2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Moving Target Estimation and Active Tracking in Multi-Robot Systems

Jie Xu, Pengxiang Zhu, Yanyu Zhang and Wei Ren

Abstract— **In this paper, we propose a comprehensive solution for 3-D active target tracking with multiple robots in a fully distributed setting. Here multiple robots cooperatively estimate their own states and the target's state and actively plan their motions to achieve better estimation of the target. For cooperative localization and target state estimation, each robot maintains a state vector consisting of its own state, the target's state, and its own cloned history states. The challenge of localizing moving robots in 3-D is addressed by using multi-robot cooperative visual-inertial odometry algorithm, which improves the estimation accuracy by using environmental common feature measurements. Each robot's target measurement (if available) and its neighbors' target estimators are then exploited for estimation updates. To preserve and update the correlations between the target and robot states while limiting the influence of bad target estimates on localization accuracy, the Schmidt-Kalman Filter framework is adopted. For motion planning, a gradient-based approach that uses differentiable field-of-view and potential functions is employed to achieve efficient and accurate active target tracking while avoiding collisions and maintaining communication connectivity. Numerous simulations show that our proposed algorithm provides an accurate and efficient solution for cooperative localization and active target tracking.**

I. INTRODUCTION

Multi-robot systems that can cooperatively localize themselves and track targets have numerous applications in surveillance, rescue, and autonomous driving. All the applications will benefit from the active motion planning of the robots known as the active target tracking problem, where the robots equipped with sensors actively plan their trajectories to achieve better estimation of the target.

The literature provides significant attention to the active target tracking problem. The multi-robot target tracking studies in [1]–[5] design centralized control policies. Decentralized control policies are designed in [6], [7] but require multi-hop communication. On the other hand, [8] proposes a flocking-based mobility control model based on the Kalman consensus filter framework to minimize the uncertainty of the target, but the estimation framework requires joint local observability and is limited to 2-D scenarios. Additionally, most of the aforementioned works assume that the robots' states are perfectly known and the control model of the robot is deterministic.

To actively estimate the target state in an efficient way, fully distributed target state estimation is necessary (see [9] and references there). However, these algorithms assume that the robot states are static and known. If the robots are moving in the environment, it is crucial to localize them accurately to achieve successful target tracking. Multi-robot simultaneous localization and mapping (SLAM) algorithms, such as those proposed by [10] and [11], can be used for this purpose. However, such algorithms, especially the mapping part, are computationally intensive and not well-suited for the efficient active target tracking problem. To localize multiple robots using resource-limited platforms, cooperative visual-inertial odometry (CVIO) algorithms can be utilized. The CL-MSCKF algorithm proposed by [12] uses common feature measurements to improve the localization accuracy under the multi-state constrained Kalman filter (MSCKF) framework but in a centralized formulation. More recently, [13] proposes a new CVIO algorithm, which utilizes common feature measurements in a distributed manner. However, all of these localization algorithms do not plan the robots' motions to actively track the target.

Several works have addressed the active joint localization and target tracking (AJLATT) problem, which involves the self-localization of robots and the active tracking of a target. Refs. [14], [15] propose centralized methods while [16] relies on all-to-all communications for solving the problem. Ref. [17] proposes an algorithm that allows decentralized implementation. However, multi-hop communication will be required, rendering it not fully distributed. Our previous work [18] proposes a distributed optimization-based active target tracking algorithm. However, the use of grid search results in heavy computational costs.

In this paper, we present a novel algorithm that integrates cooperative estimation and active motion control. To use environmental features for updating the state vector containing each robot's own localization state and the target state, we employ the CVIO algorithm, which allows the robots to update their estimates based on IMU and camera measurements. To use the target measurements or the neighbors' target estimators while ensuring that these measurements do not negatively affect localization accuracy, we use the Schmidt-Kalman Filter (SKF) framework, which updates the correlations between the target and robot states but does not update the localization part. For planning the robots' motions, we adopt a gradient-based approach that is based on the current and predicted state estimates. We utilize the differentiable field-of-view [19] for minimizing the uncertainty, and potential functions for communication maintenance and collision avoidance. Overall, we offer a distributed solution that is both accurate and efficient for 3-D multi-robot cooperative estimation and active target tracking.

This work was supported by National Science Foundation under Grant CMMI-2027139.

Jie Xu, Pengxiang Zhu, Yanyu Zhang and Wei Ren are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA (jxu150@ucr.edu, pzhu008@ucr.edu, yzhan831@ucr.edu, ren@ee.ucr.edu).

II. Preliminaries

A. Problem Formulation

Consider a team of N robots, represented by the set $\mathcal{V}$. Each robot is equipped with a monocular camera and an IMU. Each camera has a fixed field of view of a pyramid shape, where the camera is located at the apex. The communication range of each robot is given by $\bar{\nu}$. The task of the robots is to cooperatively estimate their own poses and the target pose, and in the meantime, actively plan their trajectories to track the target, utilizing their own measurements of the environmental features and the target (if available) and information obtained through one-hop communication neighbors.

B. Modeling of Motions and Measurements

For the purpose of localization, each robot is modeled by the typical IMU dynamics given by

$$\begin{aligned} {}^{I_i}_{G}\dot{\bar{q}} &= \frac{1}{2}\begin{bmatrix} -({}^{I}\omega_{I_i})_\times & ({}^{I}\omega_{I_i})_\times \\ -({}^{I}\omega_{I_i})^\top_\times & 0 \end{bmatrix} {}^{I_i}_{G}\bar{q}, \\ {}^{G}\dot{p}_{I_i} &= {}^{G}v_{I_i}, \quad {}^{G}\dot{v}_{I_i} = {}^{I_i}_{G}R^\top {}^{I}a_{I_i}, \\ \dot{b}_{a_{I_i}} &= n_{b,a_i}, \quad \dot{b}_{\omega_{I_i}} = n_{b,\omega_i}, \end{aligned} \tag{1}$$

where ${}^{I_i}_{G}\bar{q}$ is the quaternion in JPL format representing the rotation from the global frame G to robot i's IMU frame I_i, ${}^{I_i}_{G}\mathbf{R}$ is the rotation matrix formulation of the quaternion ${}^{I_i}_{G}\bar{q}$, ${}^{G}p_i$ and ${}^{G}v_i$ are, respectively, the position and velocity of IMU in the global frame, ${}^{I}\omega_{I_i}$ and ${}^{I}a_{I_i}$ denote, respectively, the angular velocity and linear acceleration of the IMU represented in its own IMU frame, $b_{a_{I_i}}$ and $b_{\omega_{I_i}}$ denote the biases of the IMU accelerometer and gyroscope, respectively, and n_{b,a_i} and n_{b,ω_i} are white Gaussian noises.

In active target tracking, we need to plan the motion of each robot. For simplicity, we adopt the kinematic model for motion planning to plan the IMU frame angular velocity ${}^{I}\omega_{I_i}$ and linear velocity ${}^{I}v_{I_i}$ to guide the robot's movements. In applications, low-level control commands can be involved to implement ${}^{I}\omega_{I_i}$ and ${}^{I}v_{I_i}$. The kinematic motion model based on ${}^{I}\omega_{I_i}$ and ${}^{I}v_{I_i}$ can be represented as

$$\begin{aligned} {}^{I_i}_{G}\dot{\bar{q}} &= \frac{1}{2}\begin{bmatrix} -({}^{I}\omega_{I_i} + n_{\omega_i})_\times & {}^{I}\omega_{I_i} + n_{\omega_i} \\ -({}^{I}\omega_{I_i} + n_{\omega_i})^\top & 0 \end{bmatrix} {}^{I_i}_{G}\bar{q}, \\ {}^{G}\dot{p}_{I_i} &= {}^{I_i}_{G}R^\top({}^{I}v_{I_i} + n_{v_i}), \end{aligned} \tag{2}$$

where n_{ω_i} and n_{v_i} are white Gaussian noises representing the fact that the velocity commands might not be implemented accurately. Note that (1) is the model used for the state estimation, and (2) is used for the motion planning.

Each robot's camera can capture environmental features as well as the target feature if they are inside its field of view. Consider a feature with position ${}^{C_i}p_f^k = [x_{i,k}, y_{i,k}, z_{i,k}]^\top$ observed in the camera frame of robot i at time t_k. The camera projection function is given by $H_p({}^{C_i}p_f^k) = \frac{1}{z_{i,k}}[x_{i,k}, y_{i,k}]^\top$. Robot i's camera measurement of this feature is given by

$$z_{i,k}^m = H_p({}^{C_i}_{I_i}R\,{}^{I_i,k}_{G}R({}^{G}p_f - {}^{G}p_{I_i,k}) + {}^{C_i}p_{I_i}) + n_{i,k}, \tag{3}$$

where $n_{i,k}$ denotes the white Gaussian noise that corresponds to the measurement.

III. Proposed Algorithm

In this section, we describe our algorithm for 3-D cooperative target state estimation and active target tracking.

A. Cooperative Localization and Target State Estimation

1) Propagation and Cloning for Localization and Target State Estimation: Robot i's state vector at timestep t_k, denoted as $\mathsf{x}_{i,k}$, is defined by

$$\begin{aligned} \mathsf{x}_{i,k} &= [\mathsf{x}_{I_i,k}^\top,\ \mathsf{x}_{T,k}^\top,\ \mathsf{x}_{C_i,k}^\top]^\top, \\ \mathsf{x}_{I_i,k} &= [{}^{I_i,k}_{G}\bar{q}^\top,\ {}^{G}p_{I_i,k}^\top,\ {}^{G}v_{I_i,k}^\top,\ b_{a_i,k},\ b_{\omega_i,k}]^\top, \\ \mathsf{x}_{T,k} &= [{}^{I_{T,k}}_{G}\bar{q}^\top,\ {}^{G}p_{T,k}^\top,\ {}^{G}v_{T,k}^\top]^\top, \\ \mathsf{x}_{C_i,k} &= [{}^{I_i,k-c}_{G}\bar{q}^\top,\ {}^{G}p_{I_i,k-c}^\top \cdots\ {}^{I_i,k-1}_{G}\bar{q}^\top,\ {}^{G}p_{I_i,k-1}^\top]. \end{aligned}$$

where $\mathsf{x}_{I_i,k}$ is robot i's IMU state, $\mathsf{x}_{T,k}$ is the target state, $\mathsf{x}_{C_i,k}$ is the cloned history IMU poses when features are observed by robot i's camera.

Each robot estimates its own state and maintains an estimate of the target. The error state Kalman filter will be used. Throughout the remainder of the paper, we use $\hat{\mathsf{x}}$ to denote the estimate of x, and $\tilde{\mathsf{x}}$ to denote the estimation error. For the orientation error, we use a minimum 3-dimensional representation, denoted as $\tilde{\theta}$, which satisfies $\mathrm{Exp}(\tilde{\theta}_{I_i,k}) = {}^{I_i,k}_{G}\hat{R}^\top {}^{I_i,k}_{G}R$, or $\tilde{\theta}_{I_i,k} = \mathrm{Log}({}^{I_i,k}_{G}\hat{R}^\top {}^{I_i,k}_{G}R)$, with Exp and Log representing the exponential and logarithm map of $\mathrm{SO}(3)$. For other variables, we use the standard additive error definition (e.g. ${}^{G}\hat{p}_{I_i,k} + {}^{G}\tilde{p}_{I_i,k} = {}^{G}p_{I_i,k}$).

When robot i's camera captures an image at timestep t_k, the IMU measurements received over time are collected and utilized to propagate the state estimate $\hat{\mathsf{x}}_{i,k-1|k-1}$ to $\hat{\mathsf{x}}_{i,k|k-1}$ based on the IMU dynamics (1) from $k-1$ to k. Robot i's target state estimate $\hat{\mathsf{x}}_{T_i,k|k-1}$ is propagated from $\hat{\mathsf{x}}_{T_i,k-1|k-1}$ analogously according to the target model analogous to (1). The subscript $k|k-1$ denotes the prediction of the estimate at timestep t_k given the measurement up to timestep t_{k-1}. we can also compute the state transition matrix for the error state vector $\tilde{\mathsf{x}}_{i,k}$ as $\Phi_{i,k}$. The covariance propagation can be calculated as $\mathbf{P}_{i,k|k-1} = \Phi_{i,k}\mathbf{P}_{i,k|k-1}\Phi_{i,k}^\top + \mathbf{U}_{i,k}$, where $\mathbf{U}_{i,k}$ denotes the discrete-time noise covariance for the IMU measurement from timestep t_{k-1} to t_k.

After the propagation step, the clone state vector $\mathsf{x}_{C_i,k}$ is augmented by appending a copy of the current estimated robot's IMU pose.

2) Environmental Features Measurements Update: At timestep t_k, robot i receives its camera measurement of environmental features, as well as from its neighbors. If the measurement of one environmental feature from the robot itself and any of its neighbors are matched, i.e., the same feature is captured, this feature is classified as a common feature. Otherwise, the feature will be classified as an independent feature. If a feature is a common feature, robot i will store its neighbors' estimated states, estimated covariances, and measurements.

To update an independent feature, we will use the standard MSCKF update [20]. On the other hand, if this feature is a common feature, we will also utilize the measurements from the neighbors by following the CVIO update technique [13]. Denote the updated covariance as $\mathbf{P}_{i,k|k}$, and the state correction term as $\delta \mathrm{x}_{i,k}$ which is used to update $\hat{\mathrm{x}}_{i,k|k-1}$ to obtain $\hat{\mathrm{x}}_{i,k|k}$. It is worth noting that as the covariance estimate $\mathbf{P}_{i,k|k-1}$ contains the cross-correlations between the robots' localization and target estimates within the state vector $\hat{\mathrm{x}}_{i,k|k-1}$, the target state estimate is also updated.

3) Target Measurement Update: When a robot captures an image with the target in the field of view, the robot will update its state estimate $\hat{\mathrm{x}}_{i,k|k}$ using the target measurement. The linearized residual system corresponding to the target measurement can be calculated as

$$r_{T,i,k} = \mathbf{H}_{T,i,k}\tilde{\mathrm{x}}_{i,k|k} + n_{T,i,k}, \tag{4}$$

where $r_{T,i,k}$ denotes the residual of the measurement, $\mathbf{H}_{T,i,k}$ is the linearized Jacobian matrix, and $n_{T,i,k}$ denotes measurement noise. Note that in $\mathbf{H}_{T,i,k}$, the columns corresponding to $\tilde{\mathrm{x}}_{C_i,k}$ are all zero. From (4), one idea is to directly update using EKF. However, as the localization from the environmental features update is more accurate than the target state estimation, to avoid the bad influence of the less accurate target estimate on the localization, we propose to utilize the SKF to update the estimator by using (4). The SKF can provide a consistent estimation result by only updating part of the state. We propose to only update the target estimation part by using the target measurement. The Kalman innovation $\mathbf{S}_{T,i,k}$ can be calculated as $\mathbf{S}_{T,i,k} = \mathbf{H}_{T,i,k}\mathbf{P}_{i,k|k}\mathbf{H}_{T,i,k}^{\top} + \mathbf{R}_{T,i,k}$, where $\mathbf{R}_{T,i,k}$ denotes the covariance matrix associated with $n_{T,i,k}$. The traditional Kalman gain $\mathbf{K}_{T,i,k}^{\text{tra}}$ is calculated as $\mathbf{K}_{T,i,k}^{\text{tra}} = \mathbf{P}_{i,k|k}\mathbf{H}_{T,i,k}^{\top}\mathbf{S}_{T,i,k}^{-1}$, which can be written as $\mathbf{K}_{T,i,k}^{\text{tra}} = [\dots, \mathbf{K}_{T_i,k}^{\top}, \dots]^{\top}$, where $\mathbf{K}_{T_i,k}$ denotes the rows that correspond to the target error state $\tilde{\mathrm{x}}_{T_i,k|k}$. Using the SKF idea, we set the rows in $\mathbf{K}_{T,i,k}^{\text{tra}}$ other than $\mathbf{K}_{T_i,k}$ to be zeros. The Kalman gain for the SKF can be written as

$$\mathbf{K}_{T,i,k}^{\text{skf}} = [\mathbf{0}^{\top}, \mathbf{K}_{T_i,k}^{\top}, \mathbf{0}^{\top}]^{\top}, \tag{5}$$

where $\mathbf{0}$ denotes the zero matrices of the same dimension corresponding to the matrix blocks in $\mathbf{K}_{T,i,k}^{\text{tra}}$. The error state correction term $\delta\tilde{\mathrm{x}}_{T,i,k}$ is then calculated as

$$\delta\tilde{\mathrm{x}}_{i,k} = \mathbf{K}_{T,i,k}^{\text{skf}} r_{T,i,k}, \tag{6}$$

and the covariance is updated as

$$\begin{aligned}\mathbf{P}_{i,k|k}^{+} =& (\mathbf{I} - \mathbf{K}_{T,i,k}^{\text{skf}}\mathbf{H}_{T,i,k})\mathbf{P}_{i,k|k}(\mathbf{I} - \mathbf{K}_{T,i,k}^{\text{skf}}\mathbf{H}_{T,i,k})^{\top} \\ &+ \mathbf{K}_{T,i,k}^{\text{skf}}\mathbf{R}_{T,i,k}{\mathbf{K}_{T,i,k}^{\text{skf}}}^{\top},\end{aligned} \tag{7}$$

where the superscript $+$ means that $\mathbf{P}_{i,k|k}$ is updated to a new value. In essence, only the estimate of $\mathrm{x}_{T_i,k}$ in $\mathrm{x}_{i,k}$ is updated while the other parts remain unchanged. The sub-blocks in $\mathbf{P}_{i,k|k}$ corresponding to $\tilde{\mathrm{x}}_{T_i,k}$, as well as its correlation with other states, are updated, while the sub-blocks corresponding to $\tilde{\mathrm{x}}_{I_i,k}$ and $\tilde{\mathrm{x}}_{C_i,k}$ are unchanged.

If the robot does not see the target in the current field of view, the neighbor estimator fusion step is triggered, which updates the current estimate of robot i's state and covariance by using its neighbors' latest estimate of the target. The SKF is also used to allow less influence on the localization estimate. If any of the neighbors has a measurement of the target at the current timestep, that neighbor should perform its own target measurement update first before sending its target estimate to robot i.

After robot i receives its neighbors' current estimates of the target, we perform the SKF update on robot i by treating the neighbors' current estimates of the target as measurements, and their error state covariances as the measurements noises. Denote robot i's neighbors as $j_1, j_2, \dots, j_M$. Specifically, for any neighbor j_m, it has a latest estimator of the target $(\hat{\mathrm{x}}_{T_{j_m},k|k}, \mathbf{P}_{T_{j_m},k|k})$, where $\mathbf{P}_{T_{j_m},k|k}$ is the sub-block that corresponds to the target's error state in j_m's latest covariance estimate. The linearized residual system can be written as $r_{i,j_m,k} = \mathbf{H}_{T,i,j_m,k}\tilde{\mathrm{x}}_{i,k|k} - \tilde{\mathrm{x}}_{T_{j_m},k|k}$, where the entries of position and velocity of the target in $\tilde{\mathrm{x}}_{T_{j_m},k|k}$ is calculated by the standard subtraction as, respectively, ${}^{G}\hat{p}_{T_{j_m},k|k} - {}^{G}\hat{p}_{T_i,k|k}$ and ${}^{G}\hat{v}_{T_{j_m},k|k} - {}^{G}\hat{v}_{T_i,k|k}$, the entries of the orientation error is calculated as $\mathrm{Log}({}^{I_{T_i,k|k}}_{G}\mathbf{R})({}^{I_{T_{j_m},k|k}}_{G}\mathbf{R})^{\top})$, and $\mathbf{H}_{T,i,j_m,k} = [\mathbf{0}, \mathbf{I}, \mathbf{0}]$, with the dimension of the identity matrix $\mathbf{I}$ corresponding to that of the target error state. By stacking the linearized residuals for all j_m, where $m = 1, 2, \dots, M$, we obtain a new system as $r_{i,j,k} = \mathbf{H}_{T,i,j,k}\tilde{\mathrm{x}}_{i,k|k} - \tilde{\mathrm{x}}^{n}_{T_j,k|k}$, where $r_{i,j,k}$ and $\mathbf{H}_{T,i,j,k}$ are obtained by stacking $r_{i,j_m,k}$ and $\mathbf{H}_{T,i,j_m,k}$ for all neighbors of agent i. It is important to note that $\tilde{\mathrm{x}}_{T_j,k|k}$ is subject to influence from other robots' estimates because each robot may have used other robots' estimates to update its own estimator in previous timesteps. As a result, $\tilde{\mathrm{x}}_{T_{j_m},k|k}$ $(m = 1, 2, \dots, M)$, whose covariance matrices are $\mathbf{P}_{T_{j_m},k|k}$, defining neighbors' target estimate covariances, are generally correlated with each other, as well as robot i's own error state $\tilde{\mathrm{x}}_{i,k|k}$, with unknown correlations. Thus, the covariance intersection (CI) algorithm [21] is applied to ensure consistency:

$$\begin{aligned}\hat{\mathbf{P}}_{T,j,k} &= \mathrm{Diag}\Big(\frac{1}{w_{T,j_1,k}}\mathbf{P}_{T_{j_1},k|k}, \dots, \frac{1}{w_{T,j_M,k}}\mathbf{P}_{T_{j_M},k|k},\Big) \\ \mathbf{P}_{i,k|k}^{+} &= \frac{1}{w_{T,i,k}}\mathbf{P}_{i,k|k}\end{aligned} \tag{8}$$

where $w_{T,i,k}$ and $w_{T,j_m,k}$ denote the CI weights satisfying $w_{T,i,k} \geq 0$, $w_{T,j_m,k} \geq 0$, and $w_{T,i,k} + \sum_{m=1}^{M} w_{T,j_m,k} = 1$. The Kalman innovation is calculated as $\mathbf{S}_{T,i,j,k} = \mathbf{H}_{T,i,j,k}\mathbf{P}_{i,k|k}^{+}\mathbf{H}_{T,i,j,k}^{\top} + \hat{\mathbf{P}}_{T,j,k}$, The traditional Kalman gain is calculated as $\mathbf{K}_{T,i,j,k}^{\text{tra}} = \mathbf{P}_{i,k|k}^{+}\mathbf{H}_{T,i,j,k}^{\top}\mathbf{S}_{T,i,j,k}^{-1}$. To make use of the SKF to update the estimate of the target at robot i, we adopt a similar approach to that in (5) to construct $\mathbf{K}_{T,i,j,k}^{\text{skf}}$. Then the error state and covariance update can be written analogously as (6) and (7).

B. Active Target Tracking

After the measurements update step, the control inputs are generated at each robot in order to actively track the target at timestep t_k. The planning horizon is given as Z. Based on the predicted robot's state as a function of its control inputs, the motion of the robot can be designed such that the robot is able to follow the target and have the target in its field of view effectively.

Let $u_{i,k} = [{}^I\omega_{I_i,k}^\top, {}^Iv_{I_i,k}^\top]^\top$. At the first timestep, i.e. $k = 0$, the initial control inputs from timestep t_k to t_{k+Z-1}, denoted as $[u_{i,k}^\top, u_{i,k+1}^\top, \dots, u_{i,k+Z-1}^\top]^\top$, are generated randomly. At $k > 0$, to initialize the control inputs from t_k to t_{k+Z-1}, the first $Z-1$ entries are inherited from the calculated results in the previous timestep t_{k-1}, and the last entry $u_{i,k+Z-1}$ is generated randomly. Define $\bar{\mathsf{x}}_{i,k} = [{}_G^{I_{i,k|k}}\hat{\bar{q}}^\top, {}^G\hat{p}_{I_i,k|k}^\top, {}_G^{I_{T_i,k|k}}\hat{\bar{q}}^\top, {}^G\hat{p}_{T_i,k|k}^\top]^\top$ as the pose estimate of the robot and the target, and $\bar{\mathbf{P}}_{i,p,k}$ as the covariance estimate by selecting the corresponding blocks in the original covariance matrix $\mathbf{P}_{i,k|k}$ at the current timestep k.

Define the propagated stacked vector of predicted states as $\bar{\mathsf{x}}_{i,k:k+z} = [\bar{\mathsf{x}}_{i,k}^\top, \dots, \bar{\mathsf{x}}_{i,k+z}^\top]^\top$ with the corresponding covariance $\bar{\mathbf{P}}_{i,p,k:k+z}$. Given the control inputs, $\bar{\mathsf{x}}_{i,k:k+z}$ can be obtained by appending calculated predicted states according to (2) by setting noises to 0, and assuming that the target propagates with the same angular and linear velocities as timestep k. The covariance update for each timestep can be written as

$$\bar{\mathbf{P}}_{i,p,k:k+z} = \Phi_{i,p,k+z}\bar{\mathbf{P}}_{i,p,k:k+z-1}\Phi_{i,p,k+z}^\top + \mathbf{U}_{p,k+z}, \quad (9)$$

where $\mathbf{U}_{p,k+z}$ is the propagation noise covariance at the prediction timestep t_{k+z}, and $\Phi_{i,p,k+z}$ denotes the state transition matrix. Note that the predicted covariance is augmented at each timestep, and the final predicted covariance $\bar{\mathbf{P}}_{i,p,k:k+Z}$ corresponds to the stacked vector $\bar{\mathsf{x}}_{i,k:k+Z}$.

To calculate a set of control inputs that works better for our goal, we adopt the differentiable field of view [19] and use a gradient descent algorithm to minimize a cost which is a combination of the log-determinant of the final predicted covariance $\bar{\mathbf{P}}_{i,p,k:k+Z}$ and the defined potential functions among robots and the target. With the field of view of the camera, we can calculate the estimated distance $d_{i,k+z}$ of the target to the nearest boundary of the field of view from the predicted states. The distance is set to be positive if the target is in the field of view of the robot, and negative otherwise. A coefficient $\alpha_{i,k+z} \in (0,1)$ is defined by using the logistic function which depends on $d_{i,k+z}$ as

$$\alpha_{i,k+z} = \text{logis}(d_{i,k+z}) = \frac{1}{1+e^{-qd_{i,k+z}}} \quad (10)$$

where q is a parameter defining the stiffness. The logistic function is close to 1 as $d_{i,k+z} \gg 0$, and is close to 0 as $d_{i,k+z} \ll 0$. Denote $\mathbf{M}_{T,i,k+z,p} = \mathbf{H}_{T,i,k+z,p}^\top \mathbf{R}_{T,i,k+z,p}^{-1} \mathbf{H}_{T,i,k+z,p}$ as the information gain for the measurement at each timestep, where $\mathbf{H}_{T,i,k+z,p}$ denotes the measurement Jacobian matrix corresponding to each prediction timestep t_{k+z} of robot i, and $\mathbf{R}_{T,i,k+z,p}$ is the known measurement noise covariance. The coefficient $\alpha_{i,k+z}$ is multiplied in front of the information gain $\mathbf{M}_{T,i,z,p}$ to make the information gain differentiable. As a result, the update of the final covariance matrix becomes

$$\bar{\mathbf{P}}_{i,p,k:k+Z}^{-1} \leftarrow \bar{\mathbf{P}}_{i,p,k:k+Z}^{-1} + \sum_{z=1}^{Z} \alpha_{i,k+z}\mathbf{M}_{T,i,k+z,p}. \quad (11)$$

In the meantime, the robots should avoid collision as well as maintain communication connections when generating their control inputs. Let $\mathcal{N}_{i,k}$ denote the set that includes all neighbors of i (excludes i itself) at time t_k. At the predicted planning timestep t_{k+z}, denote the predicted distance between robots i and j, where $j \in \mathcal{N}_{i,k} \cup \{T\}$, to be $\nu_{i,j,k+z}$. We introduce potential function $h(\nu_{i,j,k+z})$ that can characterize $\nu_{i,j,k+z}$. Let the safe distance between robots and the target to avoid collisions be denoted as $\underline{\nu}$, and let the communication range of any two robots be denoted as $\bar{\nu}$. We define two types of potential functions depending on whether robots are neighbors from timestep 0 or not [18].

Define the cost c_i^k as $c_i^k = c_{i,1}^k + c_{i,2}^k$, where

$$\begin{aligned} c_{i,1}^k &= \log(\det(\bar{\mathbf{P}}_{i,p,k:k+Z})), \\ c_{i,2}^k &= \sum_{z=1}^{Z}\Big(\sum_{j\in\mathcal{N}_{i,k}\cup\{T\}} h(\nu_{i,j,k+z})\Big). \end{aligned} \quad (12)$$

To simplify the procedure of deriving gradients, we use the equivalent SE(3) representation of (2). Let $\bar{\mathbf{X}}_{i,k} = \begin{bmatrix} {}_G^{I_i,k}\hat{R}^\top & {}^G\hat{p}_{I_i,k} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$ be the estimate in SE(3) at time t_k. It follows that $\bar{\mathbf{X}}_{i,k+1} = \bar{\mathbf{X}}_{i,k}\text{Exp}(\tau_k u_{i,k})$, where τ_k is the sampling period. The gradient of $c_{i,1}^k$ with respect to the sth entry $u_{i,k+z-1}^{(s)} \in [{}^I\omega_{k+z-1}^\top, {}^Iv_{k+z-1}^\top]^\top$ can be calculated from [19] as

$$\begin{aligned} \frac{\partial c_{i,1}^k}{\partial u_{i,k+z-1}^{(s)}} &= -\text{trace}(\bar{\mathbf{P}}_{i,p,k:k+Z} \\ &\quad \sum_{\gamma=k+z}^{k+Z}\frac{\partial \alpha_{i,\gamma}}{\partial u_{i,k+z-1}^{(s)}}\mathbf{M}_{T,i,\gamma,p} + \sum_{\gamma=k+z}^{k+Z}\alpha_{i,\gamma}\frac{\partial \mathbf{M}_{T,i,\gamma,p}}{\partial u_{i,k+z-1}^{(s)}}), \\ \frac{\partial \alpha_{i,\gamma}}{\partial u_{i,k+z-1}^{(s)}} &= \frac{\partial \alpha_{i,\gamma}}{\partial d_{i,\gamma}}\frac{\partial d_{i,\gamma}}{\partial \bar{\mathbf{X}}_{T,i,\gamma}}\frac{\partial \bar{\mathbf{X}}_{T,i,\gamma}}{\partial u_{i,k+z-1}^{(s)}}, \\ \frac{\partial \bar{\mathbf{X}}_{T,i,\gamma}}{\partial u_{i,k+z-1}^{(s)}} &= -\mathbf{Q}\bar{\mathbf{X}}_{i,\gamma}^{-1}\frac{\partial \bar{\mathbf{X}}_{i,\gamma}}{\partial u_{i,k+z-1}^{(s)}}\bar{\mathbf{X}}_{i,\gamma}^{-1}\begin{bmatrix}\bar{p}_{T_i,\gamma} \\ 1\end{bmatrix} \end{aligned} \quad (13)$$

where $\bar{\mathbf{X}}_{T,i,\gamma} = \mathbf{Q}\bar{\mathbf{X}}_{i,\gamma}^{-1}[\bar{p}_{T_i,\gamma}^\top, 1]^\top$ represents the target state in the robot i's body frame using the predicted states from $\bar{\mathsf{x}}_{i,k:k+Z}$, and $\mathbf{Q}$ is defined as $\mathbf{Q} = [\mathbf{I}_3, \mathbf{0}_{3\times 1}]$. Moreover, the linearization states used to calculate $\mathbf{M}_{T,i,\gamma,p}$ is assumed to be unchanged during every iteration, and thus the partial fraction $\frac{\partial \mathbf{M}_{T,i,\gamma,p}}{\partial u_{i,k+z}^{(s)}}$ is set to be zero in our setting.

To calculate the gradient of $c_{i,2}^k$, note that the predicted distance $\nu_{i,j,k+z}$ is a function of robot i's control inputs, as well as robot j's control inputs, up to prediction time t_{k+z}. Although the gradient of $c_{i,2}^k$ with respect to robot j's control

inputs should also be considered, we do not have access to them at robot i. As a result, we use the initial control input of robot j for the calculation throughout the process. Then the gradient of $c_{i,2}^k$ is calculated as

$$\frac{\partial c_{i,2}^k}{\partial u_{i,k+z-1}^{(s)}} = \sum_{\gamma=k+1}^{k+Z} \sum_{j \in \mathcal{N}_{i,k} \cup \{T\}} \frac{\partial h}{\partial \nu_{i,j,\gamma}} \frac{\partial \nu_{i,j,\gamma}}{\partial u_{i,k+z-1}^{(s)}}, \\ \frac{\partial \nu_{i,j,\gamma}}{\partial u_{i,k+z-1}^{(s)}} = \frac{\partial \nu_{i,j,\gamma}}{\partial {}^G\bar{p}_{I_i,\zeta}} \frac{\partial {}^G\bar{p}_{I_i,\zeta}}{\partial u_{i,k+z-1}^{(s)}}, \tag{14}$$

where ${}^G\bar{p}_{I_i,\zeta}$ is the corresponding position elements in $\bar{\mathrm{X}}_{i,\gamma}$.

As a result, the gradient of c_i^k can be computed from (13) and (14) and the control inputs are updated by

$$u_{i,k+z-1}^{(s)} \leftarrow \text{Projection}(u_{i,k+z-1}^{(s)} - \beta^{(s)} \frac{\partial c_{i,}^k}{\partial u_{i,k+z-1}^{(s)}}), \tag{15}$$

where $\beta^{(s)}$ is a fixed step size, and "Projection" projects the calculated result onto the feasible set. With the new control inputs, new iterations need to be performed following the steps shown in equations from (9) to (15) until the result converges. At timestep t_k, each robot i applies $u_{i,k}$. At timestep t_{k+1}, each robot applies the resulting control input $u_{i,k}$ and replans future control inputs.

IV. Simulation Results

In this section, we use Monte Carlo simulations to demonstrate the performance of our algorithm.

A. Simulation Settings

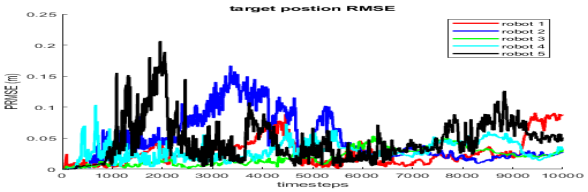
We consider a scenario where 5 robots and one target move in a 3-D environment. The robots are equipped with IMUs and cameras to estimate the target state. The camera measurements $z_{i,k}^m$ in (3) are assumed to be corrupted by zero-mean Gaussian noises $\mathcal{N}(0, \sigma_{z_i}^2)$, which is calculated from one-pixel noise and camera intrinsics. The noise models for the IMUs can be written as:

$${}^I\omega_{I_i,k}^m = {}^I\omega_{I_i,k} + n_{I_i,k,\omega}, \;\; n_{I_i,k,\omega} \sim \mathcal{N}(0, \sigma_{\omega_{I_i}}^2), \\ {}^I a_{I_i,k}^m = {}^I a_{I_i,k} - ({}_G^{I_i,k}\mathbf{R})g + n_{I_i,k,a}, \;\; n_{I_i,k,a} \sim \mathcal{N}(0, \sigma_{a_{I_i}}^2),$$

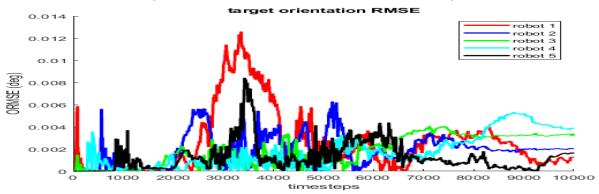
where $i = 1, 2, ..., 5$, k denotes the timestep t_k, $g = [0, 0, 9.81]^\top$ is the gravity, $\sigma_{\omega_{I_i}} = 1.122 \times 10^{-4}\ \mathbf{I}_3$, and $\sigma_{a_{I_i}} = 5.0119 \times 10^{-3}\ \mathbf{I}_3$.

In each camera frame, the sensing zone is a fixed pyramid shape in the camera frame. The five points of the pyramid in the camera frame are located at $[0, 0, 0]^\top$ and $[-5, -8, 8]^\top$, $[-5, 8, 8]^\top$, $[5, -8, 8]^\top$, $[5, 8, 8]^\top$, respectively. At each time step, we generate linear and angular velocity control inputs using our active tracking algorithm for the next 3 time steps $Z = 3$. We then apply the actual control input for the next time step by adding Gaussian noises to the calculated inputs to account for noisy control inputs applied to the robots.

We specify the communication range $\bar{\nu}$ between the robots to be 12 meters, and the minimum safe distance $\underline{\nu}$ between robots to be 1.5 meters. The initial positions of the five robots are set to be $[0.1, -0.1, 0]^\top$, $[-10.1, 0.1, 8.1]^\top$, $[0.1, 0.2, 8.1]^\top$, $[-9.9, 0.1, 2.5]^\top$, and $[-19.9, 0.1, 2.5]^\top$, respectively. In this setting, the initial communication topology is connected. We set the initial estimate for each robot's state to be the corresponding true state, i.e., $\hat{x}_{I_i,0} = x_{I_i,0}$. We also provide an initial covariance estimate of $10^{-3}\mathbf{I}_{15}$ for each robot's state. For the target estimate, we set the initial estimate to be the ground truth, and we set the initial covariance matrix to be $10^{-3}\mathbf{I}_9$.



(a) Average PRMSE of each robot's target estimate



(b) Average ORMSE of each robot's target estimate

Regarding the CI weights in (8), similar to the CVIO weight selection [13], we set a fixed weight of $\omega_{T,i,k} = 0.99$ for robot i itself, and evenly distribute the remaining weight among its neighbors to ensure that they sum up to 1.

B. Simulation Results

We conducted 20 Monte Carlo simulations to evaluate the accuracy of the estimation. The root-mean-square error (RMSE) was used as the performance metric to evaluate the localization accuracy of the robots and the target state estimation. Specifically, the position and orientation RMSEs were calculated and analyzed.

The average robot position RMSE (PRMSE) and orientation RMSE (ORMSE) are calculated to be 2.7×10^{-2} m and 1×10^{-3} rad, respectively, indicating a high level of accuracy in the localization performance of our robots. Therefore, we conclude that our cooperative localization algorithm provides a satisfactory level of accuracy.

Furthermore, we evaluate the performance of the target tracking in our cooperative estimation and active target tracking algorithm. Figures 1a and 1b present the target tracking performance of each robot, with an average PRMSE of 4.6×10^{-2} m and an average ORMSE of 1.3×10^{-3} rad for all robots' estimates of the target. We compare cooperative and non-cooperative target tracking by presenting the average PRMSE without cooperation in Figure 2. It can be observed that the robots may fail to obtain an accurate estimate of the target and lose track. Our results show that the robots' performance is better when they cooperate to track the target. We further present the outcomes of our target tracking approach in a scenario where the robots move in a random manner within the environment. To simulate the robot's behavior, we employ Gaussian processes to generate

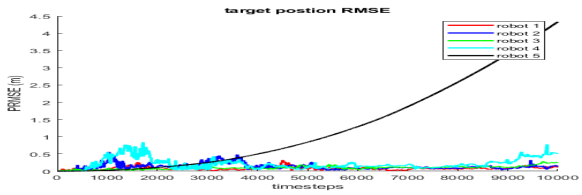


Fig. 2: Average PRMSE of each robot's target estimate without cooperation

linear velocity and angular control commands for each robot, with standard deviations of $2 \times 10^{-1}\mathbf{I}_3$ and $5 \times 10^{-2}\mathbf{I}_3$, respectively. We show the PRMSE of the target estimate in Figure 3, demonstrating that the proposed active target tracking technique is effective in reducing the uncertainty of the target estimate in comparison to random motion.

In Figure 4a, we show the trajectories of 5 robots and the target from the first Monte Carlo iteration. Figure 4b includes the distances of the robots to the target. The robots effectively approach and follow the target, improving target estimation even from distant starting points. Notably, the minimum inter-robot distance is consistently 3.9 meters, ensuring collision avoidance.

V. Conclusion

In this paper, we have presented a distributed approach to address the coupled problem of cooperative localization, target tracking, and active motion planning. We used Monte Carlo simulations to show the effectiveness of our algorithm. Our comparative analysis between the cooperative and non-cooperative estimation performance, as well as the active motion planning and random walk case, has clearly demonstrated the significant improvements achieved through using our algorithm. Our contribution is to provide a solution to the problem that is both practical and effective.

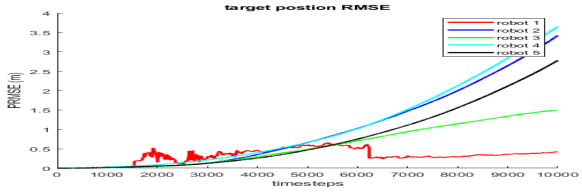
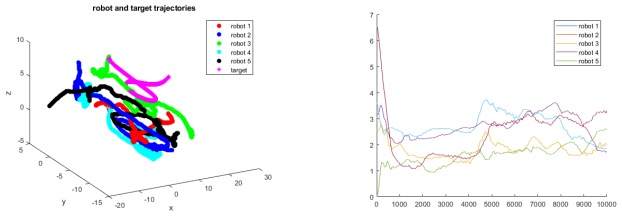


Fig. 3: Average PRMSE of each robot's target estimate with random motion of the robots



(a) Trajectories of the robots and the target (run 1)

(b) Distance of each robot to the target (run 1)

Fig. 4: Trajectories and distances

References

[1] P. Tokekar, V. Isler, and A. Franchi, "Multi-target visual tracking with aerial robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3067–3072.

[2] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: taxonomy and survey," *Autonomous Robots*, vol. 40, pp. 729–760, 2016.

[3] A. W. Stroupe and T. Balch, "Value-based action selection for observation with robot teams using probabilistic techniques," *Robotics and Autonomous Systems*, vol. 50, no. 2-3, pp. 85–97, 2005.

[4] K. Zhou, S. I. Roumeliotis *et al.*, "Multirobot active target tracking with combinations of relative observations," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 678–695, 2011.

[5] B. Charrow, N. Michael, and V. Kumar, "Cooperative multi-robot estimation and control for radio source localization," *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 569–580, 2014.

[6] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.

[7] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1025–1032, 2018.

[8] R. Olfati-Saber and P. Jalalkamali, "Coupled distributed estimation and control for mobile sensor networks," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2609–2614, 2012.

[9] J. Xu, P. Zhu, and W. Ren, "Distributed invariant extended Kalman filter for 3-d dynamic state estimation using Lie groups," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 2367–2372.

[10] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, 2006.

[11] S. Thrun and Y. Liu, "Multi-robot slam with sparse extended information filers," in *Robotics Research. The Eleventh International Symposium: With 303 Figures*. Springer, 2005, pp. 254–266.

[12] I. V. Melnyk, J. A. Hesch, and S. I. Roumeliotis, "Cooperative vision-aided inertial navigation using overlapping views," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 936–943.

[13] P. Zhu, Y. Yang, W. Ren, and G. Huang, "Cooperative visual-inertial odometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 135–13 141.

[14] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *2014 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 6447–6454.

[15] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. S. Sukhatme, "Cooperative multi-robot control for target tracking with onboard sensing," *The International Journal of Robotics Research*, vol. 34, no. 13, pp. 1660–1677, 2015.

[16] F. Morbidi and G. L. Mariottini, "Active target tracking and cooperative localization for teams of aerial vehicles," *IEEE transactions on control systems technology*, vol. 21, no. 5, pp. 1694–1707, 2012.

[17] F. Meyer, H. Wymeersch, M. Fröhle, and F. Hlawatsch, "Distributed estimation with information-seeking control in agent networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2439–2456, 2015.

[18] S. Su, P. Zhu, and W. Ren, "Multirobot fully distributed active joint localization and target tracking," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 4, pp. 1594–1606, 2023.

[19] S. Koga, A. Asgharivaskasi, and N. Atanasov, "Active exploration and mapping via iterative covariance regulation over continuous se (3) trajectories," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2735–2741.

[20] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE international conference on robotics and automation*. IEEE, 2007, pp. 3565–3572.

[21] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, "Decentralized multi-robot cooperative localization using covariance intersection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1412–1417.