{R}) \in \mathbb{R}^9, \\ \mathbf{r}_s &= \text{vec}(\Delta\tau \mathcal{L}_2^1 \mathbf{R}) \in \mathbb{R}^9. \end{aligned} \quad (5)$$

Since $\mathbf{R}\mathbf{x} = (\mathbf{x}^T \otimes \mathbf{I}_3) \text{vec}(\mathbf{R})$, where \otimes denotes Kronecker product, the error (4) can be rewritten in linear form

$$\begin{aligned} \mathbf{e}_{\tau_k} &= \mathbf{g}_{\tau_k} D_{\tau_k} + \mathcal{L}_1 \mathbf{t}_{\tau_k} - \mathcal{L}_2^1 \mathbf{t} \\ &\quad - (\mathcal{L}_2 \mathbf{t}_{\tau_k}^T \otimes \mathbf{I}_3) \mathbf{r}_p - (\mathcal{L}_2 \mathbf{v}_{\tau_k}^T \otimes \mathbf{I}_3) \mathbf{r}_s, \end{aligned} \quad (6)$$

where $\mathbf{g}_{\tau_k} = \mathcal{L}_1 \mathbf{R}_{\tau_k} \mathbf{b}_{\tau_k}$. Then, we introduce a homogenizing variable y with $y^2 = 1$ to rid a quadratic equation of linear terms. Defining a complete optimization variable \mathbf{x} and data matrix \mathbf{A}_{τ_k} , we obtain $\mathbf{e}_{\tau_k} = \mathbf{A}_k \mathbf{x}$ where

$$\begin{aligned} \mathbf{x} &= [\mathbf{r}_s^T, \mathbf{r}_p^T, y, \mathcal{L}_2^1 \mathbf{t}, D_{\tau_1}, \dots, D_{\tau_N}]^T \in \mathbb{R}^{(22+N)}, \\ \mathbf{A}_{\tau_k} &= [-\mathcal{L}_2 \mathbf{t}_{\tau_k}^T \otimes \mathbf{I}_3, -\mathcal{L}_2 \mathbf{v}_{\tau_k}^T \otimes \mathbf{I}_3, \mathcal{L}_1 \mathbf{t}_{\tau_k}, -\mathbf{I}_3, \\ &\quad \mathbf{0}_{3 \times 1}, \dots, \mathbf{g}_{\tau_k}, \dots, \mathbf{0}_{3 \times 1}] \end{aligned} \quad (7)$$

where N is the number of total measurements. Subsequently, we can derive a least square problem:

$$\begin{aligned} \mathbf{x}^* &= \arg \min_{\mathbf{x}} \sum_{k=1}^N (\mathbf{e}_{\tau_k})^T \mathbf{W}_{\tau_k} \mathbf{e}_{\tau_k} \\ &= \arg \min_{\mathbf{x}} \mathbf{x}^T \mathbf{Q} \mathbf{x} \\ \text{s.t. } & y^2 = 1, \mathcal{L}_2^1 \mathbf{R} \in SO(3), \end{aligned} \quad (8)$$

where \mathbf{W}_{τ_k} is the weight matrix and the data matrix $\mathbf{Q} = \sum_{k=1}^N \mathbf{A}_{\tau_k}^T \mathbf{W}_{\tau_k} \mathbf{A}_{\tau_k}$. Then, we eliminate the unconstrained distance and relative translation variables. Define the truncated optimization variable $\tilde{\mathbf{x}} = [\mathbf{r}_s^T, \mathbf{r}_p^T, y]^T$ and divide \mathbf{Q} into four blocks

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}, \quad (9)$$

where \mathbf{A} represent the block corresponding to $\tilde{\mathbf{x}}$. The unconstrained variables are marginalized by Schur Complement, and we obtain

$$\begin{aligned} \tilde{\mathbf{x}}^* &= \arg \min_{\tilde{\mathbf{x}}} \tilde{\mathbf{x}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{x}} \\ \text{s.t. } & y^2 = 1, \mathcal{L}_2^1 \mathbf{R} \in SO(3), \end{aligned}$$

where $\tilde{\mathbf{Q}} = \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$. In this way, the optimization variable only involves the norm-1 and rotation constraints.

IV. RELAXATION AND ITERATIVE OPTIMIZATION

In this section, we solve the formulated problem to recover the time offset and relative poses simultaneously. First, we introduce auxiliary variables and transform the original problem into a QCQP to handle the rotation constraints. Then, we apply semi-definite relaxation to solve the problem with global optimality. Last, we proposed an iterative algorithm, which recovers the time offset from coarse to fine.

A. QCQP Formulation

This section aims to handle the rotation constraint involved in \mathbf{r}_p and \mathbf{r}_s . Previous work [6] has pointed out the generalized rotation matrix constraints for scaled rotation $n\mathbf{R}$ consist of the following three types of constraints:

$$(n\mathbf{R}^T)(n\mathbf{R}) = n^2 \mathbf{I}_3, \quad (10)$$

$$(n\mathbf{R})(n\mathbf{R})^T = n^2 \mathbf{I}_3, \quad (11)$$

$$(n\mathbf{R})^{(i)} \times (n\mathbf{R})^{(j)} = n(n\mathbf{R})^{(k)}, \quad (12)$$

$$\forall (i, j, k) = (1, 2, 3), (2, 3, 1), (3, 1, 2),$$

The superscript $(\mathbf{M})^{(i)}$ denotes the i th column of matrix \mathbf{M} . In this paper, our faced rotation matrix constraints are corresponding to $n = y$ and $n = \Delta\tau$, respectively. Briaies [18, 19] indicates that all above constraints can be expressed in quadratic terms, which provides an approach to transform the matrix constraint into quadratic constraints. However, our original variable $\tilde{\mathbf{x}}$ does not include the time offset variable $\Delta\tau$, which means we can not construct constraint using $\tilde{\mathbf{x}}$ directly. Therefore, we introduce an auxiliary variable $\Delta\tau$ to obtain an lifted optimization variable $z =$

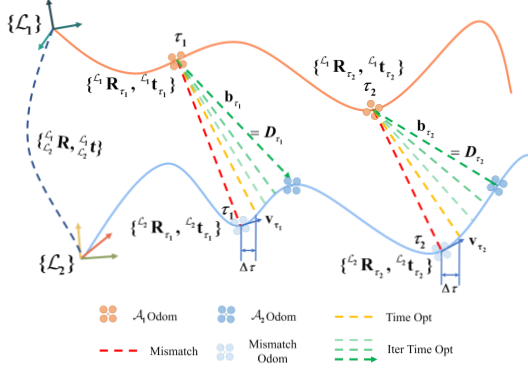


Fig. 2. Demonstration of the time offset estimation and mutual localization in bearing-based multi-robot system.

$[\mathbf{r}_s^T, \mathbf{r}_p^T, y, \Delta\tau]^T \in \mathbb{R}^{20}$. With \mathbf{z} , we can express the above constraints as

$$\mathbf{z}^T \mathbf{Q}_i \mathbf{z} = 0. \quad (13)$$

The detailed structure of \mathbf{Q}_i for $n = y$ and $n = \Delta\tau$ is listed in Table I. Besides the rotation matrix constraint, the involved constraints include $y^2 = 1$ and $(\Delta\tau)(\mathbf{r}_p) = (y)(\mathbf{r}_s)$, which can also be expressed as quadratic constraints in term as \mathbf{z} . Therefore, by lifting variable dimension, we transform the original problem with matrix constraint into a QCQP:

$$\begin{aligned} \mathbf{z}^* &= \min_{\mathbf{z}} \mathbf{z}^T \mathbf{Q}_0 \mathbf{z} \\ \text{s.t. } \mathbf{z}^T \mathbf{Q}_i \mathbf{z} &= g_i, i = 1, \dots, m. \end{aligned} \quad (14)$$

where $\mathbf{Q}_0 = \begin{bmatrix} \bar{\mathbf{Q}} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{20 \times 20}$ and m is the number of involved constraints.

B. SDR and Iterative Optimization

Since the convex programming can be solved in polynomial time and has strict global optimality, it is appealing to transform the QCQP into a convex problem losslessly. Due to $\mathbf{v}^T \mathbf{M} \mathbf{v} = \text{tr}(\mathbf{M} \mathbf{v} \mathbf{v}^T)$, we replace $\mathbf{z}^T \mathbf{z}$ with \mathbf{Z} and drop the nonconvex rank constraint by Shor's relaxation, resulting the relaxed problem:

$$\begin{aligned} \min_{\mathbf{Z}} \text{tr}(\mathbf{Q}_0 \mathbf{Z}) \\ \text{s.t. } \mathbf{Z} \succeq 0, \text{tr}(\mathbf{Q}_i \mathbf{Z}) = g_i, i = 1, \dots, m. \end{aligned}$$

If the solution of this problem \mathbf{Z}^* meets the condition $\text{rank}(\mathbf{Z}^*) = 1$, it means the relaxation is tight, and we can exactly recover the global minimizer of the original problem. The tightness is widely proven in robotics when the noise upon the data matrix is limited, and our subsequent experiment also verifies it. Therefore, given \mathbf{Z}^* , we can perform rank-1 decomposition to obtain \mathbf{z}^* and directly recover time offset and relative poses from it.

Up to this point, we have proposed a complete method to achieve mutual localization with time offset estimation, including a novel error model, problem formulation, and

lossless relaxation. However, the entire method builds on the constant velocity assumption, which is only valid when the time offset is limited. When the time drift between robots is sufficiently large, which is common in practice, the assumption may be violated and lead to incorrect estimation.

To address this issue, we propose an iterative method to recover the time offset from coarse to fine. After each completed estimate, we shift the timestamps of robot \mathcal{A}_2 's local odometry using the estimated time offset. Then the shifted odometry is used for the next estimation until convergence. By this iterative method, summarized in Algorithm 1, the range of time offset to be recovered should be enhanced. An example of the entire iteration process is illustrated in Fig.1, where \mathcal{A}_1 move along "IC" and \mathcal{A}_2 move with "RA". After several iterations, our method recovers the correct relative pose, even when the time offset is 3s.

V. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of our proposed method, including the non-iterative time optimization (NTO) method and iterative time optimization (ITO) method. In simulation experiments, the result shows that ITO has a better tolerance to time offset than NTO. Furthermore, by comparing the two methods with the method in Wang. [6], which lacks time offset estimation, our method presents better robustness and accuracy in general noisy cases. Note that we eliminate the anonymity consideration in Wang. [6] for a fair comparison. We then apply our methods in the real world to verify their practicality and robustness. All methods are implemented in MATLAB using cvx [20] as an SDP solver and run on a PC (Intel i5-9400F) with a single core.

Algorithm 1: Iterative Time Optimization

Input: Bearings from \mathcal{A}_1 to \mathcal{A}_2 : $bearing_1$,
Iteration times: N_{iter} ,
Robot \mathcal{A}_1 's local odometry: $odom_1$,
Robot \mathcal{A}_2 's local odometry: $odom_2$
Output: Total time offset T

begin
 $T \leftarrow 0$
for each $i \in N_{iter}$ **do**
 $\Delta\tau_i \leftarrow \text{TimeOpt}(bearing_1, odom_1, odom_2)$
 $T \leftarrow T + \Delta\tau_i$
if $\Delta\tau_i < \epsilon$ **then**
 break
 $odom_2 \leftarrow \text{ShiftOdom}(odom_2, \Delta\tau_i)$
return T

A. Simulation Experiment

To simulate data containing both poses and velocities, we follow the method in [21] to construct third-order SE3 B-splines for trajectory representation. For each robot, the corresponding B-spline is generated from 40 randomly sampled

TABLE I
CONSTRAINTS AND CONSTRAINT MATRIX

Constraint	$(n_{\mathcal{L}_2^1} \mathbf{R})^T (n_{\mathcal{L}_2^1} \mathbf{R}) = n^2 \mathbf{I}_3$	$(n_{\mathcal{L}_2^1} \mathbf{R}) (n_{\mathcal{L}_2^1} \mathbf{R})^T = n^2 \mathbf{I}_3$	$(n_{\mathcal{L}_2^1} \mathbf{R})^{(i)} \times (n_{\mathcal{L}_2^1} \mathbf{R})^{(j)} = n (n_{\mathcal{L}_2^1} \mathbf{R})^{(k)}$
\mathbf{Q}_i	$\begin{bmatrix} \mathbf{0}_{9 \times 9} & \mathbf{0}_{9 \times 9} & \mathbf{0}_{9 \times 1} & \mathbf{0}_{9 \times 1} \\ \mathbf{0}_{9 \times 9} & \mathbf{J}_3 \otimes \mathbf{I}_3 & \mathbf{0}_{9 \times 1} & \mathbf{0}_{9 \times 1} \\ \mathbf{0}_{1 \times 9} & \mathbf{0}_{1 \times 9} & \alpha & 0 \\ \mathbf{0}_{1 \times 9} & \mathbf{0}_{1 \times 9} & 0 & \beta \end{bmatrix}$	$\begin{bmatrix} \mathbf{0}_{9 \times 9} & \mathbf{0}_{9 \times 9} & \mathbf{0}_{9 \times 1} & \mathbf{0}_{9 \times 1} \\ \mathbf{0}_{9 \times 9} & \mathbf{I}_3 \otimes \mathbf{J}_3 & \mathbf{0}_{9 \times 1} & \mathbf{0}_{9 \times 1} \\ \mathbf{0}_{1 \times 9} & \mathbf{0}_{1 \times 9} & \alpha & 0 \\ \mathbf{0}_{1 \times 9} & \mathbf{0}_{1 \times 9} & 0 & \beta \end{bmatrix}$	$\begin{bmatrix} \mathbf{U} & \mathbf{0}_{9 \times 9} & \mathbf{0}_{9 \times 1} & \mathbf{0}_{9 \times 1} \\ \mathbf{0}_{9 \times 9} & \mathbf{V} & \mathbf{u} & \mathbf{v} \\ \mathbf{0}_{1 \times 9} & \mathbf{0}_{1 \times 9} & 0 & 0 \\ \mathbf{0}_{1 \times 9} & \mathbf{0}_{1 \times 9} & 0 & 0 \end{bmatrix}$
$n = y$	$\alpha = -tr(\mathbf{J}_3), \beta = 0$	$\alpha = -tr(\mathbf{J}_3), \beta = 0$	$\mathbf{U} = \mathbf{0}_{9 \times 9}, \mathbf{V} = \mathbf{E}_{ij} \otimes [\mathbf{j}_3]_{\times}$ $\mathbf{u} = \mathbf{e}_k \otimes \mathbf{j}_3, \mathbf{v} = \mathbf{0}_{9 \times 1}$
$n = \Delta\tau$	$\alpha = 0, \beta = -tr(\mathbf{J}_3)$	$\alpha = 0, \beta = -tr(\mathbf{J}_3)$	$\mathbf{U} = \mathbf{0}_{9 \times 9}, \mathbf{V} = \mathbf{E}_{ij} \otimes [\mathbf{j}_3]_{\times}$ $\mathbf{u} = \mathbf{0}_{9 \times 1}, \mathbf{v} = \mathbf{e}_k \otimes \mathbf{j}_3$

Table.I: Generalized rotation matrix constraints and their corresponding constraint matrix \mathbf{Q}_i for $n = y$ or $\Delta\tau$. \mathbf{J}_3 and \mathbf{j}_3 are matrix and vector will all element 1. \mathbf{E}_{ij} and \mathbf{e}_i are (i, j) th coordinate matrix and i th coordinate vector. $[\mathbf{v}]_{\times}$ denotes the skew symmetric matrix of a vector \mathbf{v} .

control points. To guarantee the smoothness, the distance between two control points does not exceed 0.5m and the time interval of B-spline is set to 1 second. We then generate synthetic data using these B-splines. Firstly, we sample 4000 poses at 5ms intervals as the robot's local odometry. Then, we produce 200 bearing measurements for each pair of robots. To control noise levels, we add Gaussian noise into bearing measurements with different standard deviations σ

1) *Tolerance and Optimality Comparison:* In this experiment, we aim to evaluate the influence of iterative optimization in time offset estimation in both tolerance level and optimality. We add no noise in bearing measurements and change the time offset of two robots' trajectories on purpose from 0s to 1.4s with an interval of 0.02s. We perform 100 Monte-Carlo simulations with randomly generated B-spline for each determined time offset. The time offset estimation, cost, and the rank of the solution are shown in Fig.3. It shows the NTO can not recover the correct time offset when it exceeds around 0.4s. The reason is that the uniform velocity assumption can not hold when the time offset is too large. In contrast, the ITO method can maintain optimality even for time offsets exceeding 1s and still precisely estimate the time offset. It suggests that the from-coarse-to-fine algorithm effectively extends the applicability of the uniform velocity assumption in extreme situations. In summary, ITO outperforms NTO in terms of time tolerance and optimality, especially in more challenging scenarios (large time offset), which validates our perspective in Sec.IV-B.

2) *Estimation Accuracy Comparison:* In this experiment, we aim to evaluate our method in more general cases for comprehensive assessments. We compare our methods (NTO and ITO) with Wang. [6] in two dimensions, times offset and noise level. We change the times offset (from 0 to 1s with 0.1s interval) and noise (from 0 to 0.1 with 0.01 interval) respectively, and perform 100 trials for each configuration, resulting in 11000 data. Two metrics, the mean estimation error and the meantime offset estimation error, are considered in this experiment. The results are shown in Fig.4 and

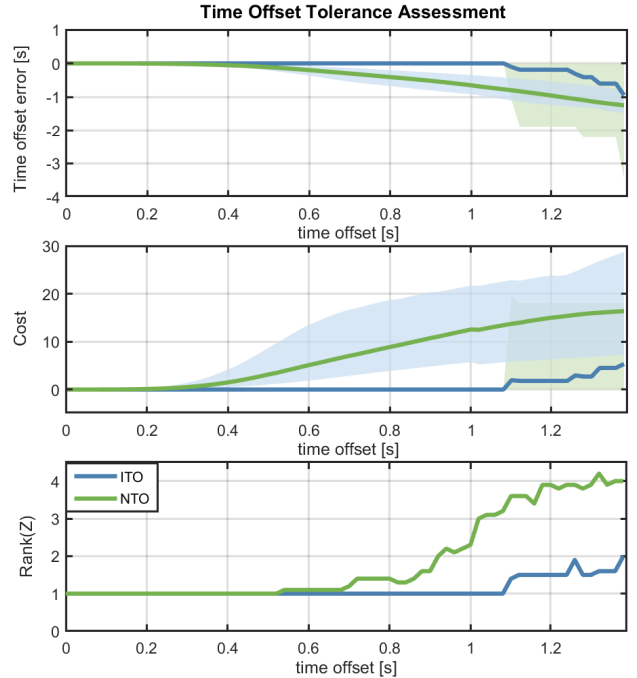


Fig. 3. The change of time offset estimation error, objective cost, and rank of solution \mathbf{Z}^* under different time offsets between robots.

Fig.5, respectively. Note that since there is no time offset estimation in Wang. [6], we take zeros as its estimation result to compute the error.

We obtained two findings from the results of these experiments. First, as the time offset and noise level change, the error distributions of both pose estimation and time offset estimation show a clear consistency. It means better time offset estimation can improve mutual localization's performance, highlighting the significance of time synchronization. Second, the results show that the dark region of ITO is larger than NTO, while that of Wang [6] is the smallest, especially in areas where time offset is large (> 0.5). It means both NTO and ITO outperform the existing method, with ITO emerging as the superior algorithm among the three algorithms. This experiment presents how our proposed

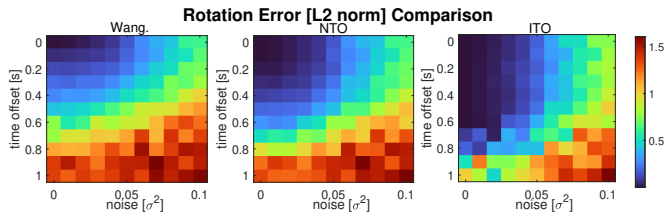


Fig. 4. Distribution of relative rotation estimation error with change of time drift and noise level.

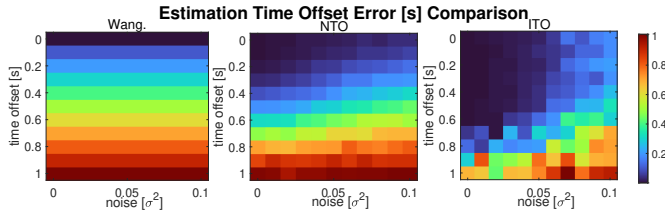


Fig. 5. Distribution of time offset estimation error with change of time drift and noise level.

method extends the applicability of mutual localization in time-drifted and noisy cases.

B. Real-world Experiments

In this part, we apply our method to real-world data to evaluate the practicality and robustness. The quadrotor platforms and a snapshot of our experiments are shown in Fig.6. Quadrotors are equipped with LED arrays and infrared cameras for mutual detection. We allow robots to move along pre-determined 3D trajectories and obtain stable inter-robot bearing measurements during flight. We conducted totally five experiments with different trajectory patterns, including

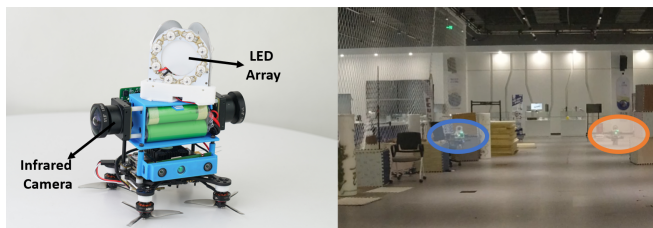


Fig. 6. Platform and snapshot in real-world experiment.

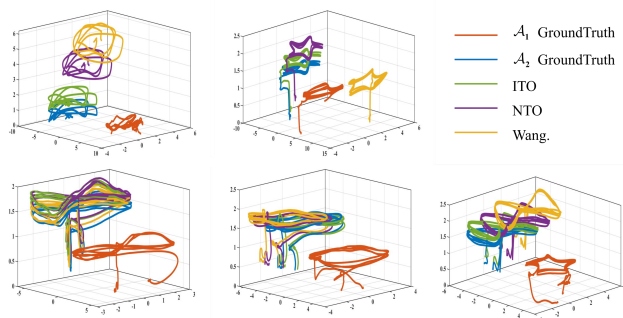


Fig. 7. Comparisons of estimated trajectories in real-world experiments. Two robots' trajectories are shown in a common frame.

TABLE II
RESULTS OF REAL-WORLD EXPERIMENTS.

Method	Cost	Trans(m)	Rot($^{\circ}$)	Estimation Time offset(s)	Runtime(s)
Wang.	65.524	2.82	28.26	-	0.418
	51.776	6.52	144.19	-	0.441
	18.184	0.17	1.16	-	0.436
	15.516	0.72	10.78	-	0.465
	92.19	1.11	21.57	-	0.465
NTO	47.409	1.834	18.34	0.374	0.408
	30.896	0.448	27.43	0.659	0.438
	5.305	0.187	1.85	0.698	0.437
	10.492	0.536	7.29	0.377	0.432
	57.251	0.514	12	0.545	0.464
ITO	3.994	0.489	4.4	0.905	2.064
	5.393	0.163	3.39	0.962	1.687
	1.799	0.202	1.53	0.963	1.75
	2.268	0.049	4.01	0.869	2.177
	4.22	0.142	2.87	0.961	2.337

pentagons, figure-eight, and squares. In each experiment, we fix the time offset between trajectories as 1s and utilize NOKOV motion capture system collect the ground truth.

With collected data, we compare the above-mentioned three methods in multiple dimensions: optimization cost, estimation error, estimated time offset, and runtime. As Table.II shows Wang. [6] always converge to a larger cost and obtain egregiously large error in translation and rotation. The erroneous estimation comes from the incorrect correspondence between bearing measurement and observed robot's local odometry. Compared with other methods, ITO consistently obtains the minimum cost in all experiments with the most or second most small estimation error. Note that, due to noise, obtaining the minimum cost does not mean obtaining the most accurate estimation. In the aspect of time offset estimation, ITO consistently recovers the accurate time offset while NTO has a significant error. It is clear that ITO needs more computation due to iteration. But it is acceptable since time synchronization can run as a background program periodically in multi-robot systems. The aligned trajectories using estimated relative pose by three methods are illustrated in Fig.7. It shows ITO can still work properly to recover a correct relative pose even with time-drifted odometry data, while other methods fail. These experiments verify the practicality and robustness of our proposed method.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a complete framework to simultaneously achieve time synchronization and mutual localization. The joint estimation of time offset and relative poses significantly improves the performance of mutual localization when there is a certain time drift between robots. Extensive experiments on simulated and real-world data show the outperforming accuracy of our method compared with existing methods. In the future, we will apply our mutual localization method in various multi-robot applications to facilitate efficient collaboration among multiple robots.

REFERENCES

- [1] Y. Gao, Y. Wang, X. Zhong, T. Yang, M. Wang, Z. Xu, Y. Wang, Y. Lin, C. Xu, and F. Gao, "Meeting-merging-mission: A multi-robot

- coordinate framework for large-scale communication-limited exploration,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 13 700–13 707.
- [2] H. Sugiyama, T. Tsujioka, and M. Murata, “Coordination of rescue robots for real-time exploration over disaster areas,” in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2008, pp. 170–177.
- [3] X. S. Zhou and S. I. Roumeliotis, “Determining 3-d relative transformations for any combination of range and bearing measurements,” *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 458–474, 2012.
- [4] A. Franchi, G. Oriolo, and P. Stegagno, “Mutual localization in multi-robot systems using anonymous relative measurements,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1302–1322, 2013.
- [5] T. Nguyen, K. Mohta, C. J. Taylor, and V. Kumar, “Vision-based multi-mav localization with anonymous relative measurements using coupled probabilistic data association filter,” in *2020 IEEE International Conference on Robotics and Automation*.
- [6] Y. Wang, X. Wen, L. Yin, C. Xu, Y. Cao, and F. Gao, “Certifiably optimal mutual localization with anonymous bearing measurements,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9374–9381, 2022.
- [7] D. Mills, J. Martin, J. Burbank, and W. Kasch, “Network time protocol version 4: Protocol and algorithms specification,” Tech. Rep., 2010.
- [8] J. C. Eidson, M. Fischer, and J. White, “Ieee-1588™ standard for a precision clock synchronization protocol for networked measurement and control systems,” in *Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting*, 2002, pp. 243–254.
- [9] T. Qin and S. Shen, “Online temporal calibration for monocular visual-inertial systems,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3662–3669.
- [10] A. Howard, “Multi-robot simultaneous localization and mapping using particle filters,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.
- [11] I. V. Melnyk, J. A. Hesch, and S. I. Roumeliotis, “Cooperative vision-aided inertial navigation using overlapping views,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 936–943.
- [12] J. Dong, E. Nelson, V. Indelman, N. Michael, and F. Dellaert, “Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach,” in *2015 IEEE International Conference on Robotics and Automation*.
- [13] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, “Door-slam: Distributed, online, and outlier resilient slam for robotic teams,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1656–1663, 2020.
- [14] P. Stegagno, M. Cagnetti, A. Franchi, and G. Oriolo, “Mutual localization using anonymous bearing measurements,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 469–474.
- [15] Z. Yang, J. Pan, and L. Cai, “Adaptive clock skew estimation with interactive multi-model kalman filters for sensor networks,” in *2010 IEEE International Conference on Communications*. IEEE, 2010, pp. 1–5.
- [16] F. Shi, H. Li, S. X. Yang, X. Tuo, and M. Lin, “Novel maximum likelihood estimation of clock skew in one-way broadcast time synchronization,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, pp. 9948–9957, 2019.
- [17] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.
- [18] J. Briales and J. Gonzalez-Jimenez, “Convex global 3d registration with lagrangian duality,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4960–4969.
- [19] J. Briales, L. Kneip, and J. Gonzalez-Jimenez, “A certifiably globally optimal solution to the non-minimal relative pose problem,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 145–154.
- [20] M. Grant and S. Boyd, “Cvx: Matlab software for disciplined convex programming, version 2.1,” 2014.
- [21] P. Geneva, K. Eickenhoff, W. Lee, Y. Yang, and G. Huang, “Openvins: A research platform for visual-inertial estimation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.