Consistent Unscented Incremental Smoothing for Multi-robot Cooperative Target Tracking

Guoquan Huang[a], Michael Kaess[b], John J. Leonard[c]

[a]*Department of Mechanical Engineering, University of Delaware, Newark, DE 19716.*
[b]*Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213.*
[c]*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139.*

Abstract

In this paper, we study the problem of multi-robot cooperative target tracking, where a team of mobile robots cooperatively localize themselves and track (multiple) targets using their onboard sensor measurements as well as target stochastic kinematic information, and which is hence termed cooperative localization and target tracking (CLATT). A novel efficient, consistent, unscented incremental smoothing (UIS) algorithm is introduced. The key idea of the proposed approach is that we employ unscented transform to numerically compute Jacobians so as to attain *reduced* linearization errors, while further imposing appropriate constraints on unscented transform to ensure *correct* observability properties for the *incrementally-linearized* system. In particular, for the first time we analyze the observability properties of the optimal batch maximum a posteriori (MAP)-based CLATT system, and show that the Fisher information (Hessian) matrix without prior has a nullspace of dimension *three*, corresponding to the global state information. However, this may not be the case when the Jacobians (and thus the Hessian) are computed *canonically* by the standard unscented transform, thus negatively impacting the estimation performance. To address this issue, we formulate an observability-constrained unscented transform, and find its closed-from solution as the projection of the canonical unscented Jacobian (i.e., computed by the standard unscented transform) onto an appropriate observable subspace such that the resulting Hessian has a nullspace of correct dimen-

Email addresses: ghuang@udel.edu (Guoquan Huang), kaess@cmu.edu (Michael Kaess), jleonard@mit.edu (John J. Leonard)

sions. The proposed approach is tested extensively through Monte Carlo simulations as well as a real-world experiment, and is shown to outperform the state-of-the-art incremental smoothing algorithm.

Keywords: Multi-robot systems, cooperative localization, target tracking, nonlinear estimation, maximum a posteriori (MAP), incremental smoothing and mapping (iSAM), unscented transform, parameter observability

1. Introduction

Mobile robot networks have become increasingly popular due to their ability to sense, communicate, and interact with the physical world, and offer a broad spectrum of applications. For example, a team of mobile robots have been used for inspection of nuclear power plants [1], aerial surveillance [2], search and rescue [3], and underwater or space exploration [4]. In order for multi-robot teams to navigate safely and thus successfully perform these tasks, it is essential for robots to determine their positions and orientations (poses) precisely. In GPS-denied environments and in absence of robust landmarks, a team of robots can localize by sharing relative position measurements and jointly estimating their poses, which is the problem of cooperative localization (CL) [5–7]. Current approaches, either centralized or distributed, solving this problem typically assume *static* environments, i.e., with no moving objects in the workspace or without considering them, and are based on the extended Kalman filtering (EKF) [7], maximum likelihood estimation (MLE) [8], maximum a posteriori (MAP) estimation [9], or particle filtering (PF) [10].

However, in practice a team of mobile robots often work in a *dynamic* environment with various moving objects. For example, robots co-work with humans in office buildings or museums [11]. In such cases, it is necessary for the robots to simultaneously determine their poses *and* the kinematic states of moving objects (targets), such as positions and velocities. This is in part due to the fact that jointly estimating the robots and targets results in better accuracy for the robots' position estimates, in comparison to localizing the robots by ignoring the targets (which is the case for most existing CL approaches) [12]. This is the problem of cooperative localization and target tracking (CLATT), and arises in many robotic applications such as surveillance [13, 14]. In this paper, we study in-depth the CLATT problem and develop a new efficient consistent incremental estimation algorithm.

Although many different algorithms are available for solving the CLATT problem, among them the EKF remains a popular choice primarily due to its relatively low processing requirements [12]. However, its performance depends on the magnitude of the linearization errors. In order to reduce the linearization errors, the unscented transform (statistical linearization) employed by the unscented Kalman filter (UKF) [15] is often used, which deterministically samples the nonlinear model around the current state estimate and employs linear regression to improve the accuracy of the linear approximation [16]. Nevertheless, any (explicit or implicit) linearization-based filtering approach marginalizes all but the current state and hence is unable to correct linearization errors involving previous states. This can result in large linearization errors, thus degrading the filter's performance.

For this reason, smoothing approaches, either in batch or incremental fashion, have recently become prevalent in robotics (see [17–22] and references therein). In particular, a batch-MAP estimator [23] or bundle adjustment [24] computes the estimates for the states at all time steps using all available measurements. This allows continuous relinearization around all the states, which can greatly reduce the linearization errors. However, since the size of the state vector in the batch-MAP estimator increases continuously over time, the processing and memory requirements can easily become too high to perform in real time. To reduce the computational complexity of the batch estimation, an incremental smoothing and mapping (iSAM) algorithm [19] has been developed particularly for simultaneous localization and mapping (SLAM) problems. The key idea of the iSAM is employing a factorization-updating method to allow reusing the information-matrix factorization available from previous time steps, while computationally demanding procedures, such as relinearization and batch factorization, are only performed intermittently. Although the iSAM was shown to perform well in many *static* robotic mapping applications [19], it still may suffer from large linearization errors, and yet has not been investigated in *dynamic* environments to incorporate target tracking in a CL estimation framework.

To address the aforementioned issues (i.e., reducing linearization errors and integrating target tracking), in this paper we introduce an unscented incremental smoothing (UIS) algorithm for the CLATT problem. In particular, the proposed UIS computes the Jacobians, and thus the Hessian, by unscented transform, which has been shown to reduce linearization errors [15]. However, observability analysis of the CLATT system reveals that the Fisher information (Hessian) matrix computed numerically via the stan-

dard unscented transform may have a nullspace that has different dimensions from that of the optimal (up to linearization errors) batch-MAP estimator. This implies that spurious (nonexistent) information is gained from the measurements, which can lead to estimation inconsistency[1] or even divergence (see [18]). To overcome this issue, the proposed UIS explicitly enforces correct observability constraints on the unscented transform (linear regression), whose optimal solution turns out to projecting the canonical unscented Jacobians computed by the standard unscented transform onto the desired observable subspace. At this point, we should stress that apart from the CLATT problem considered in this paper, the proposed UIS methodology can be applicable for a large class of nonlinear estimation problems in robotics, such as CL, SLAM, and cooperative SLAM (C-SLAM). Note also that, in comparison to our previous conference publication [26], in this paper we study in depth the observability properties of the batch-MAP based CLATT system, present in detail the derivations of the proposed UIS, and thoroughly validate its superior performance over the state-of-the-art iSAM algorithm, both in Monte-Carlo simulations and in a real-world experiment.

The remainder of the paper is organized as follows: After formulating the CLATT problem in the next section, we describe both the batch and incremental smoothing algorithms used to solve the problem in Section 3. In Section 4 we perform the parameter observability analysis for the optimal batch-MAP based CLATT system and analytically show the nullspace (unobservable subspace) of the Fisher information (Hessian) matrix, which is different from that of the standard incremental smoothing based system. Based on this analysis, in Section 5 we present in detail the proposed UIS algorithm which employs the unscented transform to numerically compute Jacobians and moreover introduces a key projection operation when computing the measurement Jacobians. The performance of the proposed UIS is validated through both Monte Carlo simulations in Section 6 and a real-world experiment in Section 7. Finally, Section 8 outlines the main conclusions of this work, as well as possible future research directions.

[1] As defined in [25], a state estimator is consistent if the estimation errors are zero-mean (i.e., unbiased), and have covariance equal to the true covariance.

2. The CLATT Problem

A typical CLATT scenario can be described as follows: A team of mobile robots that are equipped with proprioceptive (e.g., an odometer or an inertial measurement unit) and exteroceptive (e.g., a laser scanner or a camera) sensors, move in a plane and track (multiple) targets. In contrast to the robots, the targets generally do not have any onboard sensor, while their motion is assumed to follow a known stochastic kinematic model such as constant-velocity motion model. A tracking robot measures distance and/or bearing to the targets as well as the other ones in the team. Using all the available proprioceptive and exteroceptive measurements, we aim to estimate the state trajectory of both robots and targets at each time step. Note that in this work we do not assume communication among robots, which, however, is not a necessary assumption and can be integrated into the proposed estimation framework (see Section 5). For example, each robot can communicate its estimates or measurements to the others so that more information can be utilized by the estimator.

The state vector of the CLATT at time-step k contains all the robots' poses (positions and orientations) and the targets' kinematic states such as position, velocity, acceleration, etc.:

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_{R_1}(k) \\ \vdots \\ \mathbf{x}_{R_M}(k) \\ \mathbf{x}_{T_1}(k) \\ \vdots \\ \mathbf{x}_{T_N}(k) \end{bmatrix} \tag{1}$$

where

$$\mathbf{x}_{R_i} := \begin{bmatrix} \mathbf{p}_{R_i}^T & \phi_{R_i} \end{bmatrix}^T = \begin{bmatrix} x_{R_i} & y_{R_i} & \phi_{R_i} \end{bmatrix}^T \ , i = 1, \ldots, M$$

denotes the ith robot's pose (position and orientation), and

$$\mathbf{x}_{T_i} := \begin{bmatrix} \mathbf{p}_{T_i}^T & \mathbf{d}_{T_i}^T \end{bmatrix}^T = \begin{bmatrix} x_{T_i} & y_{T_i} & \dot{x}_{T_i} & \dot{y}_{T_i} & \ddot{x}_{T_i} & \ddot{y}_{T_i} & \cdots \end{bmatrix}^T \ , i = 1, \ldots, N$$

denotes the ith target's kinematic states, including the target's position, $\mathbf{p}_{T_i}$, and the higher-order time derivatives of the target's position, $\mathbf{d}_{T_i}$. Stacking all the states in the time interval $[0, k]$ yields the state trajectory which we

want to estimate at every time step:

$$\mathbf{x}(0:k) = \begin{bmatrix} \mathbf{x}(0) \\ \mathbf{x}(1) \\ \vdots \\ \mathbf{x}(k) \end{bmatrix} \tag{2}$$

In what follows, we describe the motion and measurement models that will be used in the smoothing algorithms in the next section.

2.1. Robot motion model

Each tracking robot is equipped with an odometer, whose measurement serves as the control input to propagate the robot pose based on the following motion model:

$$\mathbf{p}_{R_i}(k+1) = \mathbf{p}_{R_i}(k) + \mathbf{C}(\phi_{R_i}(k)) \, {}^{R_k}\mathbf{p}_{R_{i,k+1}} \tag{3}$$

$$\phi_{R_i}(k+1) = \phi_{R_i}(k) + {}^{R_k}\phi_{R_{i,k+1}} \tag{4}$$

where $\mathbf{C}(\cdot)$ denotes the 2×2 rotation matrix, and $\mathbf{u}_i(k) := {}^{R_k}\mathbf{x}_{R_{i,k+1}} := \begin{bmatrix} {}^{R_k}\mathbf{p}_{R_{i,k+1}} \\ {}^{R_k}\phi_{R_{i,k+1}} \end{bmatrix}$ is the true odometry (control input), i.e., the robot's motion between time-steps k and $k+1$, expressed with respect to the robot's frame at time-step k, $\{R_k\}$. The corresponding odometry measurement, $\mathbf{u}_{m_i}(k)$, is commonly assumed to be corrupted by zero-mean white Gaussian noise, $\mathbf{w}_i(k) = \mathbf{u}_i(k) - \mathbf{u}_{m_i}(k)$, with covariance $\mathbf{Q}_{R_i}(k)$. Clearly this motion model is nonlinear and can be written in the following generic form:

$$\mathbf{x}_{R_i}(k+1) = \mathbf{f}\left(\mathbf{x}_{R_i}(k), \mathbf{u}_{m_i}(k) + \mathbf{w}_i(k)\right) \tag{5}$$

Linearization of the above functions (3) and (4) at the current state estimates, $\hat{\mathbf{x}}(0:k+1|k+1)$, yields the following Jacobians with respect to the

entire state trajectory (2) and noise, respectively:[2]

$$\mathcal{F}_{R_i}(k) = \begin{bmatrix} \mathbf{0} & \cdots & -\mathbf{\Phi}_{R_i}(k) & \mathbf{I}_3 & \cdots & \mathbf{0} \end{bmatrix} \tag{6}$$

$$\mathbf{G}_{R_i}(k) = \begin{bmatrix} \mathbf{C}(\hat{\phi}_{R_i}(k|k)) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{7}$$

where

$$\mathbf{\Phi}_{R_i}(k) = \begin{bmatrix} \mathbf{I}_2 & \mathbf{\Gamma}\left(\hat{\mathbf{p}}_{R_i}(k+1|k+1) - \hat{\mathbf{p}}_{R_i}(k|k)\right) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{8}$$

$$\mathbf{\Gamma} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{9}$$

Note that the above model (3) and (4) is general and encompasses all the motion models used in practice, such as the unicycle and Ackerman models. For instance, if the unicycle model is used, and both the linear velocity, $v_i(k)$, and the rotational velocity, $\omega_i(k)$, are constant during each propagation interval, we have ${}^{R_k}\mathbf{x}_{R_{i,k+1}} = \begin{bmatrix} v_i(k)\delta t \\ 0 \\ \omega_i(k)\delta t \end{bmatrix}$, where δt is the sampling period. Substitution in (3)-(4) yields the following propagation equations that are commonly seen in the literature:

$$\mathbf{p}_{R_i}(k+1) = \mathbf{p}_{R_i}(k) + \begin{bmatrix} v_i(k)\delta t \cos(\phi_{R_i}(k)) \\ v_i(k)\delta t \sin(\phi_{R_i}(k)) \end{bmatrix} \tag{10}$$

$$\phi_{R_i}(k+1) = \phi_{R_i}(k) + \omega_i(k)\delta t \tag{11}$$

2.2. Target kinematic model

We consider the case where each target moves randomly but assumes a *known* stochastic kinematic model that describes its motion (e.g., constant acceleration, or constant velocity). In particular, the discrete-time

[2]Throughout the paper, $(\ell|j)$ refers to the estimate of a quantity at time-step ℓ, after all measurements up to time-step j have been processed. $\hat{\mathbf{x}}$ is used to denote the estimate of random variable $\mathbf{x}$, and $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ is the error in this estimate. $\bar{\mathbf{x}}$ and $\bar{\mathbf{P}}_{\mathbf{xx}}$ denote the sample mean and covariance of sample points $\mathcal{X}_i$, drawn from the pdf $p(\mathbf{x})$. $\bar{\mathbf{P}}_{\mathbf{xy}}$ denotes the sample cross-correlation between the sets of samples $\mathcal{X}_i$ and $\mathcal{Y}_i$, drawn from the pdfs of $\mathbf{x}$ and $\mathbf{y}$, respectively. Finally, $\mathbf{0}_{m\times n}$ is the $m \times n$ matrix of zeros, and $\mathbf{I}_n$ is the $n \times n$ identity matrix.

state propagation is given by the following linear equation:

$$\mathbf{x}_{T_i}(k+1) = \mathbf{\Phi}_{T_i}(k)\mathbf{x}_{T_i}(k) + \mathbf{G}_{T_i}(k)\mathbf{w}_{T_i}(k) \tag{12}$$

where $\mathbf{w}_{T_i}$ is zero-mean white Gaussian noise with covariance $\mathbf{Q}_{T_i}$. The state transition matrix, $\mathbf{\Phi}_{T_i}$, and the process noise Jacobian, $\mathbf{G}_{T_i}$, that appear in the preceding expression depend on the particular stochastic motion model used [25]. We will make no further assumptions on these matrices other than that their values are known. Similarly, the "Jacobian" matrix of (12) with respect to the state trajectory is given by:

$$\mathcal{F}_{T_i}(k) = \begin{bmatrix} \mathbf{0} & \cdots & -\mathbf{\Phi}_{T_i}(k) & \mathbf{I}_{\dim(\mathbf{x}_{T_i})} & \cdots & \mathbf{0} \end{bmatrix} \tag{13}$$

2.3. Measurement model

In this work the tracking robots measure distance and/or bearing to the targets *and* the other robots in the team. In what follows we present both distance and bearing measurement equations, while a particular application may have any combination of these two types of measurements. For example, we consider the range-only measurements in the simulation tests (see Section 6), while the bearing-only observations are used in the experiment (see Section 7).

2.3.1. Robot-to-robot measurements

At time-step k robot i measures distance and bearing to robot j, which is given by:

$$\begin{aligned} \mathbf{z}_{R_iR_j}(k) &= \begin{bmatrix} \sqrt{(x_{R_j}(k) - x_{R_i}(k))^2 + (y_{R_j}(k) - y_{R_i}(k))^2} \\ \operatorname{atan2}\big((y_{R_j}(k) - y_{R_i}(k)), (x_{R_j}(k) - x_{R_i}(k))\big) - \phi_{R_i}(k) \end{bmatrix} + \mathbf{n}_{R_{ij}}(k) \\ &=: \mathbf{h}\left(\mathbf{x}_{R_i}(k), \mathbf{x}_{R_j}(k)\right) + \mathbf{n}_{R_{ij}}(k) \end{aligned} \tag{14}$$

where $\mathbf{n}_{R_{ij}}$ is zero-mean white Gaussian measurement noise with covariance $\mathbf{R}_{R_{ij}}$. For the use of linearized smoothing algorithms presented in the next section, we linearize (14) with respect to the state trajectory (2), at the current state estimate, $\hat{\mathbf{x}}(0:k|k)$, and obtain the following measurement Jacobian matrix:

$$\mathcal{H}_{R_{ij}}(k) = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{H}_{R_i}(k) & \cdots & \mathbf{H}_{R_j}(k) & \cdots & \mathbf{0} \end{bmatrix} \tag{15}$$

where

$$\mathbf{H}_{R_i}(k) = \begin{bmatrix} \frac{\hat{\mathbf{p}}_{R_i}^T(k|k) - \hat{\mathbf{p}}_{R_j}^T(k|k)}{||\hat{\mathbf{p}}_{R_j}^T(k|k) - \hat{\mathbf{p}}_{R_i}^T(k|k)||} & \mathbf{0} \\ \frac{(\hat{\mathbf{p}}_{R_i}^T(k|k) - \hat{\mathbf{p}}_{R_j}^T(k|k))\mathbf{\Gamma}^T}{||\hat{\mathbf{p}}_{R_j}^T(k|k) - \hat{\mathbf{p}}_{R_i}^T(k|k)||^2} & -1 \end{bmatrix} \tag{16}$$

$$\mathbf{H}_{R_j}(k) = \begin{bmatrix} \frac{\hat{\mathbf{p}}_{R_j}^T(k|k) - \hat{\mathbf{p}}_{R_i}^T(k|k)}{||\hat{\mathbf{p}}_{R_j}^T(k|k) - \hat{\mathbf{p}}_{R_i}^T(k|k)||} & \mathbf{0} \\ \frac{(\hat{\mathbf{p}}_{R_j}^T(k|k) - \hat{\mathbf{p}}_{R_i}^T(k|k))\mathbf{\Gamma}^T}{||\hat{\mathbf{p}}_{R_j}^T(k|k) - \hat{\mathbf{p}}_{R_i}^T(k|k)||^2} & \mathbf{0} \end{bmatrix} \tag{17}$$

2.3.2. Robot-to-target measurements

Similarly, the robot-to-target distance and bearing measurement at time-step k is given by:

$$\begin{aligned} \mathbf{z}_{R_iT_j}(k) &= \begin{bmatrix} \sqrt{(x_{T_j}(k) - x_{R_i}(k))^2 + (y_{T_j}(k) - y_{R_i}(k))^2} \\ \text{atan2}\left((y_{T_j}(k) - y_{R_i}(k)), (x_{T_j}(k) - x_{R_i}(k))\right) - \phi_{R_i}(k) \end{bmatrix} + \mathbf{n}_{T_{ij}}(k) \\ &=: \mathbf{h}\left(\mathbf{x}_{R_i}(k), \mathbf{x}_{T_j}(k)\right) + \mathbf{n}_{T_{ij}}(k) \end{aligned} \tag{18}$$

where $\mathbf{n}_{T_{ij}}$ is zero-mean white Gaussian measurement noise with covariance $\mathbf{R}_{T_{ij}}$. Analogously, the measurement Jacobian of (18), $\mathcal{H}_{T_{ij}}$, can be obtained by replacing $\hat{\mathbf{p}}_{R_j}$ in (15) by $\hat{\mathbf{p}}_{T_j}$, and has the following sparse structure:

$$\mathcal{H}_{T_{ij}}(k) = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{H}_{R_i}(k) & \cdots & \mathbf{H}_{T_j}(k) & \cdots & \mathbf{0} \end{bmatrix} \tag{19}$$

3. Smoothing Algorithms

In this section, we present the batch and incremental smoothing algorithms available for the CLATT problem. In particular, the optimal (up to linearization errors) batch-MAP estimator [23] along with its QR-based solver is first described, serving as the benchmark for the incremental smoothing algorithms, and subsequently the key idea of the iSAM algorithm [19] is explained in the CLATT context.

3.1. Batch-MAP estimator

Our objective is to estimate the entire state trajectory using all the available information, which includes: (i) the prior information about the initial state, described by a Gaussian pdf with mean $\hat{\mathbf{x}}(0|0)$ and covariance $\mathbf{P}(0|0)$,

(ii) the motion information [see (3), (4), and (12)], and (iii) the sensor measurements [see (14) and (18)]. To this end, the batch-MAP estimator is employed to determine the entire trajectory estimate $\hat{\mathbf{x}}(0:k|k)$ that maximizes the following posterior pdf:

$$p\left(\mathbf{x}(0:k)|\mathbf{z}(0:k)\right) \propto p\left(\mathbf{x}(0)\right) \times \tag{20}$$
$$\prod_{\kappa=0}^{k-1}\left[\prod_{i=1}^{M} p(\mathbf{x}_{R_i}(\kappa+1)|\mathbf{x}_{R_i}(\kappa)) \prod_{j=1}^{N} p(\mathbf{x}_{T_j}(\kappa+1)|\mathbf{x}_{T_j}(\kappa))\right] \times$$
$$\prod_{\kappa=0}^{k}\left[\prod_{\{i,j\}\in\mathcal{S}_R} p(\mathbf{z}_{R_iR_j}(\kappa)|\mathbf{x}_{R_i}(\kappa),\mathbf{x}_{R_j}(\kappa)) \prod_{\{m,n\}\in\mathcal{S}_T} p(\mathbf{z}_{R_mT_n}(\kappa)|\mathbf{x}_{R_m}(\kappa),\mathbf{x}_{T_n}(\kappa))\right]$$

where $\mathbf{z}(0:k)$ denotes all the sensor measurements available in the time interval $[0,k]$, $\mathcal{S}_R := \{i,j|i=1,\ldots,M; j=1,\ldots,M; i\neq j\}$, and $\mathcal{S}_T := \{m,n|m=1,\ldots,M; n=1,\ldots,N\}$. In the above expression (20), we have used the assumptions that state and measurement noise is independent and both robot and target motion is a Markov process [see (5) and (12)]. By assuming Gaussian noise, maximization of (20) is equivalent to minimizing the following cost function:

$$\begin{aligned} c\left(\mathbf{x}(0:k)\right) :=& \frac{1}{2}||\mathbf{x}_0-\hat{\mathbf{x}}_{0|0}||^2_{\mathbf{P}_{0|0}} + \\ & \frac{1}{2}\sum_{\kappa=0}^{k-1}\sum_{i=1}^{M}||\mathbf{x}_{R_i}(\kappa+1)-\mathbf{f}(\mathbf{x}_{R_i}(\kappa),\mathbf{u}_{m_i}(\kappa))||^2_{\bar{\mathbf{Q}}_{R_i}(\kappa)} + \\ & \frac{1}{2}\sum_{\kappa=0}^{k-1}\sum_{j=1}^{N}||\mathbf{x}_{T_j}(\kappa+1)-\mathbf{\Phi}_{T_j}\mathbf{x}_{T_j}(\kappa)||^2_{\bar{\mathbf{Q}}_{T_j}(\kappa)} + \\ & \frac{1}{2}\sum_{\kappa=0}^{k}\sum_{\{i,j\}\in\mathcal{S}_R}||\mathbf{z}_{R_iR_j}(\kappa)-\mathbf{h}(\mathbf{x}_{R_i}(\kappa),\mathbf{x}_{R_j}(\kappa))||^2_{\mathbf{R}_{R_{ij}}(\kappa)} + \\ & \frac{1}{2}\sum_{\kappa=0}^{k}\sum_{\{m,n\}\in\mathcal{S}_T}||\mathbf{z}_{R_mT_n}(\kappa)-\mathbf{h}(\mathbf{x}_{R_m}(\kappa),\mathbf{x}_{T_n}(\kappa))||^2_{\mathbf{R}_{T_{mn}}(\kappa)} \end{aligned} \tag{21}$$

where $\bar{\mathbf{Q}}_{R_i} := \mathbf{G}_{R_i}\mathbf{Q}_{R_i}\mathbf{G}^T_{R_i}$ [see (5), (8), and (7)], and $\bar{\mathbf{Q}}_{T_j} := \mathbf{G}_{T_j}\mathbf{Q}_{T_j}\mathbf{G}^T_{T_j}$ [see (12)]. In the above expressions, we have also employed the notation, $||\mathbf{r}||^2_{\mathbf{\Sigma}} := \mathbf{r}^T\mathbf{\Sigma}^{-1}\mathbf{r}$, i.e., the squared Mahalanobis distance of residual $\mathbf{r}$ with covariance $\mathbf{\Sigma}$.

This is a *nonlinear* least-squares problem [see (5), (14), and (18)]. A standard iterative Newton-Raphson approach [27] is often used for its optimization, although it is only able to converge to one local minimum within the basin of attraction of the initial estimate. Specifically, at the ℓ-th iteration, a correction, $\delta\mathbf{x}^{(\ell)}(0:k)$, to the current estimate, $\hat{\mathbf{x}}^{(\ell)}(0:k|k)$, is computed by minimizing the second-order Taylor-series approximation of (21):[3]

$$\begin{aligned} & c\left(\hat{\mathbf{x}}(0:k|k)+\delta\mathbf{x}(0:k)\right) \simeq \\ & c(\hat{\mathbf{x}}(0:k|k))+\mathbf{b}_b\delta\mathbf{x}(0:k)+\frac{1}{2}\delta\mathbf{x}^T(0:k)\mathbf{A}_b\delta\mathbf{x}(0:k) \end{aligned} \tag{22}$$

In the above equation, the Jacobian matrix of $c(\cdot)$ with respect to $\mathbf{x}(0:k)$, $\mathbf{b}_b^{(\ell)}$, is computed and factorized as follows:

$$\begin{aligned} \mathbf{b}_b = & \mathbf{P}_{0|0}^{-1}\left(\hat{\mathbf{x}}(0|k)-\hat{\mathbf{x}}(0|0)\right) + \\ & \sum_{\kappa=0}^{k-1}\sum_{i=1}^{M}\mathcal{F}_{R_i}^T(\kappa)\bar{\mathbf{Q}}_{R_i}^{-1}(\kappa)\left(\hat{\mathbf{x}}_{R_i}(\kappa+1|k)-\mathbf{f}(\hat{\mathbf{x}}_{R_i}(\kappa|k),\mathbf{u}_{m_i}(\kappa))\right)+ \\ & \sum_{\kappa=0}^{k-1}\sum_{i=1}^{N}\mathcal{F}_{T_i}^T(\kappa)\bar{\mathbf{Q}}_{T_i}^{-1}(\kappa)\left(\hat{\mathbf{x}}_{T_i}(\kappa+1|k)-\mathbf{\Phi}_{T_i}(\kappa)\hat{\mathbf{x}}_{T_i}(\kappa|k)\right)+ \\ & \sum_{\kappa=0}^{k}\sum_{\{i,j\}\in\mathcal{S}_R}\mathcal{H}_{R_{ij}}^{(\ell)^T}(\kappa)\mathbf{R}_{R_{ij}}^{-1}\left(\mathbf{z}_{R_iR_j}(\kappa)-\mathbf{h}(\hat{\mathbf{x}}_{R_i}(\kappa|k),\hat{\mathbf{x}}_{R_j}(\kappa|k))\right)+ \\ & \sum_{\kappa=0}^{k}\sum_{\{m,n\}\in\mathcal{S}_T}\mathcal{H}_{T_{mn}}^{(\ell)^T}(\kappa)\mathbf{R}_{T_{mn}}^{-1}\left(\mathbf{z}_{R_mT_n}(\kappa)-\mathbf{h}(\hat{\mathbf{x}}_{R_m}(\kappa|k),\hat{\mathbf{x}}_{T_n}(\kappa|k))\right) \\ =: & \mathbf{J}^T\mathbf{r} \end{aligned} \tag{23}$$

[3]Note that hereafter we occasionally drop the time and iteration indices in order to make the presentation concise, while these can be easily inferred from the context.

where

$$\mathbf{J} := \begin{bmatrix} \mathbf{P}_{0|0}^{-\frac{1}{2}} \\ \bar{\mathbf{Q}}_{R_1}^{-\frac{1}{2}}(0)\mathcal{F}_{R_1}(0) \\ \vdots \\ \bar{\mathbf{Q}}_{R_1}^{-\frac{1}{2}}(k-1)\mathcal{F}_{R_1}(k-1) \\ \vdots \\ \bar{\mathbf{Q}}_{R_M}^{-\frac{1}{2}}(0)\mathcal{F}_{R_M}(0) \\ \vdots \\ \bar{\mathbf{Q}}_{R_M}^{-\frac{1}{2}}(k-1)\mathcal{F}_{R_M}(k-1) \\ \vdots \\ \bar{\mathbf{Q}}_{T_1}^{-\frac{1}{2}}(0)\mathcal{F}_{T_1}(0) \\ \vdots \\ \bar{\mathbf{Q}}_{T_1}^{-\frac{1}{2}}(k-1)\mathcal{F}_{T_1}(k-1) \\ \vdots \\ \bar{\mathbf{Q}}_{T_N}^{-\frac{1}{2}}(0)\mathcal{F}_{T_N}(0) \\ \vdots \\ \bar{\mathbf{Q}}_{T_N}^{-\frac{1}{2}}(k-1)\mathcal{F}_{T_N}(k-1) \\ \vdots \\ \mathbf{R}_{R_{ij}}^{-\frac{1}{2}}(0)\mathcal{H}_{R_{ij}}(0) \\ \vdots \\ \mathbf{R}_{R_{ij}}^{-\frac{1}{2}}(k)\mathcal{H}_{R_{ij}}(k) \\ \vdots \\ \mathbf{R}_{T_{mn}}^{-\frac{1}{2}}(0)\mathcal{H}_{T_{mn}}(0) \\ \vdots \\ \mathbf{R}_{T_{mn}}^{-\frac{1}{2}}(k)\mathcal{H}_{T_{mn}}(k) \\ \vdots \end{bmatrix} = \mathbf{\Lambda} \begin{bmatrix} \mathbf{I} \\ \mathcal{F}_{R_1}(0) \\ \vdots \\ \mathcal{F}_{R_1}(k-1) \\ \vdots \\ \mathcal{F}_{R_M}(0) \\ \vdots \\ \mathcal{F}_{R_M}(k-1) \\ \vdots \\ \mathcal{F}_{T_1}(0) \\ \vdots \\ \mathcal{F}_{T_1}(k-1) \\ \vdots \\ \mathcal{F}_{T_N}(0) \\ \vdots \\ \mathcal{F}_{T_N}(k-1) \\ \vdots \\ \mathcal{H}_{R_{ij}}(0) \\ \vdots \\ \mathcal{H}_{R_{ij}}(k) \\ \vdots \\ \mathcal{H}_{T_{mn}}(0) \\ \vdots \\ \mathcal{H}_{T_{mn}}(k) \\ \vdots \end{bmatrix} \tag{24}$$

$$\mathbf{r} := \begin{bmatrix} \mathbf{P}_{0|0}^{-\frac{1}{2}} \left(\hat{\mathbf{x}}(0|k) - \hat{\mathbf{x}}(0|0)\right) \\ \bar{\mathbf{Q}}_{R_1}^{-\frac{1}{2}}(0) \left(\hat{\mathbf{x}}_{R_1}(1|k) - \mathbf{f}(\hat{\mathbf{x}}_{R_1}(0|k), \mathbf{u}_{m_1}(0))\right) \\ \vdots \\ \bar{\mathbf{Q}}_{R_1}^{-\frac{1}{2}}(k-1) \left(\hat{\mathbf{x}}_{R_1}(k|k) - \mathbf{f}(\hat{\mathbf{x}}_{R_1}(k-1|k), \mathbf{u}_{m_1}(k-1))\right) \\ \vdots \\ \bar{\mathbf{Q}}_{R_M}^{-\frac{1}{2}}(0) \left(\hat{\mathbf{x}}_{R_M}(1|k) - \mathbf{f}(\hat{\mathbf{x}}_{R_M}(0|k), \mathbf{u}_{m_M}(0))\right) \\ \vdots \\ \bar{\mathbf{Q}}_{R_M}^{-\frac{1}{2}}(k-1) \left(\hat{\mathbf{x}}_{R_M}(k|k) - \mathbf{f}(\hat{\mathbf{x}}_{R_M}(k-1|k), \mathbf{u}_{m_M}(k-1))\right) \\ \vdots \\ \bar{\mathbf{Q}}_{T_1}^{-\frac{1}{2}}(0) \left(\hat{\mathbf{x}}_{T_1}(1|k) - \mathbf{\Phi}_{T_1}(0)\hat{\mathbf{x}}_{T_1}(0|k)\right) \\ \vdots \\ \bar{\mathbf{Q}}_{T_1}^{-\frac{1}{2}}(k-1) \left(\hat{\mathbf{x}}_{T_1}(k|k) - \mathbf{\Phi}_{T_1}(k-1)\hat{\mathbf{x}}_{T_1}(k-1|k)\right) \\ \vdots \\ \bar{\mathbf{Q}}_{T_N}^{-\frac{1}{2}}(0) \left(\hat{\mathbf{x}}_{T_N}(1|k) - \mathbf{\Phi}_{T_N}(0)\hat{\mathbf{x}}_{T_N}(0|k)\right) \\ \vdots \\ \bar{\mathbf{Q}}_{T_N}^{-\frac{1}{2}}(k-1) \left(\hat{\mathbf{x}}_{T_N}(k|k) - \mathbf{\Phi}_{T_N}(k-1)\hat{\mathbf{x}}_{T_N}(k-1|k)\right) \\ \vdots \\ \mathbf{R}_{R_{ij}}^{-\frac{1}{2}}(0) \left(\mathbf{z}_{R_iR_j}(0) - \mathbf{h}(\hat{\mathbf{x}}_{R_i}(0|k), \hat{\mathbf{x}}_{R_j}(0|k))\right) \\ \vdots \\ \mathbf{R}_{R_{ij}}^{-\frac{1}{2}}(k) \left(\mathbf{z}_{R_iR_j}(k) - \mathbf{h}(\hat{\mathbf{x}}_{R_i}(k|k), \hat{\mathbf{x}}_{R_j}(k|k))\right) \\ \vdots \\ \mathbf{R}_{T_{mn}}^{-\frac{1}{2}}(0) \left(\mathbf{z}_{R_mT_n}(0) - \mathbf{h}(\hat{\mathbf{x}}_{R_m}(0|k), \hat{\mathbf{x}}_{T_n}(0|k))\right) \\ \vdots \\ \mathbf{R}_{T_{mn}}^{-\frac{1}{2}}(k) \left(\mathbf{z}_{R_mT_n}(k) - \mathbf{h}(\hat{\mathbf{x}}_{R_m}(k|k), \hat{\mathbf{x}}_{T_n}(k|k))\right) \\ \vdots \end{bmatrix} \tag{25}$$

for all $\{i, j\} \in \mathcal{S}_R$ and $\{m, n\} \in \mathcal{S}_T$. In the above expression (24),

$\mathbf{\Lambda} := \mathbf{Diag}\left(\mathbf{P}_{0|0}^{-\frac{1}{2}}, \cdots, \bar{\mathbf{Q}}_{R_i}^{-\frac{1}{2}}(\kappa), \cdots, \bar{\mathbf{Q}}_{T_i}^{-\frac{1}{2}}(\kappa), \cdots, \mathbf{R}_{R_{ij}}^{-\frac{1}{2}}(\kappa), \cdots, \mathbf{R}_{T_{mn}}^{-\frac{1}{2}}(\kappa), \cdots\right)$.

On the other hand, during the Gauss-Newton iterations which are arguably the most widely used in practice, the Hessian, $\mathbf{A}_b$, is approximated by:

$$\begin{aligned}\mathbf{A}_b &= \mathbf{P}_{0|0}^{-1} + \sum_{\kappa=0}^{k-1}\sum_{i=1}^{M} \mathcal{F}_{R_i}^T(\kappa)\bar{\mathbf{Q}}_{R_i}^{-1}(\kappa)\mathcal{F}_{R_i}(\kappa) + \sum_{\kappa=0}^{k-1}\sum_{i=1}^{N} \mathcal{F}_{T_i}^T(\kappa)\bar{\mathbf{Q}}_{T_i}^{-1}(\kappa)\mathcal{F}_{T_i}(\kappa) \\ &+ \sum_{\kappa=0}^{k}\sum_{\{i,j\}\in\mathcal{S}_R} \mathcal{H}_{R_{ij}}^T(\kappa)\mathbf{R}_{R_{ij}}^{-1}(\kappa)\mathcal{H}_{R_{ij}}(\kappa) + \sum_{\kappa=0}^{k}\sum_{\{m,n\}\in\mathcal{S}_T} \mathcal{H}_{T_{mn}}^T(\kappa)\mathbf{R}_{T_{mn}}^{-1}(\kappa)\mathcal{H}_{T_{mn}}(\kappa) \\ &=: \mathbf{J}^T\mathbf{J} \end{aligned} \tag{26}$$

This is a good approximation for small-residual problems [24]. Due to the sparse structure of the matrices, $\mathcal{F}_{R_i}$, $\mathcal{F}_{T_i}$, $\mathcal{H}_{R_{ij}}$, and $\mathcal{H}_{T_{mn}}$ [see (6), (13), (15), and (19)], the matrix $\mathbf{A}_b$ is also sparse, which can be exploited to speed-up the solution of the linearized system, $\mathbf{A}_b\delta\mathbf{x} = \mathbf{b}_b$.

Based on (22), (23) and (26), it is not difficult to see that:

$$\min_{\delta\mathbf{x}(0:k)} \; c(\hat{\mathbf{x}}(0:k|k) + \delta\mathbf{x}(0:k)) \;\Leftrightarrow\; \min_{\delta\mathbf{x}(0:k)} \; ||\mathbf{J}\delta\mathbf{x}(0:k) - \mathbf{r}||^2 \tag{27}$$

The QR factorization is often employed to solve the above least-squares problem (27), i.e.,

$$\begin{aligned}\min_{\delta\mathbf{x}(0:k)} \quad & ||\mathbf{J}\delta\mathbf{x}(0:k) - \mathbf{r}||^2 = \left|\left|\mathbf{Q}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}\delta\mathbf{x}(0:k) - \mathbf{r}\right|\right|^2 = \\ & \left|\left|\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}\delta\mathbf{x}(0:k) - \mathbf{Q}^T\mathbf{r}\right|\right|^2 =: \left|\left|\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix}\delta\mathbf{x}(0:k) - \begin{bmatrix}\mathbf{d}\\ \mathbf{e}\end{bmatrix}\right|\right|^2 \qquad (28) \\ \Leftrightarrow \min_{\delta\mathbf{x}(0:k)} \quad & ||\mathbf{R}\delta\mathbf{x}(0:k) - \mathbf{d}||^2 \qquad (29)\end{aligned}$$

where we have used the reduced QR of $\mathbf{J}$ [28], i.e.,

$$\mathbf{J} = \mathbf{Q}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{R}\\ \mathbf{0}\end{bmatrix} = \mathbf{Q}_1\mathbf{R} \tag{30}$$

From (29) $\delta\mathbf{x}(0:k)$ can be efficiently solved by back substitution. Once $\delta\mathbf{x}^{(\ell)}(0:k)$ is found, the new estimate is updated as:

$$\hat{\mathbf{x}}^{(\ell+1)}(0:k|k) = \hat{\mathbf{x}}^{(\ell)}(0:k|k) + \delta\mathbf{x}^{(\ell)}(0:k) \tag{31}$$

Only given an initial estimate $\hat{\mathbf{x}}^{(0)}(0:k|k)$ that resides within the attraction basin of the global optimum, this iterative algorithm computes the global minimum (i.e., true MAP estimate) for the entire state trajectory.

3.2. iSAM algorithm

Every time when a new measurement such as $\mathbf{z}_{R_mT_n}(k+1)$ becomes available,[4] in principle, we need to *recompute* the full Jacobian $\mathbf{J}$ (24) and residual $\mathbf{r}$ (25), and then solve the batch-MAP problem from scratch. However, this is an expensive operator, even by exploiting the sparse structure of the Hessian matrix [17, 20, 29], in part because the size of the state trajectory grows unbounded over time and the batch problem can quickly become too large for real-time operation.

In order to save computations, incremental smoothing is often used. For example, the state-of-the-art iSAM algorithm reuses the previously-computed Jacobian and *incrementally* updates the QR factorization in solving the batch-MAP problem [19]. Specifically, we augment $\mathbf{J}$ (without recomputing it) with the new measurement Jacobian $\mathcal{H}_{T_{mn}}$ [see (30)]:

$$\mathbf{J}_a := \begin{bmatrix} \mathbf{J} \\ \mathbf{R}_{T_{mn}}^{-\frac{1}{2}} \mathcal{H}_{T_{mn}} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{R}_{T_{mn}}^{-\frac{1}{2}} \mathcal{H}_{T_{mn}} \end{bmatrix} \tag{32}$$

We now aim to decompose $\mathbf{J}_a$ into a triangular form (i.e., square-root information matrix). Since $\mathbf{J}$ was already factorized into the triangular $\mathbf{R}$, we only need to zero out the new block row, i.e., the new measurement Jacobian $\mathbf{H}_{T_{mn}}$, in order to obtain the updated square-root information matrix $\mathbf{R}_a$. This can be achieved efficiently, for example, by using Givens QR [28]. Similarly, the corresponding new residual vector can be obtained by applying the same Givens rotations to the augmented residual vector, $\mathbf{d}_a := \begin{bmatrix} \mathbf{d} \\ \mathbf{R}_{T_{mn}}^{-\frac{1}{2}} \left(\mathbf{z}_{R_mT_n} - \mathbf{h}(\hat{\mathbf{x}}_{R_m}, \hat{\mathbf{x}}_{T_n}) \right) \end{bmatrix}$. Similar steps are taken to process any other newly-available measurement, such as robot odometry $\mathbf{u}_{m_i}(k+1)$ and robot-to-robot observations $\mathbf{z}_{R_iR_j}(k+1)$.

It is important to note that, although relinearization is not needed at each time step when a new measurement becomes available, in order to reduce the linearization errors, we relinearize the system at the latest, and thus the best, state estimates periodically [19] or as needed when the linearization point significantly deviates from the current state estimate [21]. In addition, we can combine variable reordering [19] with this batch factorization to reduce

[4]Note that the robot and target motion [see (5) and (12)] can be considered as a different type of measurements and hence can be treated analogously.

fill-in of the resulting triangular system [see (28)], which can further speed up the subsequent incremental estimation.

4. Observability Analysis

In this section, we examine the parameter observability properties [25] of the batch-MAP based CLATT system, which, for the time being, is considered as a parameter (instead of state) estimation problem. It should be noted that even though a rich body of literature on observability analysis for SLAM or CL (e.g., see [30–36]), to the best of our knowledge, no prior work has examined the *parameter observability* for the CLATT system. The study of parameter observability examines whether the information provided by the available measurements is sufficient for estimating the parameters without ambiguity; when parameter observability holds, the Fisher information matrix (FIM) is invertible [25]. Since the FIM encapsulates all the information available in the measurements, by studying its nullspace (i.e., "unobservable subspace") we can gain insights about the directions in the parameter (state) space along which the estimator should or should not acquire information from the measurements.

In particular, close inspection of the Hessian matrix (FIM), $\mathbf{A}_b$ (26), reveals the following observability properties of the batch-MAP based CLATT, which is the benchmark system for our proposed incremental solution:

Lemma 4.1. *Without prior information,*[5] *at time-step k the FIM (Hessian) of the batch-MAP based CLATT [see* (26)*] has a nullspace of dimension three,*

[5]Since we are interested in the information contained in the available measurements, we consider the case without prior (i.e., $\mathbf{P}_{0|0} \to \boldsymbol{\infty}$). In this case, the first block row of the matrix $\mathbf{J}$ (24) corresponding to the prior becomes zeros and can be ignored without changing the rank of the matrix.

which is given by:

$$\mathbf{N}_b(k) := \operatorname{null}\left(\mathbf{A}_b(k)\right) = \underset{\text{col.}}{\operatorname{span}} \begin{bmatrix} \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_1}(0|k) \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_1}(k|k) \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_M}(0|k) \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_M}(k|k) \\ \mathbf{0} & 1 \\ & \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_1}(0|k) \\ \mathbf{0} & (\mathbf{I}_{\dim(\mathbf{d}_{T_1})/2} \otimes \boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_1}(0|k) \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_1}(k|k) \\ \mathbf{0} & (\mathbf{I}_{\dim(\mathbf{d}_{T_1})/2} \otimes \boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_1}(k|k) \\ \vdots & \vdots \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_N}(0|k) \\ \mathbf{0} & (\mathbf{I}_{\dim(\mathbf{d}_{T_N})/2} \otimes \boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_N}(0|k) \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_N}(k|k) \\ \mathbf{0} & (\mathbf{I}_{\dim(\mathbf{d}_{T_N})/2} \otimes \boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_N}(k|k) \end{bmatrix} \tag{33}$$

where null *denotes the right nullspace and* $\otimes$ *is the Kronecker product.*

Proof. See Appendix A. □

This result implies that any small change of the state along the directions spanned by the columns of $\mathbf{N}_b(k)$ cannot be discerned based on the available measurements. Thus, the subspace spanned by these directions is termed the "unobservable" subspace. In particular, the physical interpretation of $\mathbf{N}_b(k)$

can be found analogously as in [18], i.e., the first two columns describe global translations of the state, while the third one describes global rotations and higher-order kinematic states such as velocity and acceleration. We thus see that without a prior, the global pose (position and orientation) as well as the higher-order kinematic states cannot be determined. On the other hand, when the prior is available (i.e., $\mathbf{P}(0|0) < \boldsymbol{\infty}$), the Hessian generally becomes full-rank [see (24) and (26)], and thus it is possible to uniquely determine estimates for all the state variables. These results agree with our intuition that based on *relative* measurements alone, the state trajectory cannot be determined with respect to the *global* frame of reference.

When using an incremental smoothing solution to approximate the optimal batch one, the smoother is expected to have the similar observability properties as the batch-MAP estimator. However, this is not the case for the standard iSAM algorithm. In particular, as seen from (32) when augmenting the Jacobian $\mathbf{J}_a$ at time-step $k+1$, we do not recompute this Jacobian from scratch and instead reuse the previous computations of $\mathbf{J}$ that was calculated using the past state estimates $\hat{\mathbf{x}}(\ell|\ell)$, $\forall \ell = 0, \cdots, k$; while the current new Jacobians – including propagation Jacobian $\mathcal{F}_{R_i}$ and measurement Jacobians $\mathcal{H}_{T_{mm}}$ and $\mathcal{H}_{R_{ij}}$ – are evaluated at the current state estimate $\hat{\mathbf{x}}(\ell|k+1)$. This leads to different estimates for the same state variable used as linearization points in computing incremental Jacobians, which results in different observability properties for the standard incremental smoother than those of the batch-MAP estimator (which is analogous to the case of fixed-lag smoother [18, 37]).

5. Unscented Incremental Smoothing

In this section, we present in detail the proposed UIS algorithm that employs the unscented transform to compute the Jacobians and thus the Hessian (FIM) while enforcing correct dimensions for the corresponding FIM's nullspace. In the following we begin with a brief overview of unscented transform in the context of linear regression Kalman filter (LRKF) [16] and UKF [15], which will serve as the basis for the proposed approach.

5.1. Unscented transform

The LRKF [16] formulates a linear regression problem to approximate a nonlinear function $\mathbf{z} = \mathbf{h}(\mathbf{x})$ by a linear model $\mathbf{z} \simeq \mathbf{H}\mathbf{x} + \mathbf{v}$, where $\mathbf{H}$ and $\mathbf{v}$

are respectively the regression matrix and vector, and $\mathbf{e} := \mathbf{z} - (\mathbf{H}\mathbf{x} + \mathbf{v})$ is the linearization error, i.e.,

$$\min_{\mathbf{H},\mathbf{v}} \quad \int_{-\infty}^{+\infty} ||\mathbf{z} - (\mathbf{H}\mathbf{x} + \mathbf{v})||^2 p(\mathbf{x}) d\mathbf{x} \tag{34}$$

In general, it is intractable to find a closed-form solution to this problem due to the nonlinearity of $\mathbf{z} = \mathbf{h}(\mathbf{x})$. To go around it, the LRKF *randomly* selects $r+1$ weighted sample points, $\{\mathcal{X}_i, w_i\}_{i=0}^{r}$, so that their sample mean and covariance are equal to the true mean and covariance of $\mathbf{x}$:

$$\bar{\mathbf{x}} = \sum_{i=0}^{r} w_i \mathcal{X}_i \ = \mathbb{E}(\mathbf{x}) \tag{35}$$

$$\bar{\mathbf{P}}_{\mathbf{xx}} = \sum_{i=0}^{r} w_i \left(\mathcal{X}_i - \bar{\mathbf{x}}\right)\left(\mathcal{X}_i - \bar{\mathbf{x}}\right)^T = \mathbb{E}\left[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T\right] \tag{36}$$

where $\mathbb{E}(\cdot)$ is the expectation operator. Using the standard sample-based approximation $p(\mathbf{x}) \simeq \sum_{i=0}^{r} w_i \delta(\mathbf{x} - \mathcal{X}_i)$, where $\delta(\cdot)$ is the Dirac delta function, the linear regression problem (34) becomes:

$$\min_{\mathbf{H},\mathbf{v}} \quad \sum_{i=0}^{r} w_i ||\mathcal{Z}_i - (\mathbf{H}\mathcal{X}_i + \mathbf{v})||^2 \tag{37}$$

where $\mathcal{Z}_i := \mathbf{h}(\mathcal{X}_i)$, whose linearization error is denoted by $\mathbf{e}_i := \mathcal{Z}_i - (\mathbf{H}\mathcal{X}_i + \mathbf{v})$. The optimal solutions of (37) for $\mathbf{H}$ and $\mathbf{v}$ are given by [16]:

$$\mathbf{H} = \bar{\mathbf{P}}_{\mathbf{zx}} \bar{\mathbf{P}}_{\mathbf{xx}}^{-1} \ , \qquad \mathbf{v} = \bar{\mathbf{z}} - \mathbf{H}\bar{\mathbf{x}} \tag{38}$$

where

$$\bar{\mathbf{z}} = \sum_{i=0}^{r} w_i \mathcal{Z}_i \tag{39}$$

$$\bar{\mathbf{P}}_{\mathbf{zx}} = \sum_{i=0}^{r} w_i \left(\mathcal{Z}_i - \bar{\mathbf{z}}\right)\left(\mathcal{X}_i - \bar{\mathbf{x}}\right)^T \tag{40}$$

$$\bar{\mathbf{P}}_{\mathbf{zz}} = \sum_{i=0}^{r} w_i \left(\mathcal{Z}_i - \bar{\mathbf{z}}\right)\left(\mathcal{Z}_i - \bar{\mathbf{z}}\right)^T \tag{41}$$

With (38), we obtain the linear approximation for the nonlinear function $\mathbf{z} = \mathbf{h}(\mathbf{x})$, *without* performing Taylor series expansion. Rather than sampling *randomly* as in the LRKF presented above, the unscented transform (or statistical linearization) employed by the UKF [15] *deterministically* selects $2n + 1$ so-called sigma points $\mathcal{X}_i$, along with their weights w_i, $i = 1, \ldots, n$, according to the following equations:

$$\mathcal{X}_0 = \bar{\mathbf{x}} \ , \ w_0 = \frac{2\kappa}{2(n+\lambda)} \tag{42}$$

$$\mathcal{X}_i = \bar{\mathbf{x}} + \left[\sqrt{(n+\lambda)\bar{\mathbf{P}}_{\mathbf{xx}}}\right]_i \ , \ w_i = \frac{1}{2(n+\lambda)} \tag{43}$$

$$\mathcal{X}_{i+n} = \bar{\mathbf{x}} - \left[\sqrt{(n+\lambda)\bar{\mathbf{P}}_{\mathbf{xx}}}\right]_i \ , \ w_{i+n} = \frac{1}{2(n+\lambda)} \tag{44}$$

where n is the dimension of $\bar{\mathbf{x}}$ (i.e., $n = \dim(\bar{\mathbf{x}})$), $\left[\sqrt{(n+\lambda)\bar{\mathbf{P}}_{\mathbf{xx}}}\right]_i$ is the i-th column of the matrix $\sqrt{(n+\lambda)\bar{\mathbf{P}}_{\mathbf{xx}}}$, and λ is a design parameter in the selection of the sigma points, usually chosen so that $n + \lambda = 3$. This set of sigma points captures the moments of the underlying distribution up to the third-order for the Gaussian case [15]. Once the sigma points are generated, the unscented transform (UKF) computes the regression matrix $\mathbf{H}$ as in (38) which is often considered as the inferred, unscented, Jacobian, in analogy to the Jacobian in the EKF.

5.2. The proposed algorithm

The unscented transform (statistical linearization) described above provides a numerical, instead of analytical, way to compute Jacobians and achieve reduced linearization errors for many nonlinear estimation problems (e.g., [38]). We expect the similar gain when using it to compute the inferred unscented Jacobians, and thus the Hessian, in the unscented incremental smoothing. Moreover, we also expect that the resulting inferred unscented Hessian has similar observability properties as that of the batch-MAP estimator, in particular, a nullspace of dimension three (see Section 4). However, this in general is not the case. When numerically computing the dimension of the nullspace of the unscented Hessian, we find that it is three only for the first few time steps, and decreases quickly as more measurements become available. This indicates that the *naive* UIS (i.e., directly applying the standard unscented transform to compute the Jacobians and thus Hessians,

without imposing any constraint to ensure appropriate observability as in the proposed UIS presented below) acquires spurious information from the available measurements, in the directions of the state space where no information is actually available, which may degrade the performance (see Section 6).

To address the aforementioned issue, we explicitly enforce the appropriate observability constraints on the unscented transform (linear regression) (37) such that the resulting *unscented* Hessian possesses the desired nullspace, i.e.,

$$\min_{\mathbf{H},\mathbf{v}} \quad \sum_{i=0}^{r} w_i ||\mathcal{Z}_i - (\mathbf{H}\mathcal{X}_i + \mathbf{v})||^2 \tag{45}$$

$$\text{subject to} \quad \mathbf{AN} = \mathbf{0} \tag{46}$$

where $\mathbf{N}$ is a design choice that defines the desired nullspace of dimension three for the Hessian matrix $\mathbf{A}$ of the proposed UIS [see (26)]. Although, ideally we would like to have such a nullspace to be the one for the batch-MAP estimator (33), this is *not* possible during incremental estimation, because the *noncausal, smoothed* state estimates (i.e., $\hat{\mathbf{x}}(\ell|k)$ where $\ell < k$, that is, using the information from future measurements) are used in batch estimation (33), while these estimates are not available yet before executing the incremental estimator. Therefore, we instead choose the nullspace in the same form as that of the batch-MAP estimator (33), while evaluating it at the *causal,*

propagated state estimates, $\hat{\mathbf{x}}(k|k-1)$, for all the times, i.e.,

$$\mathbf{N}(k) = \underset{\text{col.}}{\text{span}} \begin{bmatrix} \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_1}(0|0) \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_1}(k|k-1) \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_M}(0|0) \\ \mathbf{0} & 1 \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_M}(k|k-1) \\ \mathbf{0} & 1 \\ & \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_1}(0|0) \\ \mathbf{0} & (\mathbf{I}_{\dim(\mathbf{d}_{T_1})/2} \otimes \boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_1}(0|0) \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_1}(k|k-1) \\ \mathbf{0} & (\mathbf{I}_{\dim(\mathbf{d}_{T_1})/2} \otimes \boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_1}(k|k-1) \\ \vdots & \vdots \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_N}(0|0) \\ \mathbf{0} & (\mathbf{I}_{\dim(\mathbf{d}_{T_N})/2} \otimes \boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_N}(0|0) \\ \vdots & \vdots \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_N}(k|k-1) \\ \mathbf{0} & (\mathbf{I}_{\dim(\mathbf{d}_{T_N})/2} \otimes \boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_N}(k|k-1) \end{bmatrix} \tag{47}$$

Once the nullspace, $\mathbf{N}(k)$, is determined, we are now to find the desired Jacobians and Hessian matrices. For simplicity of notations, we denote the propagation and measurement Jacobians [i.e., each block row in (24)] *generically* by $\mathbf{H}$. Using (24) and (26), it is not difficult to show that

$$\mathbf{HN} = \mathbf{0} \;\Rightarrow\; \mathbf{JN} = \mathbf{0} \;\Rightarrow\; \mathbf{J}^T\mathbf{JN} = \mathbf{AN} = \mathbf{0} \tag{48}$$

This implies that in order to ensure the Hessian $\mathbf{A}$ to have $\mathbf{N}$ as its desired nullspace, we only need to ensure the Jacobians to have the same nullspace.

Now we solve the constrained linear regression (unscented transform) (45)-(46) by seeking the desired Jacobian $\mathbf{H}(k)$ as the one closest to the most accurate Jacobian $\mathbf{H}_o(k)$, which is computed by the *canonical, unconstrained* unscented transform [see (42)-(44), (37), and (38)], while satisfying the observability constraint [see (48)]. This can be formulated as the following constrained minimization problem:

$$\min_{\mathbf{H}(k)} \ ||\mathbf{H}_o(k) - \mathbf{H}(k)||_F^2 \tag{49}$$

$$\text{subject to} \ \ \mathbf{H}(k)\mathbf{N}(k) = \mathbf{0} \tag{50}$$

where $||\cdot||_F$ denotes the Frobenius norm.

It is important to point out that in order to save computations, we should exploit the sparse structure of the propagation and measurement Jacobians [see (6), (15) and (19)]. Specifically, only *nonzero* entries of these matrices are considered as the optimization variables in the problem (49) and (50), and thus only the corresponding components of the nullspace $\mathbf{N}(k)$ are accordingly used in (50). To keep the presentation concise, in the above problem formulation we assume $\mathbf{H}$, $\mathbf{H}_o$ and $\mathbf{N}$ already consist only of the nonzero submatrices of the corresponding ones. The optimal, *closed-form* solution to the problem (49) and (50) is obtained based on the following lemma:

Lemma 5.1. *The optimal solution to the constrained minimization problem* (49) *and* (50) *is given by:*

$$\mathbf{H}(k) = \mathbf{H}_o(k)\left(\mathbf{I}_n - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\right) \tag{51}$$

Proof. See Appendix B. □

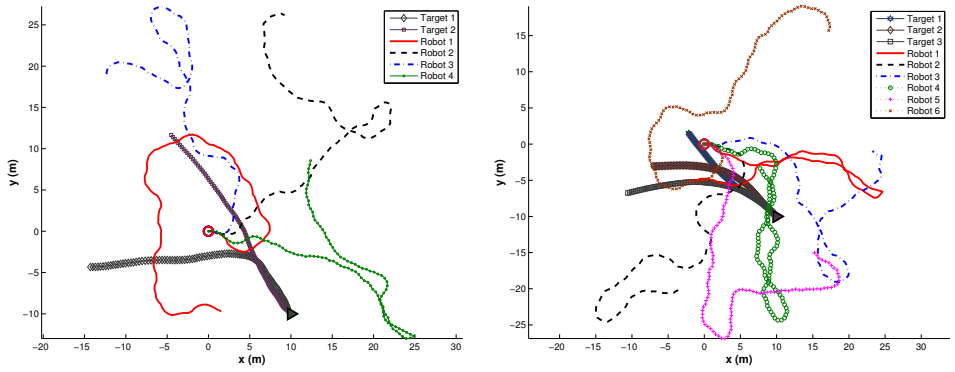
It is interesting to note that since $\mathbf{N}$ in (51) is the unobservable subspace (nullspace) at time-step k, $\left(\mathbf{I}_n - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\right)$ is the subspace orthogonal to $\mathbf{N}$, i.e., the observable subspace at time-step k. Hence, as seen from (51), $\mathbf{H}(k)$ is the *projection* of the canonical unscented Jacobian onto the observable subspace so that no spurious information is acquired from the measurements. Note also that once $\mathbf{H}(k)$ is found, it is easy to recover the full desired Jacobians by padding zeros to match the correct dimensions [see (6), (15) and (19)]. The main steps of the proposed UIS are outlined in Algorithm 1.

Algorithm 1 Unscented Incremental Smoothing (UIS) Algorithm

At each incremental step
Construct the desired nullspace $\mathbf{N}$ (47).
Compute the canonical measurement Jacobian $\mathbf{H}_o$ through the standard unscented transform [see (42)-(44), (37), and (38)].
Project $\mathbf{H}_o$ onto the observable subspace to obtain $\mathbf{H}$ (51).
Augment the Jacobian $\mathbf{J}_a$ (32) with the new unscented measurement Jacobian $\mathbf{H}$ padded with zeros to match dimensions [see (6), (15) and (19)].
Perform QR for $\mathbf{J}_a$ using Givens rotations.
Solve for $\delta\mathbf{x}$ by QR factorization (28).
Update the state estimate (31).
At periodic batch step
Relinearize all the available measurements and compute the whole Jacobian matrix $\mathbf{J}$ (24) and residual vector $\mathbf{r}$ (25).
Solve for $\delta\mathbf{x}$ by QR factorization (28).
Update the state estimate (31).

6. Simulation Results

In order to validate the proposed UIS algorithm for the CLATT problem, a series of Monte Carlo simulations were conducted under various conditions, such as different types of measurements, different target kinematic models, and different numbers of robots and targets. In the simulation tests, we consider the scenarios in which a team of mobile robots randomly move in an area of size 50 m $\times$ 50 m and track multiple moving targets. For example, Fig. 1 shows the typical trajectories of robots and targets, which are obtained from one realization of the Monte Carlo trials. It is important to point out that in this work we concentrate solely on *estimation* and do not intend to address (optimal) motion planning for the tracking robots, although the estimation (tracking and localization) performance can be improved if adopting an optimal motion strategy [39]. Thus, in the following tests, we focus on such adverse scenarios with random robot motion in order to show the localization and tracking capability of the proposed estimation algorithm, while in the ensuing experiment, we integrate the optimal motion of tracking robots [39] to further validate the proposed approach.



(a) (b)

Figure 1: A team of mobile robots move randomly inside a 50 m $\times$ 50 m arena and cooperatively track multiple targets whose motions follow (a) constant-velocity and (b) constant-acceleration kinematic models. Starting positions of the robots are marked by $\bigcirc$ and those of the targets by $\triangle$. Note that this is one realization of Monte Carlo simulations.

6.1. Range-only measurements and constant-velocity model

In the first test, *four* identical robots with a simple differential drive model start from the same location and randomly move on a planar surface at a constant linear velocity of $v = 0.5$ m/sec, while the rotational velocities are drawn from the uniform distribution over $[-0.5, 0.5]$ rad/sec. The two drive wheels are equipped with encoders, which measure their revolutions and provide measurements of velocity (i.e., right and left wheel velocities, v_r and v_l, respectively), with standard deviation equal to $\sigma = 2\%v$ for each wheel. These measurements are used to obtain the linear and rotational velocity measurements for the robots, which are given by:

$$v = \frac{v_r + v_l}{2} \ , \qquad \omega = \frac{v_r - v_l}{a} \tag{52}$$

where $a = 0.5$ m is the distance between the active wheels. Thus, the standard deviations of the linear and rotational velocity measurements are $\sigma_v = \frac{\sqrt{2}}{2}\sigma$ and $\sigma_\omega = \frac{\sqrt{2}}{a}\sigma$, respectively.

In this test, we adopt a constant-velocity or zero-acceleration kinematic model for the *two* targets [25]:

$$\dot{\mathbf{x}}_{T_i}(t) = \mathbf{F}_{T_i}\mathbf{x}_{T_i}(t) + \mathbf{G}_{T_i}\mathbf{w}_{T_i}(t) \tag{53}$$

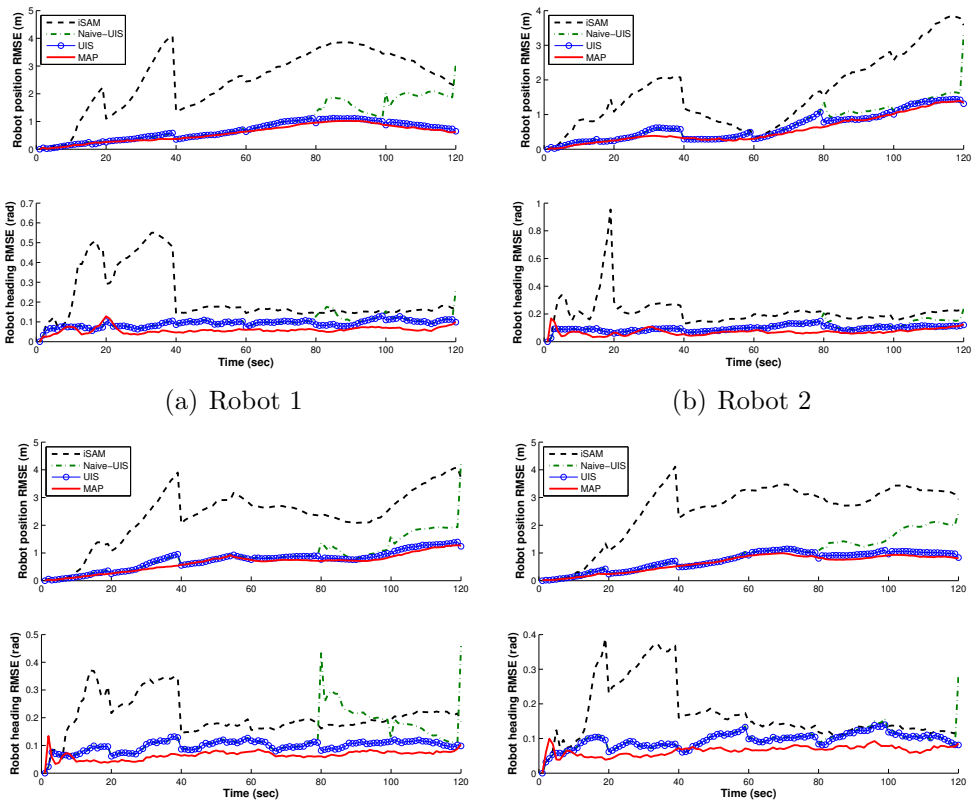
where

$$\mathbf{F}_{T_i} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{G}_{T_i} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad , \quad \mathbf{x}_{T_i}(t) = \begin{bmatrix} x_{T_i}(t) \\ y_{T_i}(t) \\ \dot{x}_{T_i}(t) \\ \dot{y}_{T_i}(t) \end{bmatrix} \tag{54}$$

and $\mathbf{w}_{T_i}(t) = \begin{bmatrix} w_x(t) & w_y(t) \end{bmatrix}^T$ is zero-mean, white Gaussian noise with covariance $\mathbb{E}\left[\mathbf{w}_{T_i}(t)\mathbf{w}_{T_i}(\tau)^T\right] = q\mathbf{I}_2\delta(t-\tau)$, $q = 0.01$, and $\delta(t-\tau)$ is the Dirac delta function. In our implementation, we discretize the continuous-time kinematic model (53) with time step $\delta t = 1$ sec. We stress that a known kinematic model such as the above constant-velocity model and the below constant-acceleration model is a common assumption for target tracking, while accurately modelling a target's motion is out of scope of this work. Nevertheless, by choosing appropriate driven white noise [see (53)], such stochastic models are able to describe a wide range of target motions in practice [25]. The initial true state of the target is $\mathbf{x}_{T_i}(0) = \begin{bmatrix} 10 & -10 & -0.1 & 0.1 \end{bmatrix}^T$, while the initial estimate of the target state is set to $\hat{\mathbf{x}}_{T_i}(0|0) \sim \mathcal{N}(\mathbf{x}_{T_i}(0), \mathbf{P}_{T_i}(0|0))$, with the initial covariance $\mathbf{P}_{T_i}(0|0) = \mathbf{Diag}(\mathbf{I}_2, 10^{-2}\mathbf{I}_2)$. Fig. 1(a) depicts one example of the trajectories of the robots and targets.

While any type of measurements is applicable for the proposed algorithm, we here first consider the range-only case where each robot records distance measurements to all the other robots and targets. Note that for simplicity we assume that each robot can observe all others at every time step. However, this is not a necessary assumption, as the analysis can easily be extended to the case where multiple propagation steps occur between measurement updates (e.g., limited sensing range, or different sampling frequencies between proprioceptive and exteroceptive sensors). In this test, data association is also assumed to be known in order to isolate its negative impact on the estimation performance and focus on the estimation algorithm itself. The standard deviation, σ_d, of the distance measurement noise is set to be 3% of the actual distance, which is relatively larger than what is typically encountered in practice. This is done purposefully, since larger noise leads to higher estimation errors and makes the comparison more apparent.

In this test, we perform 50 Monte Carlo simulations and compare four estimators: (i) the batch MAP, which is the best achievable in practice and serves as the benchmark, (ii) the standard iSAM [19], which computes the Jacobians and thus Hessians analytically, (iii) the *naive* UIS, which directly



(a) Robot 1 (b) Robot 2

(c) Robot 3 (d) Robot 4

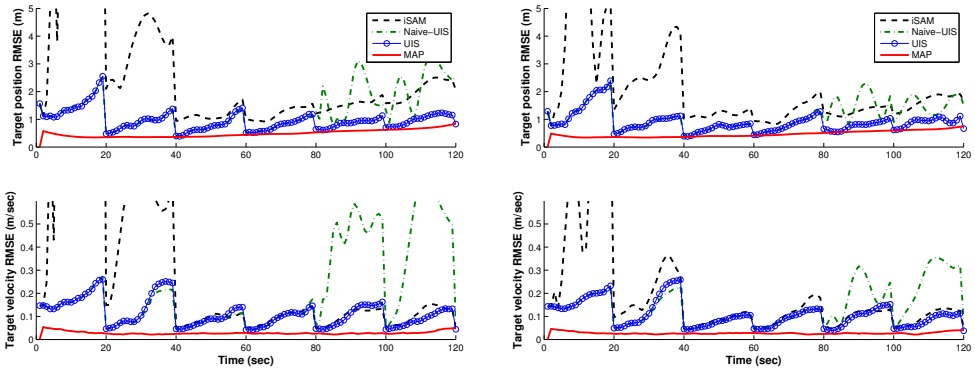
Figure 2: [Range-Only Constant-Velocity] Monte Carlo results for the four tracking robots' poses (positions and orientations). In these plots, the solid lines correspond to the batch MAP, the dashed lines to the standard iSAM, the dash-dotted lines to the naive UIS, and the solid lines with circles to the proposed UIS.

employs the standard unscented transform to compute the Jacobians, and (iv) the *proposed* UIS, which enforces the appropriate observability constraint in the unscented transform. Note that the three incremental smoothing algorithms (i.e., the iSAM, and the two UIS) relinearize all the available measurements and perform a batch update periodically every 20 time steps. This number can be selected in a more intelligent manner, for example, by taking into account the measurement noise, or by monitoring the quality of the current state estimates [21].

The comparison results of root mean squared errors (RMSE) [25] which quantify the estimation accuracy, are shown in Figs. 2 and 3 and Table 1. Specifically, Figs. 2 and 3 depict the RMSE over time for the four robots and two targets, respectively, while Table 1 averages these results over all the times and over all the simulations. As expected, the batch MAP estimator is optimal up to linearization errors and achieves the best accuracy. Since the naive UIS directly uses unscented transform to compute Jacobians numerically, which is expected to achieve reduced linearization errors as compared to the standard analytical way of computing Jacobians, it can perform better than the standard iSAM. However, as discussed before, the naive UIS does not ensure correct observability properties, which degrades its performance as shown in Figs. 2 and 3. Note that the periodic fluctuations in these figures are due to the periodic batch update carried out every 20 time steps by the three incremental smoothers. From these results, it becomes clear that the proposed UIS outperforms the other two incremental smoothing algorithms and performs closest (in particular, for the robot localization) to the benchmark, the batch MAP estimator. This is attributed to the fact that the proposed UIS not only employs the unscented transform to improve the accuracy of linearization approximation, but ensures the correct observability properties so that no spurious information is gained from the available measurements.

6.2. Bearing-only measurements and constant-acceleration model

To further validate the proposed UIS algorithm, we now consider the bearing-only CLATT case of significant nonlinearity, where the tracking robots measure relative bearing angles to the other robots and targets, with the standard deviation of measurement noise equal to $\sigma_\theta = 3$ deg. In this test, we deploy *six* robots with the same model as in the preceding simulations to track *three* targets whose motions however follow a constant-acceleration or zero-jerk kinematic model [25]. In this case, the continuous-time kinematic



(a) Target 1 (b) Target 2

Figure 3: [Range-Only Constant-Velocity] Monte Carlo results for the two targets' states (positions and velocities). In these plots, the solid lines correspond to the batch MAP, the dashed lines to the standard iSAM, the dash-dotted lines to the naive UIS, and the solid lines with circles to the proposed UIS.

Table 1: [Range-Only Constant-Velocity] Robot and Target Estimation Errors

	iSAM [19]	Naive-UIS	UIS	MAP
Robot Position RMSE (m)				
Robot 1:	2.4906	0.9349	0.6590	0.5782
Robot 2:	1.6401	0.7129	0.6444	0.5596
Robot 3:	2.3422	0.8935	0.7383	0.6504
Robot 4:	2.5375	0.9596	0.7270	0.6343
Robot Heading RMSE (rad)				
Robot 1:	0.2221	0.0987	0.0908	0.0590
Robot 2:	0.2141	0.1055	0.0958	0.0703
Robot 3:	0.2056	0.1304	0.0974	0.0652
Robot 4:	0.1702	0.0976	0.0945	0.0654
Target Position RMSE (m)				
Target 1:	3.2570	1.3278	0.9591	0.4833
Target 2:	2.1561	1.1266	0.8951	0.4623
Target Velocity RMSE (m/sec)				
Target 1:	0.4868	0.2223	0.1123	0.0285
Target 2:	0.2142	0.1465	0.1072	0.0274

model parameters for the i-th ($i = 1, 2, 3$) target become [see (53) and (54)]:

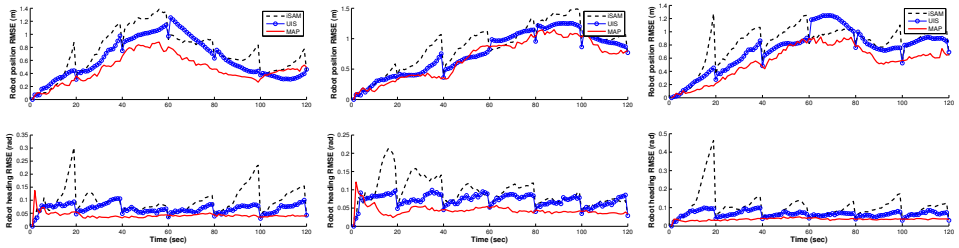
$$\mathbf{F}_{T_i} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad , \quad \mathbf{G}_{T_i} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad , \quad \mathbf{x}_{T_i}(t) = \begin{bmatrix} x_{T_i}(t) \\ y_{T_i}(t) \\ \dot{x}_{T_i}(t) \\ \dot{y}_{T_i}(t) \\ \ddot{x}_{T_i}(t) \\ \ddot{y}_{T_i}(t) \end{bmatrix} \tag{55}$$

The covariance of white jerk noise $\mathbf{w}_{T_i}(t)$ is set to $10^{-3}\mathbf{I}_2\delta(t-\tau)$. The initial acceleration is equal to 10^{-6} m/sec^2, while all the other parameters are identical to those used in the preceding test. Fig. 1(b) shows the trajectories of the robots and targets, which are obtained from one Monte Carlo trial.

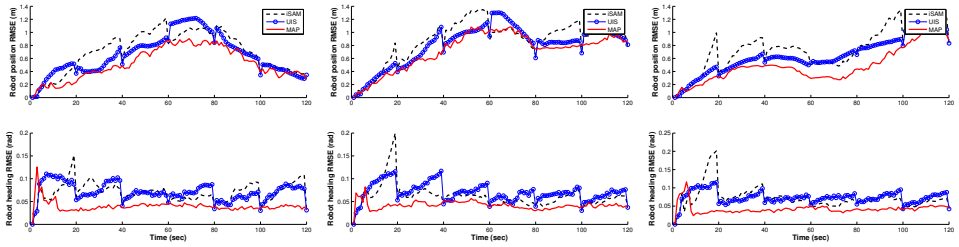
Similarly, we conduct 50 Monte Carlo simulations and compare: (i) the batch MAP, (ii) the iSAM [19], and (iii) the proposed UIS. Since the previous simulation results have already demonstrated the naive UIS (which can be considered as an intermediate step to our proposed UIS) does not work as well as the proposed algorithm, we hereafter omit the comparison to it in order to make presentation concise. The comparative results are shown in Figs. 4 and 5 and Table 2. As evident, these results are very similar to the ones in the previous simulations: Even with *different* type of sensor measurements, *different* target kinematic model, and *different* number of robots and targets, the proposed UIS still performs better than the standard iSAM, which is attributed to its reduced linearization errors due to unscented transform and correct observability properties.

7. Experimental Results

In this section, we present a real-world experiment conducted in a controlled indoor environment to further test the proposed UIS algorithm for a CLATT scenario, where the tracking robots follows an optimal motion strategy, rather than randomly as in the preceding simulations. During this test, three Pioneer-III robots – one acting as the moving target and the others serving as the tracking sensors – moved inside a rectangular arena of 4 m $\times$ 3 m, and their poses (positions and orientations) were being tracked by an overhead camera. For this purpose, rectangular tracking patterns were mounted on top of the robots and the vision system was calibrated in order to provide ground-truth measurements of the robots' poses in a global coordinate

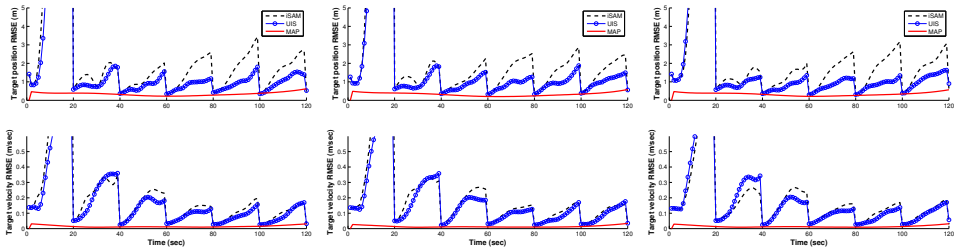


(a) Robot 1 (b) Robot 2 (c) Robot 3



(d) Robot 4 (e) Robot 5 (f) Robot 6

Figure 4: [Bearing-Only Constant-Acceleration] Monte Carlo results for the six tracking robots' poses (positions and orientations). In these plots, the solid lines correspond to the batch MAP, the dashed lines to the standard iSAM, and the solid lines with circles to the proposed UIS.

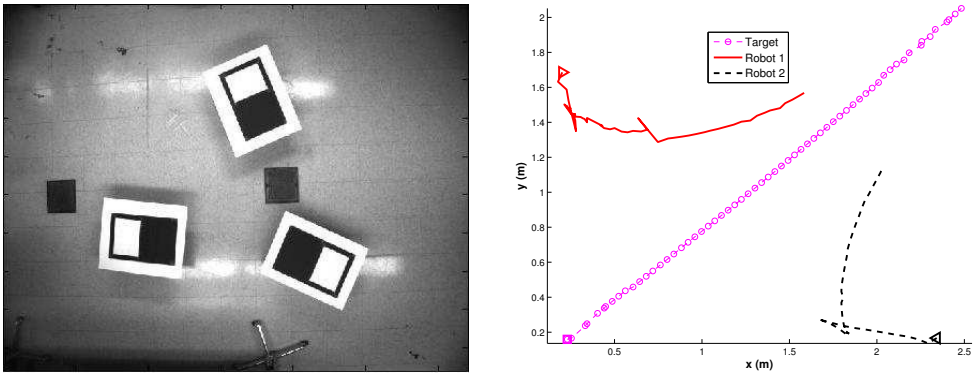


(a) Target 1 (b) Target 2 (c) Target 3

Figure 5: [Bearing-Only Constant-Acceleration] Monte Carlo results for the three targets' states (positions and velocities). In these plots, the solid lines correspond to the batch MAP, the dashed lines to the standard iSAM, and the solid lines with circles to the proposed UIS. Note that we omit the visualization of targets' accelerations in order to preserve consistent and concise presentation as before, while these results are similar to the ones presented here.

Table 2: [Bearing-Only Constant-Acceleration] Robot and Target Estimation Errors

	iSAM [19]	UIS	MAP
		Robot Position RMSE (m)	
Robot 1:	0.7268	0.6213	0.4701
Robot 2:	0.8785	0.7652	0.6837
Robot 3:	0.8203	0.7303	0.5562
Robot 4:	0.7023	0.6507	0.5202
Robot 5:	0.8799	0.8112	0.7260
Robot 6:	0.7554	0.6425	0.4857
		Robot Heading RMSE (rad)	
Robot 1:	0.0971	0.0674	0.0421
Robot 2:	0.0940	0.0714	0.0441
Robot 3:	0.1042	0.0652	0.0369
Robot 4:	0.0706	0.0707	0.0398
Robot 5:	0.0610	0.0698	0.0419
Robot 6:	0.0722	0.0732	0.0424
		Target Position RMSE (m)	
Target 1:	5.0386	2.4039	0.3424
Target 2:	4.6294	2.2274	0.3251
Target 3:	3.2746	2.2967	0.3296
		Target Velocity RMSE (m/sec)	
Target 1:	0.2614	0.1737	0.0147
Target 2:	0.2483	0.1700	0.0140
Target 3:	0.1820	0.1700	0.0141

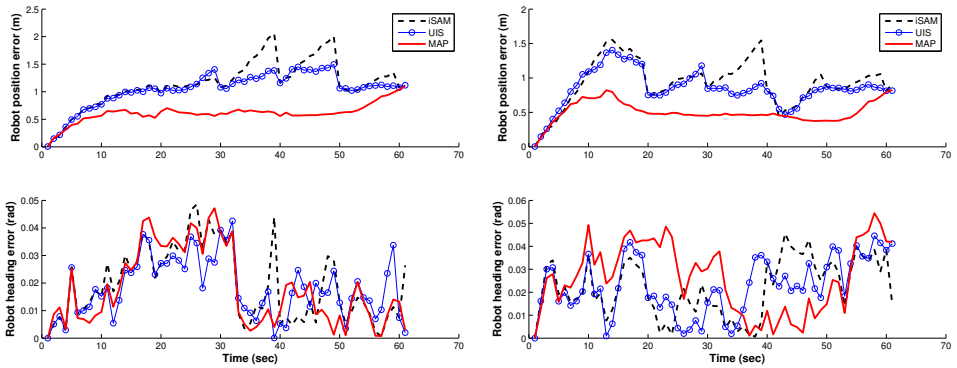


(a) (b)

Figure 6: Experimental setup: (a) Calibrated image of three Pioneer III robots, each of which is mounted a pattern board attached on its top. The target is located at the bottom right of the image, while the other two robots act as tracking sensors. (b) Trajectories of the three robots (target and sensor) that move inside a 4 m $\times$ 3 m arena during the indoor experiment. The symbol $\triangleright$ denotes the starting positions of the tracking robots, while $\square$ is the starting position of the target.

frame. The standard deviation of the noise in these measurements was approximately 0.5 deg for orientation and 0.01 m, along each axis, for position. The target robot was commanded to move along a straight line at a constant velocity of $v = 0.1$ m/sec, and thus a constant-velocity kinematic model with $q = 0.001 \left(\frac{\text{m}}{\text{sec}^2}\right)^2 \frac{1}{\text{Hz}}$ was used to describe this motion [see (53)], while the tracking robots were operated based on the optimal motion planning that minimizes the one-step-ahead uncertainty of the target [39]. Fig. 6(a) shows a snapshot of the experiment, and Fig. 6(b) depicts the true trajectories of the target and the sensor.

In this test, the initial true state of the target, computed from the overhead camera, was $\mathbf{x}_T(0) = \begin{bmatrix} 0.23 & 0.16 & 0.05 & 0.01 \end{bmatrix}^T$, while the initial estimate for the targets state was set to $\mathbf{x}_T(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$. The two tracking robots started at $\mathbf{p}_{R_1}(0) = \begin{bmatrix} 0.20 & 1.69 \end{bmatrix}^T$ and $\mathbf{p}_{R_2}(0) = \begin{bmatrix} 2.34 & 0.17 \end{bmatrix}^T$, respectively. The maximum speed for each tracking robot was set to 0.12 m/sec, and the minimum distance between the target and robots was 1 m. We consider the scenario where each tracking robot measures only relative bearing to the target (as compared to the simulations presented in preceding section where distance-only measurements were used). Relative bearing measurements were produced synthetically using the differences in the positions of



(a) Robot 1 (b) Robot 2

Figure 7: [Experimental Results] Estimation errors for the two tracking robots' poses (positions and orientations). In these plots, the dashed lines correspond to the batch MAP, the dash-dotted lines to the iSAM, and the solid lines to the proposed UIS.

the target robot and the tracking robots, as these were recorded by the overhead camera, with the addition of noise. For the experimental results shown in the following, the bearing measurements were corrupted by zero-mean white Gaussian noise, with standard deviation $\sigma_\theta = 1$ deg.

The same three estimators as in the preceding simulation test are implemented, i.e., the batch MAP, the standard iSAM, and the proposed UIS, while the two incremental smoothers perform periodic batch updates every 10 time steps. The comparative results are shown in Figs. 7 and 8, and Table 3. In particular, Fig. 7 depicts the estimation errors for the two tracking robots and Fig. 8 shows the target results, while Table 3 shows the results averaged over all the time steps. It becomes clear that the batch MAP generally performs the best and well serves as a comparable benchmark, and the proposed UIS achieves better accuracy than the standard iSAM, which agrees with the previous Monte-Carlo simulation results. It should be noted that the proposed UIS outperforms the standard iSAM in both bearing-only and range-only cases, and both constant-velocity and constant-acceleration models, which indicates that the UIS is truly a general incremental estimation algorithm applicable to any measurement/motion model.

8. Conclusions and Future Work

In this paper, we have introduced a new consistent unscented incremental smoothing (UIS) algorithm for multi-robot cooperative target tracking (i.e.,

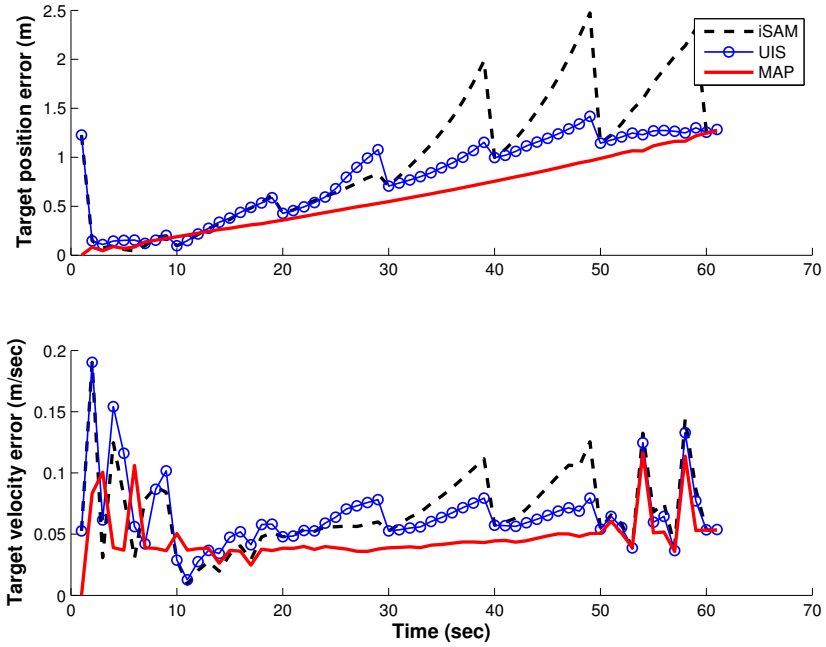


Figure 8: [Experimental Results] Estimation errors for the target state (position and velocity). In these plots, the dashed lines correspond to the batch MAP, the dash-dotted lines to the iSAM, and the solid lines to the proposed UIS.

CLATT problem), which is of practical significance when a team of robots operate in dynamic environments. By performing parameter observability analysis, we have proved that in the case of no prior, the FIM (Hessian) of the optimal batch-MAP based CLATT system has a nullspace of dimension three, corresponding to the global state of the system. However, in general, this is not the case for an incremental smoother (i.e., the naive UIS) that directly employs the canonical unscented transform to numerically compute the inferred, unscented, Jacobians, and thus the Hessians, even though it attains reduced linearization errors. In order to ensure that the Hessian of the UIS has a nullspace of correct dimensions, the proposed UIS explicitly enforces the appropriate observability constraint on the unscented transform when computing Jacobians, which renders the optimal solution as the projection of the canonical Jacobians computed by the standard unscented transform onto the desired observable subspace. The proposed approach has been tested extensively in both Monte Carlo simulations and real-world experimental tests, and is shown to perform significantly better than the standard

Table 3: [Experimental Results] Robot and Target Estimation Errors

	iSAM [19]	UIS	MAP
		Robot Position Estimation Error (m)	
Robot 1:	1.1439	1.0463	0.6126
Robot 2:	0.9303	0.8427	0.5028
		Robot Heading Estimation Error (rad)	
Robot 1:	0.0193	0.0181	0.0179
Robot 2:	0.0218	0.0227	0.0263
		Target Position Estimation Error (m)	
Target:	0.9847	0.7952	0.5971
		Target Velocity Estimation Error (m/sec)	
Target:	0.0678	0.0652	0.0465

incremental smoothing such as iSAM. In the future, we plan to extend the proposed algorithm to 3D, analytically study the effect of the team size on the estimation performance, and also design optimal motion planning within the proposed incremental estimation framework in order to achieve better localization and tracking performance.

Appendix A. Proof of Lemma 4.1

We assume a variable reordering of first all the robot poses, then all the target states. This assumption is employed only to simplify the notation and does not affect the results of the analysis, since the ordering of the variables does not change the rank and nullspace of the Hessian matrix. Using the sparse structure of the robot propagation Jacobian $\mathcal{F}_{R_i}$, we have (note that zeros in $\mathcal{F}_{R_i}$ annihilate the corresponding nonzero submatrices in $\mathbf{N}_b$ in the following matrix multiplication) [see (6) and (8)]:

$$\begin{aligned}
&\mathcal{F}_{R_i}(k-1)\mathbf{N}_b(k) \\
&= \begin{bmatrix} -\mathbf{I}_2 & -\boldsymbol{\Gamma}\left(\hat{\mathbf{p}}_{R_i}(k|k) - \hat{\mathbf{p}}_{R_i}(k-1|k)\right) & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & -1 & \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_i}(k-1|k) \\ \mathbf{0} & 1 \\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{R_i}(k|k) \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \mathbf{0}
\end{aligned} \tag{A.1}$$

Using the sparse structure of the robot-to-robot measurement Jacobian, we have [see (15), (16), and (17)]:

$$\begin{aligned}
&\mathcal{H}_{R_{ij}}(k)\mathbf{N}_b(k) \\
&= \begin{bmatrix} \frac{\hat{\mathbf{p}}_{R_i}^T(k|k)-\hat{\mathbf{p}}_{R_j}^T(k|k)}{||\hat{\mathbf{p}}_{R_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)||} & \mathbf{0} & \frac{\hat{\mathbf{p}}_{R_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)}{||\hat{\mathbf{p}}_{R_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)||} & \mathbf{0} \\ \frac{(\hat{\mathbf{p}}_{R_i}^T(k|k)-\hat{\mathbf{p}}_{R_j}^T(k|k))\mathbf{\Gamma}^T}{||\hat{\mathbf{p}}_{R_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)||^2} & -1 & \frac{(\hat{\mathbf{p}}_{R_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k))\mathbf{\Gamma}^T}{||\hat{\mathbf{p}}_{R_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)||^2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{\Gamma}\hat{\mathbf{p}}_{R_i}(k|k) \\ \mathbf{0} & 1 \\ \mathbf{I}_2 & \mathbf{\Gamma}\hat{\mathbf{p}}_{R_j}(k|k) \\ \mathbf{0} & 1 \end{bmatrix} \\
&= \mathbf{0}
\end{aligned} \tag{A.2}$$

Similarly, we also have [see (19)]:

$$\begin{aligned}
&\mathcal{H}_{T_{ij}}(k)\mathbf{N}_b(k) \\
&= \begin{bmatrix} \frac{\hat{\mathbf{p}}_{R_i}^T(k|k)-\hat{\mathbf{p}}_{T_j}^T(k|k)}{||\hat{\mathbf{p}}_{T_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)||} & \mathbf{0} & \frac{\hat{\mathbf{p}}_{T_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)}{||\hat{\mathbf{p}}_{T_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)||} & \mathbf{0} \\ \frac{(\hat{\mathbf{p}}_{R_i}^T(k|k)-\hat{\mathbf{p}}_{T_j}^T(k|k))\mathbf{\Gamma}^T}{||\hat{\mathbf{p}}_{T_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)||^2} & -1 & \frac{(\hat{\mathbf{p}}_{T_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k))\mathbf{\Gamma}^T}{||\hat{\mathbf{p}}_{T_j}^T(k|k)-\hat{\mathbf{p}}_{R_i}^T(k|k)||^2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{\Gamma}\hat{\mathbf{p}}_{R_i}(k|k) \\ \mathbf{0} & 1 \\ \mathbf{I}_2 & \mathbf{\Gamma}\hat{\mathbf{p}}_{T_j}(k|k) \\ \mathbf{0} & (\mathbf{I}_\alpha \otimes \mathbf{\Gamma})\hat{\mathbf{d}}_{T_j}(k|k) \end{bmatrix} \\
&= \mathbf{0}
\end{aligned} \tag{A.3}$$

where $\alpha = \dim(\mathbf{d}_{T_i})/2$.

In order to show $\mathcal{F}_{T_i}(k-1)\mathbf{N}_b(k) = \mathbf{0}$, we first notice that a stochastic motion model (e.g., constant-velocity model, or constant-acceleration model) for the ith target can be written in the following general form:

$$\mathbf{p}_{T_i}(k) = \mathbf{p}_{T_i}(k-1) + (\mathbf{f_t} \otimes \mathbf{I}_2)\,\mathbf{d}_{T_i}(k-1) + \mathbf{G}_{T_i}(k-1)\mathbf{w}_{T_i}(k-1) \tag{A.4}$$

$$\mathbf{d}_{T_i}(k) = (\mathbf{F_t} \otimes \mathbf{I}_2)\,\mathbf{d}_{T_i}(k-1) \tag{A.5}$$

where δt is the sampling time interval between time-steps $k-1$ and k; $\mathbf{f_t}$ and $\mathbf{F_t}$ are a row vector and a square matrix, respectively, which are functions of δt, and whose expressions depending on the particular stochastic motion model used. Clearly the target propagation Jacobian evaluated is given by:

$$\mathbf{\Phi}_{T_i}(k-1) = \begin{bmatrix} \mathbf{I}_2 & \mathbf{f_t} \otimes \mathbf{I}_2 \\ \mathbf{0} & \mathbf{F_t} \otimes \mathbf{I}_2 \end{bmatrix} \tag{A.6}$$

Based on the structure of this Jacobian as well as the state-estimate propagation equation evaluated at the smoothed state estimates, $\hat{\mathbf{x}}_{T_i}(k-1|k)$ and

$\hat{\mathbf{x}}_{T_i}(k|k)$, and the expected noise vector equal to zero (i.e., $\hat{\mathbf{w}}_{T_i}(k-1|k)=\mathbf{0}$), we have [see (A.6), (A.4), (A.5), and (13)]:

$$
\begin{aligned}
&\mathcal{F}_{T_i}(k-1)\mathbf{N}_b(k)\\
&=\begin{bmatrix}-\mathbf{I}_2 & -\mathbf{f_t}\otimes\mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0}\\ \mathbf{0} & -\mathbf{F_t}\otimes\mathbf{I}_2 & \mathbf{0} & \mathbf{I}_{2\alpha}\end{bmatrix}\begin{bmatrix}\mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_i}(k-1|k)\\ \mathbf{0} & (\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_i}(k-1|k)\\ \mathbf{I}_2 & \boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_i}(k|k)\\ \mathbf{0} & (\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_i}(k|k)\end{bmatrix}\\
&=\begin{bmatrix}\mathbf{0} & -\boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_i}(k-1|k)-(\mathbf{f_t}\otimes\mathbf{I}_2)(\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_i}(k-1|k)+\boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_i}(k|k)\\ \mathbf{0} & -(\mathbf{F_t}\otimes\mathbf{I}_2)(\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_i}(k-1|k)+(\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_i}(k|k)\end{bmatrix}\\
&=\begin{bmatrix}\mathbf{0} & -\boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_i}(k-1|k)-\boldsymbol{\Gamma}(\mathbf{f_t}\otimes\mathbf{I}_2)\hat{\mathbf{d}}_{T_i}(k-1|k)+\boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_i}(k|k)\\ \mathbf{0} & -(\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})(\mathbf{F_t}\otimes\mathbf{I}_2)\hat{\mathbf{d}}_{T_i}(k-1|k)+(\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_i}(k|k)\end{bmatrix}\\
&=\begin{bmatrix}\mathbf{0} & -\boldsymbol{\Gamma}\left[\hat{\mathbf{p}}_{T_i}(k-1|k)-(\mathbf{f_t}\otimes\mathbf{I}_2)\hat{\mathbf{d}}_{T_i}(k-1|k)\right]+\boldsymbol{\Gamma}\hat{\mathbf{p}}_{T_i}(k|k)\\ \mathbf{0} & -(\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})\left[(\mathbf{F_t}\otimes\mathbf{I}_2)\hat{\mathbf{d}}_{T_i}(k-1|k)\right]+(\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})\hat{\mathbf{d}}_{T_i}(k|k)\end{bmatrix}\\
&=\mathbf{0}
\end{aligned}
\tag{A.7}
$$

where in the third equality we have employed the mixed-product property of a Kronecker product [40]:

$$
\begin{aligned}
(\mathbf{f_t}\otimes\mathbf{I}_2)(\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma}) &= (\mathbf{f_t}\mathbf{I}_\alpha)\otimes(\mathbf{I}_2\boldsymbol{\Gamma}) = \mathbf{f_t}\otimes\boldsymbol{\Gamma}\\
&= (1\otimes\boldsymbol{\Gamma})(\mathbf{f_t}\otimes\mathbf{I}_2) = \boldsymbol{\Gamma}(\mathbf{f_t}\otimes\mathbf{I}_2) \qquad \text{(A.8)}\\
(\mathbf{F_t}\otimes\mathbf{I}_2)(\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma}) &= (\mathbf{F_t}\mathbf{I}_\alpha)\otimes(\mathbf{I}_2\boldsymbol{\Gamma}) = (\mathbf{I}_\alpha\otimes\boldsymbol{\Gamma})(\mathbf{F_t}\otimes\mathbf{I}_2) \qquad \text{(A.9)}
\end{aligned}
$$

As an example to illustrate the above identity (A.7), we here consider the constant-velocity motion model (53) that is most commonly used in practice, and whose expressions are given by [see (A.4), and (A.5)]:

$$
\begin{aligned}
\mathbf{p}_T(k) &= \mathbf{p}_T(k-1)+\mathbf{d}_T(k-1)\delta t+\mathbf{G}_{T_i}(k-1)\mathbf{w}_{T_i}(k-1) \qquad \text{(A.10)}\\
\mathbf{d}_T(k) &= \mathbf{d}_T(k-1) \qquad \text{(A.11)}
\end{aligned}
$$

In this case, $\mathbf{f_t}=\delta t$ and $\mathbf{F_t}=1$. Thus, the target propagation Jacobian

becomes $\mathbf{\Phi}_{T_i}(k-1) = \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2\delta t \\ \mathbf{0} & \mathbf{I}_2 \end{bmatrix}$. By noting $\alpha = 1$, we have:

$$\mathcal{F}_{T_i}(k-1)\mathbf{N}_b(k) = \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{I}_2\delta t & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_2 & \mathbf{0} & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{\Gamma}\hat{\mathbf{p}}_{T_i}(k-1|k) \\ \mathbf{0} & \mathbf{\Gamma}\hat{\mathbf{d}}_{T_i}(k-1|k) \\ \mathbf{I}_2 & \mathbf{\Gamma}\hat{\mathbf{p}}_{T_i}(k|k) \\ \mathbf{0} & \mathbf{\Gamma}\hat{\mathbf{d}}_{T_i}(k|k) \end{bmatrix} = \mathbf{0} \tag{A.12}$$

which clearly is a special case of (A.7).

Finally, based on the identities (A.1), (A.7), (A.2), and (A.3), we immediately have the following equivalence [see (24) and (26)]:

$$\mathbf{J}\mathbf{N}_b = \mathbf{0} \;\Rightarrow\; \mathbf{0} = \mathbf{J}^T\mathbf{J}\mathbf{N}_b = \mathbf{A}_b\mathbf{N}_b \tag{A.13}$$

This completes the proof.

Appendix B. Proof of Lemma 5.1

From (50) we know that $\mathbf{H}$ is the left nullspace of the matrix $\mathbf{N}$. By denoting $\mathbf{L}$ the matrix whose rows span this nullspace, we can write $\mathbf{H}$ as:

$$\mathbf{H} = \mathbf{\Theta}\mathbf{L} \tag{B.1}$$

where $\mathbf{\Theta}$ is the unknown matrix we seek to find. Note that there are several possible ways of computing an appropriate matrix $\mathbf{L}$, whose rows lie in the left nullspace of $\mathbf{N}$. For instance, such a matrix is given, in closed form, by the expression:

$$\mathbf{L} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0}_{m\times(n-m)} \end{bmatrix} \left(\mathbf{I}_n - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\right) =: \mathbf{\Psi}\mathbf{\Pi} \tag{B.2}$$

where n is the dimension of the state and m is the dimension of the measurement. It is not difficult to see that $\mathbf{\Pi} := \mathbf{I}_n - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T$ is an orthogonal projection matrix (i.e., $\mathbf{\Pi}^2 = \mathbf{\Pi}$ and $\mathbf{\Pi}^T = \mathbf{\Pi}$) and hence has the eigenvalues of either 1 or 0, whose reduced SVD is given by $\mathbf{\Pi} = \mathbf{U}\mathbf{U}^T$. Using this result, $\mathbf{L}^T$ immediately can be written as $\mathbf{L}^T = \mathbf{U}\mathbf{U}^T\mathbf{\Psi}^T$. By substituting this identity into the cost function (49), we have:

$$\begin{aligned} \min \;\; & ||\mathbf{H}_o - \mathbf{H}||_F^2 = ||\mathbf{U}^T\mathbf{\Psi}^T\mathbf{\Theta}^T - \mathbf{U}^T{\mathbf{H}_o}^T||_F^2 \\ \Rightarrow \;\; & \mathbf{\Theta} = \mathbf{H}_o\mathbf{U}\left(\mathbf{\Psi}\mathbf{U}\right)^{-1} \end{aligned} \tag{B.3}$$

Therefore, substitution of the above equation in (B.1) yields:

$$\begin{aligned}\mathbf{H} &= \mathbf{H}_o\mathbf{U}\left(\mathbf{\Psi}\mathbf{U}\right)^{-1}\mathbf{\Psi}\mathbf{U}\mathbf{U}^T = \mathbf{H}_o\mathbf{U}\mathbf{U}^T = \mathbf{H}_o\mathbf{\Pi} \\ &= \mathbf{H}_o\left(\mathbf{I}_n - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\right) \end{aligned} \tag{B.4}$$

This completes the proof.

Acknowledgements

This work was partially supported by ONR grants N00014-12-1-0093, N00014-10-1-0936, N00014-11-1-0688, N00014-12-10020, and NSF grant IIS-1318392. The authors would like to thank Ke Zhou (Starkey Laboratories) for sharing the experimental data set.

References

[1] A. Davison, N. Kita, Active visual localisation for cooperating inspection robots, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Takamatsu, Japan, 2000, pp. 1709–1715.

[2] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, D. Johanson, Decentralized cooperative aerial surveillance using fixed-wing miniature uavs, Proceedings of the IEEE 94 (7) (2006) 1306 – 1324.

[3] H. Sugiyama, T. Tsujioka, M. Murata, Collaborative movement of rescue robots for reliable and effective networking in disaster area, in: Proc. of the International Conference on Collaborative Computing: Networking, Applications and Worksharing, San Jose, CA, 2005.

[4] C. J. Cannell, D. J. Stilwell, A comparison of two approaches for adaptive sampling of environmental processes using autonomous underwater vehicles, in: Proc. of the MTS/IEEE OCEANS, Washington, DC, 2005, pp. 1514–1521.

[5] R. Kurazume, S. Hirose, An experimental study of a cooperative positioning system, Autonomous Robots 8 (1) (2000) 43–52.

[6] I. M. Rekleitis, G. Dudek, E. E. Milios, Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Switzerland, 2002, pp. 2690–2695.

[7] S. I. Roumeliotis, G. A. Bekey, Distributed multirobot localization, IEEE Transactions on Robotics and Automation 18 (5) (2002) 781–795.

[8] A. Howard, M. J. Mataric, G. S. Sukhatme, Localization for mobile robot teams using maximum likelihood estimation, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and System, Lausanne,Switzerland, 2002, pp. 434–439.

[9] E. D. Nerurkar, S. I. Roumeliotis, A. Martinelli, Distributed maximum a posteriori estimation for multi-robot cooperative localization, in: Proc. of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009, pp. 1402–1409.

[10] D. Fox, W. Burgard, H. Kruppa, S. Thrun, A probabilistic approach to collaborative multi-robot localization, Autonomous Robots 8 (3) (2000) 325–344.

[11] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, Artificial Intelligence 128 (1-2) (2000) 99–141.

[12] F. M. Mirzaei, A. I. Mourikis, S. I. Roumeliotis, On the performance of multi-robot target tracking, in: Proc. of the IEEE International Conference on Robotics and Automation, Rome, Italy, 2007, pp. 3482–3489.

[13] A. Sarwal, D. Agrawal, S. Chaudhary, Surveillance in an open environment by co-operative tracking amongst sensor enabled robots, in: International Conference on Information Acquisition, Seogwipo-si, South Korea, 2007, pp. 345–349.

[14] L. Parker, Cooperative robotics for multi-target observation, Intelligent Automation and Soft Computing 5 (1999) 5–20.

[15] S. Julier, J. Uhlmann, H. F. Durrant-Whyte, A new method for the nonlinear transformation of means andcovariances in filters and estimators, IEEE Transactions on Automatic Control 45 (3) (2000) 477–482.

[16] T. Lefebvre, H. Bruyninckx, J. De Schuller, Comment on "a new method for the nonlinear transformation of means and covariances in filters and estimators" [and authors' reply], IEEE Transactions on Automatic Control 47 (8) (2002) 1406–1409.

[17] F. Dellaert, M. Kaess, Square root SAM: Simultaneous localization and mapping via square root information smoothing, International Journal of Robotics Research 25 (12) (2006) 1181–1203.

[18] G. P. Huang, A. I. Mourikis, S. I. Roumeliotis, An observability constrained sliding window filter for SLAM, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, 2011, pp. 65–72.

[19] M. Kaess, A. Ranganathan, F. Dellaert, iSAM: Incremental smoothing and mapping, IEEE Transactions on Robotics 24 (6) (2008) 1365–1378.

[20] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, g2o: A general framework for graph optimization, in: Proc. of the IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 3607–3613.

[21] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, F. Dellaert, iSAM2: Incremental smoothing and mapping using the Bayes tree, International Journal of Robotics Research 31 (2012) 217–236.

[22] H. Strasdat, J. M. M. Montiel, A. J. Davison, Visual SLAM: Why filter?, Image and Vision Computing 30 (2) (2012) 65–77.

[23] S. Kay, Fundamentals of Statistical Signal Processing, Vol. I - Estimation Theory, Prentice Hall, 1993.

[24] B. Triggs, P. F. McLauchlan, R. I. Hartley, A. W. Fitzgibbon, Bundle adjustment – a modern synthesis, Lecture Notes in Computer Science 1883 (2000) 298–375.

[25] Y. Bar-Shalom, X. R. Li, T. Kirubarajan, Estimation with applications to tracking and navigation, New York: Wiley, 2001.

[26] G. Huang, R. Truax, M. Kaess, J. J. Leonard, Unscented iSAM: A consistent incremental solution to cooperative localization and target tracking, in: Proc. of the European Conference on Mobile Robots, Barcelona, Spain, 2013.

[27] J. Nocedal, S. Wright, Numerical Optimization, Springer, 2006.

[28] G. H. Golub, C. F. V. Loan, Matrix Computations, The Johns Hopkins University Press, 1996.

[29] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, R. Vincent, Efficient sparse pose adjustment for 2D mapping, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 2010, pp. 22–29.

[30] J. Andrade-Cetto, A. Sanfeliu, The effects of partial observability in SLAM, in: Proc. of the IEEE International Conference on Robotics and Automation, New Orleans, LA, 2004, pp. 394–402.

[31] J. Andrade-Cetto, A. Sanfeliu, The effects of partial observability when building fully correlated maps, IEEE Transactions on Robotics 21 (4) (2005) 771–777.

[32] T. Vidal-Calleja, M. Bryson, S. Sukkarieh, A. Sanfeliu, J. Andrade-Cetto, On the observability of bearing only SLAM, in: Proc. of the IEEE International Conference on Robotics and Automation, Roma, Italy, 2007, pp. 4114–4118.

[33] D. Goshen-Meskin, I. Bar-Itzhack, Observability analysis of piece-wise constant systems - Part I: Theory, IEEE Transactions on Aerospace and Electronics Systems 28 (4) (1992) 1056–1067.

[34] K. Lee, W. Wijesoma, J. Guzman, On the observability and observability analysis of SLAM, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006, pp. 3569–3574.

[35] G. P. Huang, A. I. Mourikis, S. I. Roumeliotis, Observability-based rules for designing consistent EKF SLAM estimators, International Journal of Robotics Research 29 (5) (2010) 502–528.

[36] G. P. Huang, N. Trawny, A. I. Mourikis, S. I. Roumeliotis, Observability-based consistent EKF estimators for multi-robot cooperative localization, Autonomous Robots 30 (1) (2011) 99–122.

[37] M. Li, A. I. Mourikis, Improving the accuracy of EKF-based visual-inertial odometry, in: Proc. of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, 2012, pp. 828–835.

[38] G. P. Huang, A. I. Mourikis, S. I. Roumeliotis, A quadratic-complexity observability-constrained unscented Kalman filter for SLAM, IEEE Transactions on Robotics 29 (5) (2013) 1226–1243.

[39] K. Zhou, S. Roumeliotis, Multirobot active target tracking with combinations of relative observations, IEEE Transactions on Robotics 27 (4) (2011) 678 –695.

[40] R. A. Horn, C. R. Johnson, Topics in Matrix Analysis, Cambridge University Press, 1991.