

An Optimization Approach to Fully Distributed Active Joint Localization and Target Tracking in Multi-Robot Systems

Shaoshu Su, Pengxiang Zhu and Wei Ren

Abstract—In this paper, the problem of multi-robot active joint localization and target tracking (AJLATT) is studied. Here a team of mobile robots equipped with sensors of limited field of view is driven to *actively* and cooperatively estimate their own and the target's states. A motion strategy is designed to drive each robot to achieve better self-localization and target-tracking performance while avoiding collisions. By leveraging our previous work on joint localization and target tracking (JLATT), we propose a fully distributed optimization-based algorithm to solve the problem of AJLATT, where each robot uses only its own and one-hop communicating neighbors' information. The proposed algorithm aims to minimize an objective function which includes both the target's and robots' estimation uncertainties and as well as a potential function which is used for collision avoidance and communication connectivity maintenance. Monte Carlo simulations are performed to demonstrate the effectiveness of our algorithm. As compared to JLATT with random motion, the accuracy of both localization and target tracking is improved.

I. INTRODUCTION

Autonomous multi-robot systems have wide applications such as search, rescue, and region monitoring. However, in some scenarios where a GPS signal, motion-capture system, or prior map is not available, the positioning of the multi-robot system and the target becomes a critical problem. There are many results on multi-robot cooperative localization or pure target tracking¹ or both. We refer the task that jointly estimates the states of a team of robots and the target as joint localization and target tracking (JLATT). There are some existing works that solve the JLATT problem, both in a centralized [1]–[3] or fully distributed manner [4], [5].² However, all the aforementioned works [1]–[5] have a assumption that the sensors or robots are either static or move randomly without active motion control.

Some algorithms have been proposed to solve the active target tracking problem from the control or optimization perspective. In the controls field, there are numerous results on distributed tracking or leader-follower tracking (see, e.g., [6]). However, in these results, both the robots' and target's states are usually assumed to be known, and the emphasis is

on designing distributed controllers. In [7], the target's state is assumed to be unknown, but an all-to-all communication network is required. In [8], an information-driven flocking algorithm is proposed to achieve the distributed target state estimation, but the estimator relies on a restrictive assumption that the target is jointly observed by each robot and its neighbors. For the optimization-based approaches, the idea is to find control to optimize certain objective functions in a centralized manner [9]–[12] or a decentralized manner but with multi-hop information transmission required [13]. In all the works above, the robots' states are required to be known, which is not so practical in real-world applications.

There exist some works addressing the active joint localization and target tracking (AJLATT) problem. In these works, the improvement of the target tracking and/or robot self-localization performance can be actively achieved by designing certain motion strategies for the robots. Considering limited sensing capabilities, [14] proposes an algorithm which plans robots' motions and switches the team's sensing topology to minimize the uncertainty of the target's state estimate. In [15], a gradient-based control strategy is designed to minimize the estimation uncertainty of both the robots' and the target's states. While these two works take jointly estimating both robots' states and target's state into consideration, both approaches utilize a centralized estimation framework. Although centralized approaches are well known for their optimal estimation and planning performance, a heavy burden in computation and the strict requirement on the communication topology limit their applications in real-world scenarios. In [16], a distributed method is proposed to solve the AJLATT problem. However, the method is not fully distributed since multi-hop information transmission is required. To the best of our knowledge, there is no existing fully distributed algorithm solving the AJLATT problem.

Considering all the limitations of the previous works, we aim to propose a fully distributed optimization-based algorithm to solve the problem of AJLATT. Our previous work [5] introduces a framework to distributively solve the pure joint localization and target tracking (JLATT) problem without designing motion strategies for the robots. Leveraging that framework, in this paper we design a fully distributed motion strategy for each robot so that each robot can actively improve its self-localization and target tracking performance. Using only its own and one-hop communicating neighbors' information, each robot selects its motion strategy by minimizing an objective function which includes both the target's and robots' estimation uncertainties and as well as a potential

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

Shaoshu Su, Pengxiang Zhu and Wei Ren are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521 USA. shaoshu.su@email.ucr.edu, pzhu008@ucr.edu, ren@ee.ucr.edu.

¹Here the term *target tracking* purely refers to target state estimation.

²We use the term *fully distributed* to describe an algorithm that uses only a robot's own and one-hop neighbors' information without the requirement for global parameters, multi-hop information transmission, or multiple communication iterations on certain quantities for iterative consensus-type calculations per time instant in the execution stage.

function used for collision avoidance and communication connectivity maintenance. To the best of our knowledge, our algorithm is the first fully distributed approach to solve the AJLATT problem. Extensive Monte-Carlo simulations are performed to validate its performance.

II. PRELIMINARIES

A. Motion and Measurement Models

We consider the scenario where M robots track a target on a surface. We use the vectors $\mathbf{x}_i^k$ and $\mathbf{x}_T^k$ to represent, respectively, the true state of robot i and the target at time k . Their movements are driven by a nonlinear motion model as

$$\mathbf{x}_i^k = f_i(\mathbf{x}_i^{k-1}, \mathbf{u}_i^{k-1}, \mathbf{w}_i^{k-1}), \quad (1)$$

$$\mathbf{x}_T^k = g(\mathbf{x}_T^{k-1}, \mathbf{u}_T^{k-1}, \mathbf{w}_T^{k-1}), \quad (2)$$

where $\mathbf{u}_i^{k-1}$ and $\mathbf{u}_T^{k-1}$ are, respectively, the control inputs for robot i and the target. $\mathbf{w}_i^{k-1} \sim \mathcal{N}(0, \mathbf{Q}_i^{k-1})$ and $\mathbf{w}_T^{k-1} \sim \mathcal{N}(0, \mathbf{Q}_T^{k-1})$ are, respectively, the zero-mean white Gaussian process noises for robot i and the target.

We define $\bar{\mathbf{x}}_i^k$ and $\hat{\mathbf{x}}_i^k$ as, respectively, robot i 's prior and posterior estimates of its true state $\mathbf{x}_i^k$, and $\bar{\mathbf{x}}_{T_i}^k$ and $\hat{\mathbf{x}}_{T_i}^k$ as, respectively, each robot i 's prior and posterior estimates of the target's state $\mathbf{x}_T^k$. Their corresponding approximated prior and posterior covariances are defined as $\bar{\mathbf{p}}_i^k$, $\hat{\mathbf{p}}_i^k$, $\bar{\mathbf{p}}_{T_i}^k$, and $\hat{\mathbf{p}}_{T_i}^k$, respectively.

At time k , if robot j or the target is within the sensing region of robot i , robot i can obtain a robot-to-robot measurement $\mathbf{z}_{R_{ij}}^k$ or a robot-to-target measurement $\mathbf{z}_{RiT}^k$. The measurement models are defined as

$$\begin{aligned} \mathbf{z}_{R_{ij}}^k &= h_{ij}(\mathbf{x}_i^k, \mathbf{x}_j^k) + \mathbf{v}_{R_{ij}}^k, \\ \mathbf{z}_{RiT}^k &= h_{iT}(\mathbf{x}_i^k, \mathbf{x}_T^k) + \mathbf{v}_{RiT}^k, \end{aligned} \quad (3)$$

where $\mathbf{v}_{R_{ij}}^k \sim \mathcal{N}(0, \mathbf{R}_{ij}^k)$ and $\mathbf{v}_{RiT}^k \sim \mathcal{N}(0, \mathbf{R}_{iT}^k)$ are the measurement noises assumed to be zero-mean white Gaussian. The measurement noises are assumed to be mutually uncorrelated across robots and uncorrelated with the process noises.

B. Graphs

In the team of M robots, a directed communication graph $G_c^k = (\mathcal{V}, \mathcal{E}_c^k)$ is defined, where $\mathcal{V} = \{R_1, \dots, R_M\}$ is the robot set and $\mathcal{E}_c^k \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set representing the communication links between robots at time k . If robot i receives information from robot j at time k , then a directed edge (j, i) exists in $\mathcal{E}_c^k$. We assume that the self edge (i, i) exists in $\mathcal{E}_c^k$, $\forall i \in \mathcal{V}$, which means robot i can also use the information from itself. At time k , the communicating neighbor set of robot i is defined as $\mathcal{N}_{c,i}^k = \{l | (l, i) \in \mathcal{E}_c^k, \forall l \neq i, l \in \mathcal{V}\}$. Then, the inclusive communicating neighbor set of robot i is $\mathcal{I}_{c,i}^k = \mathcal{N}_{c,i}^k \cup \{i\}$. Similarly, we define a directed sensing graph $G_s^k = (\mathcal{V}, \mathcal{E}_s^k)$ to describe robot-to-robot measurements, where $\mathcal{E}_s^k \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, which represents the detection links between robots at time k . If robot i can detect robot j at time k , a directed edge (j, i) exists in $\mathcal{E}_s^k$. At time k , we define robot i 's

sensing neighbors (all robots detected by robot i) as $\mathcal{N}_{s,i}^k = \{l | (l, i) \in \mathcal{E}_s^k, \forall l \neq i, l \in \mathcal{V}\}$. We assume that for each robot, the communication radius is larger than the sensing radii of all robots. Then when robot i detects robot j , robot i can receive the information from robot j .

C. Joint Localization and Target Tracking

In this section, we briefly introduce our previous work about JLATT [5]. In this work, each robot can estimate the pose of itself (localization) and the state of a target (tracking) using only its own information and the information from its one-hop communicating neighbors while preserving estimation consistency.

1) *Robot Propagation*: The estimate of robot i 's state and its corresponding covariance are propagated at time $k-1$ as

$$\begin{aligned} \bar{\mathbf{x}}_i^k &= f_i(\hat{\mathbf{x}}_i^{k-1}, \mathbf{u}_i^{k-1}, 0), \\ \bar{\mathbf{p}}_i^k &= \Phi_i^{k-1} \hat{\mathbf{p}}_i^{k-1} (\Phi_i^{k-1})^\top + \bar{\mathbf{Q}}_i^{k-1} \\ &= \rho_{Rp}(\hat{\mathbf{p}}_i^{k-1}, \hat{\mathbf{x}}_i^{k-1}, \mathbf{Q}_i^{k-1}), \end{aligned} \quad (4)$$

where $\Phi_i^{k-1} = \frac{\partial f_i}{\partial \mathbf{x}_i}(\hat{\mathbf{x}}_i^{k-1}, \mathbf{u}_i^{k-1}, 0)$, $\mathbf{G}_i^{k-1} = \frac{\partial f_i}{\partial \mathbf{w}_i}(\hat{\mathbf{x}}_i^{k-1}, \mathbf{u}_i^{k-1}, 0)$, and $\bar{\mathbf{Q}}_i^{k-1} = \mathbf{G}_i^{k-1} \mathbf{Q}_i^{k-1} (\mathbf{G}_i^{k-1})^\top$.

2) *Target Propagation*: Robot i propagates its estimate of the target's state and its corresponding covariance at time $k-1$ as

$$\begin{aligned} \bar{\mathbf{x}}_{T_i}^k &= g(\hat{\mathbf{x}}_{T_i}^{k-1}, \mathbf{u}_T^{k-1}, 0), \\ \bar{\mathbf{p}}_{T_i}^k &= \Phi_{T_i}^k \hat{\mathbf{p}}_{T_i}^{k-1} (\Phi_{T_i}^{k-1})^\top + \bar{\mathbf{Q}}_{T_i}^{k-1} \\ &= \rho_{Tp}(\hat{\mathbf{p}}_{T_i}^{k-1}, \hat{\mathbf{x}}_{T_i}^{k-1}, \mathbf{Q}_{T_i}^{k-1}), \end{aligned} \quad (5)$$

where $\Phi_{T_i}^{k-1} = \frac{\partial g}{\partial \mathbf{x}_T}(\hat{\mathbf{x}}_{T_i}^{k-1}, \mathbf{u}_T^{k-1}, 0)$, $\mathbf{G}_{T_i}^{k-1} = \frac{\partial g}{\partial \mathbf{w}_T}(\hat{\mathbf{x}}_{T_i}^{k-1}, \mathbf{u}_T^{k-1}, 0)$, and $\bar{\mathbf{Q}}_{T_i}^{k-1} = \mathbf{G}_{T_i}^{k-1} \mathbf{Q}_{T_i}^{k-1} (\mathbf{G}_{T_i}^{k-1})^\top$.

3) *Robot State Update*: To update the estimate of robot i 's state, we first obtain the correction pairs $(\mathbf{s}_{R_{il}}^k, \mathbf{y}_{R_{il}}^k)$ and $(\mathbf{s}_{RiT}^k, \mathbf{y}_{RiT}^k)$ using robot i 's robot-robot measurements $\mathbf{z}_{R_{il}}^k$, $l \in \mathcal{N}_{s,i}^k$, and robot-target measurement $\mathbf{z}_{RiT}^k$, respectively. These correction pairs are calculated as

$$\mathbf{s}_{R_{il}}^k = (\mathbf{H}_{il}^k)^\top (\bar{\mathbf{R}}_{il}^k)^{-1} \mathbf{H}_{il}^k, \quad (6a)$$

$$\mathbf{y}_{R_{il}}^k = (\mathbf{H}_{il}^k)^\top (\bar{\mathbf{R}}_{il}^k)^{-1} (\bar{\mathbf{z}}_{R_{il}}^k + \mathbf{H}_{il}^k \bar{\mathbf{x}}_i^k), \quad (6b)$$

where $\mathbf{H}_{il}^k = \frac{\partial h_{il}}{\partial \mathbf{x}_i}(\bar{\mathbf{x}}_i^k, \bar{\mathbf{x}}_l^k)$, $\bar{\mathbf{R}}_{il}^k = \mathbf{R}_{il}^k + \tilde{\mathbf{H}}_{il}^k \bar{\mathbf{p}}_l^k (\tilde{\mathbf{H}}_{il}^k)^\top$, $\tilde{\mathbf{H}}_{il}^k = \frac{\partial h_{il}}{\partial \mathbf{x}_l}(\bar{\mathbf{x}}_i^k, \bar{\mathbf{x}}_l^k)$, and $\bar{\mathbf{z}}_{R_{il}}^k = \mathbf{z}_{R_{il}}^k - h_{il}(\bar{\mathbf{x}}_i^k, \bar{\mathbf{x}}_l^k)$, and

$$\mathbf{s}_{RiT}^k = (\mathbf{H}_{iT}^k)^\top (\bar{\mathbf{R}}_{iT}^k)^{-1} \mathbf{H}_{iT}^k, \quad (7a)$$

$$\mathbf{y}_{RiT}^k = (\mathbf{H}_{iT}^k)^\top (\bar{\mathbf{R}}_{iT}^k)^{-1} (\bar{\mathbf{z}}_{RiT}^k + \mathbf{H}_{iT}^k \bar{\mathbf{x}}_i^k), \quad (7b)$$

where $\mathbf{H}_{iT}^k = \frac{\partial h_{iT}}{\partial \mathbf{x}_i}(\bar{\mathbf{x}}_i^k, \bar{\mathbf{x}}_T^k)$, $\bar{\mathbf{R}}_{iT}^k = \mathbf{R}_{iT}^k + \tilde{\mathbf{H}}_{iT}^k \bar{\mathbf{p}}_T^k (\tilde{\mathbf{H}}_{iT}^k)^\top$, $\tilde{\mathbf{H}}_{iT}^k = \frac{\partial h_{iT}}{\partial \mathbf{x}_T}(\bar{\mathbf{x}}_i^k, \bar{\mathbf{x}}_T^k)$, and $\bar{\mathbf{z}}_{RiT}^k = \mathbf{z}_{RiT}^k - h_{iT}(\bar{\mathbf{x}}_i^k, \bar{\mathbf{x}}_T^k)$.

Then, we apply the CI algorithm [17] on these correction pairs to compute a consistent estimate $\check{\mathbf{x}}_i$ and its corresponding covariance $\check{\mathbf{p}}_i$ as

$$\check{\mathbf{p}}_i^k = \left(\sum_{l \in \mathcal{N}_{s,i}^k} \eta_{il}^k \mathbf{s}_{R_{il}}^k + \eta_{iT}^k \mathbf{s}_{RiT}^k \right)^{-1}, \quad (8a)$$

$$\check{\mathbf{x}}_i^k = \check{\mathbf{p}}_i^k \left(\sum_{l \in \mathcal{N}_{s,i}^k} \eta_{il}^k \mathbf{y}_{R_{il}}^k + \eta_{iT}^k \mathbf{y}_{RiT}^k \right), \quad (8b)$$

where $\eta_{il}^k \in [0, 1]$, and $\eta_{iT}^k = 0$ if robot i cannot detect the target and otherwise $\eta_{iT}^k \in [0, 1]$ subject to $\sum_{l \in \mathcal{N}_{s,i}^k} \eta_{il}^k + \eta_{iT}^k = 1$. These η_{il} and η_{iT} can be calculated to minimize the trace of $\check{\mathbf{p}}_i^k$.

After we obtain the estimation pair $(\check{\mathbf{p}}_i^k, \check{\mathbf{x}}_i^k)$ using relative measurements and also the prior estimation pair $(\bar{\mathbf{p}}_i^k, \bar{\mathbf{x}}_i^k)$ from the robot propagation step, we can use the CI algorithm to fuse these two estimation pairs to obtain the posterior estimation pair $(\hat{\mathbf{p}}_i^k, \hat{\mathbf{x}}_i^k)$ as

$$\hat{\mathbf{p}}_i^k = \left(\zeta_{i1}^k (\check{\mathbf{p}}_i^k)^{-1} + \zeta_{i2}^k (\bar{\mathbf{p}}_i^k)^{-1} \right)^{-1}, \quad (9a)$$

$$\hat{\mathbf{x}}_i^k = \hat{\mathbf{p}}_i^k \left(\zeta_{i1}^k (\check{\mathbf{p}}_i^k)^{-1} \check{\mathbf{x}}_i^k + \zeta_{i2}^k (\bar{\mathbf{p}}_i^k)^{-1} \bar{\mathbf{x}}_i^k \right), \quad (9b)$$

where ζ_{i1}^k and $\zeta_{i2}^k \in [0, 1]$, subject to $\zeta_{i1}^k + \zeta_{i2}^k = 1$, can be calculated to minimize the trace of $\hat{\mathbf{p}}_i^k$.

4) *Target State Update*: Similarly, for the target state estimation, each robot i first collects the target correction pairs $(\tilde{\mathbf{s}}_{R_j T}^k, \tilde{\mathbf{y}}_{R_j T}^k)$ from available robot-target measurements calculated by its inclusive communicating neighbors j , $j \in \mathcal{I}_{c,i}^k$,

$$\tilde{\mathbf{s}}_{R_j T}^k = (\tilde{\mathbf{H}}_{jT}^k)^\top (\tilde{\mathbf{R}}_{jT}^k)^{-1} \tilde{\mathbf{H}}_{jT}^k, \quad (10a)$$

$$\tilde{\mathbf{y}}_{R_j T}^k = (\tilde{\mathbf{H}}_{jT}^k)^\top (\tilde{\mathbf{R}}_{jT}^k)^{-1} (\tilde{\mathbf{z}}_{R_j T}^k + \tilde{\mathbf{H}}_{jT}^k \bar{\mathbf{x}}_{T_j}^k), \quad (10b)$$

where $\tilde{\mathbf{R}}_{jT}^k = \mathbf{R}_{jT}^k + \mathbf{H}_{jT}^k \bar{\mathbf{p}}_j^k (\mathbf{H}_{jT}^k)^\top$.

Then, at time k , by combining all available target correction pairs, we obtain an intermediate estimation pair as

$$\check{\mathbf{p}}_{T_i}^k = \left(\sum_{j \in \mathcal{I}_{c,i}^k} \tilde{\eta}_j^k \tilde{\mathbf{s}}_{R_j T}^k \right)^{-1}, \quad (11a)$$

$$\check{\mathbf{x}}_{T_i}^k = \check{\mathbf{p}}_{T_i}^k \left(\sum_{j \in \mathcal{I}_{c,i}^k} \tilde{\eta}_j^k \tilde{\mathbf{y}}_{R_j T}^k \right), \quad (11b)$$

where $\tilde{\eta}_j^k = 0$ if robot j cannot directly detect the target, and otherwise $\tilde{\eta}_j^k \in [0, 1]$ subject to $\sum_{j \in \mathcal{I}_{c,i}^k} \tilde{\eta}_j^k = 1$.

Then we fuse all available prior estimation pairs $(\bar{\mathbf{p}}_{T_j}^k, \bar{\mathbf{x}}_{T_j}^k)$, $\forall j \in \mathcal{I}_{c,i}^k$ with the CI algorithm as

$$\check{\mathbf{p}}_{T_i}^k = \left(\sum_{j \in \mathcal{I}_{c,i}^k} \pi_j^k (\bar{\mathbf{p}}_{T_j}^k)^{-1} \right)^{-1}, \quad (12a)$$

$$\check{\mathbf{x}}_{T_i}^k = \check{\mathbf{p}}_{T_i}^k \left(\sum_{j \in \mathcal{I}_{c,i}^k} \pi_j^k (\bar{\mathbf{p}}_{T_j}^k)^{-1} \bar{\mathbf{x}}_{T_j}^k \right), \quad (12b)$$

where $\pi_j^k \in [0, 1]$, subject to $\sum_{j \in \mathcal{I}_{c,i}^k} \pi_j^k = 1$, can be calculated to minimize the trace of $\check{\mathbf{p}}_{T_i}^k$.

Eventually, after obtaining the pairs $(\check{\mathbf{p}}_{T_i}^k, \check{\mathbf{x}}_{T_i}^k)$ and $(\bar{\mathbf{p}}_{T_i}^k, \bar{\mathbf{x}}_{T_i}^k)$, we calculate the posterior estimation pair $(\hat{\mathbf{p}}_{T_i}^k, \hat{\mathbf{x}}_{T_i}^k)$ of the target as

$$\hat{\mathbf{p}}_{T_i}^k = \left(\zeta_{iT_1}^k (\check{\mathbf{p}}_{T_i}^k)^{-1} + \zeta_{iT_2}^k (\bar{\mathbf{p}}_{T_i}^k)^{-1} \right)^{-1}, \quad (13a)$$

$$\hat{\mathbf{x}}_{T_i}^k = \hat{\mathbf{p}}_{T_i}^k \left(\zeta_{iT_1}^k (\check{\mathbf{p}}_{T_i}^k)^{-1} \check{\mathbf{x}}_{T_i}^k + \zeta_{iT_2}^k (\bar{\mathbf{p}}_{T_i}^k)^{-1} \bar{\mathbf{x}}_{T_i}^k \right), \quad (13b)$$

where $\zeta_{iT_1}^k$ and $\zeta_{iT_2}^k \in [0, 1]$, subject to $\zeta_{iT_1}^k + \zeta_{iT_2}^k = 1$, can be calculated to minimize the trace of $\hat{\mathbf{p}}_{T_i}^k$.

III. ACTIVE JOINT LOCALIZATION AND TARGET TRACKING

While our previous work [5] achieves fully distributed JLATT, the robots' motions are not actively controlled or planned and they simply move around randomly. As a result, the full potential to improve the localization and target tracking performance is not exploited. In this paper, we aim to actively control the motions of the robots to achieve better localization and target tracking performance than random movements. We use the term AJLATT to emphasise the fact that we try to not only estimate the states of the robots and the target but also design proper control actions to improve the estimation performance. Our goal is to design the control input $\mathbf{u}_i^k$ for each robot i modeled by (1) at each time k in a fully distributed manner by using only its own information and the information from its one-hop communicating neighbors. Next, an optimization approach to AJLATT is presented.

An overview of the proposed AJLATT algorithm is summarized in Table I. In **Step 3**, we implement our optimization-based AJLATT algorithm (14) to solve the AJLATT problem. The goal is to find an optimal control input $\mathbf{u}_i^{k-1}$ for each robot i modeled by (1), to optimize certain objective functions.

There exist some previous works solving a related problem from the optimization perspective. However, they either ignore the robot localization [9]–[13] or are implemented in a centralized manner [14], [15]. In contrast, here the robot localization is explicitly considered and the problem is solved in a fully distributed manner by using each robot's own and its one-hop neighbors' information. There is no center node required for computation, global parameter shared among the entire team, or information transmitted via multiple hops.

Our optimization-based AJLATT algorithm can be summarized as

$$\begin{aligned} \mathbf{u}_i^{k-1} = & \arg \min_{\mathbf{u}_i^{k-1} \in \mathcal{U}_i} \alpha_{R_i} \text{tr}(\bar{\mathbf{p}}_i^{k+t_2}) + \alpha_{T_i} \text{tr}(\bar{\mathbf{p}}_{T_i}^{k+t_2}) \\ & + \beta_i \sum_{j \in \mathcal{N}_{c,i}^{k-1} \cup \{T_i\}} J_{R_{ij}}^{k+t_1}, i = 1, \dots, M \end{aligned}$$

$$\begin{aligned} \text{s.t.} \quad & (\bar{\mathbf{x}}_j^{k+t_1}, \bar{\mathbf{x}}_{T_j}^{k+t_1}, \bar{\mathbf{p}}_i^{k+t_2}, \bar{\mathbf{p}}_{T_i}^{k+t_2}) = \\ & \text{SACP}(\hat{\mathbf{x}}_j^{k-1}, \hat{\mathbf{x}}_{T_j}^{k-1}, \hat{\mathbf{p}}_j^{k-1}, \hat{\mathbf{p}}_{T_j}^{k-1}, \mathbf{u}_j^{k-2}, \mathbf{u}_T^{k-1}), j \in \mathcal{I}_{c,i}^{k-1} \end{aligned} \quad (14)$$

where $\mathcal{U}_i$ is the control space of robot i . α_{R_i} , α_{T_i} , and β_i are three positive constant parameters that are used to adjust the influence of each term on the objective function. $\bar{\mathbf{p}}_i^{k+t_2}$ and $\bar{\mathbf{p}}_{T_i}^{k+t_2}$ are, respectively, the predicted robot and target covariances at time $k + t_2$ representing, respectively, robot i 's predicted self-localization and target tracking uncertainty. $J_{R_{ij}}^{k+t_1}$, $j \in \mathcal{N}_{c,i}^{k-1} \cup \{T_i\}$, is the potential function term at time $k + t_1$, which is used to maintain communication connectivity, avoid collisions, and keep sight of the target. t_1 and t_2 are, respectively, the planning horizons for the potential function term and covariance terms. In general, we have $t_1 < t_2$, since the difference of uncertainty between different control inputs needs more time to show up, but the potential function term that helps robots to quickly avoid

TABLE I: AJLATT by Robot i

Initialization:	
1	Initialize $\hat{\mathbf{x}}_i^0, \hat{\mathbf{p}}_i^0, \hat{\mathbf{x}}_{T_i}^0, \hat{\mathbf{p}}_{T_i}^0$, and set $u_i^{-1} = 0$.
At time $k - 1$	
Information Exchange for AJLATT:	
2.1	Send $\hat{\mathbf{x}}_i^{k-1}, \hat{\mathbf{p}}_i^{k-1}, \mathbf{u}_i^{k-2}, f_i(\cdot), \hat{\mathbf{x}}_{T_i}^{k-1}, \hat{\mathbf{p}}_{T_i}^{k-1}$ to robot j , $i \in \mathcal{N}_{c,j}^{k-1}$.
2.2	Receive $\hat{\mathbf{x}}_j^{k-1}, \hat{\mathbf{p}}_j^{k-1}, \mathbf{u}_j^{k-2}, f_j(\cdot), \hat{\mathbf{x}}_{T_j}^{k-1}, \hat{\mathbf{p}}_{T_j}^{k-1}$ from robot $j, j \in \mathcal{N}_{c,i}^{k-1}$.
AJLATT Motion Planning:	
3	Calculate $\mathbf{u}_i^{k-1}$ using the optimization-based AJLATT algorithm (14).
Propagation:	
4	Propagate robot i 's estimate of its own state $\hat{\mathbf{x}}_i^{k-1}$ and covariance $\hat{\mathbf{p}}_i^{k-1}$ with the obtained $\mathbf{u}_i^{k-1}$ using (4) and the target's state $\hat{\mathbf{x}}_{T_i}^{k-1}$ and covariance $\hat{\mathbf{p}}_{T_i}^{k-1}$ using (5) to obtain $\bar{\mathbf{x}}_i^k, \bar{\mathbf{p}}_i^k, \bar{\mathbf{x}}_{T_i}^k$ and $\bar{\mathbf{p}}_{T_i}^k$.
At time k	
Update:	
5.1	Obtain the robot-robot measurements $\mathbf{z}_{R_{il}}^k, l \in \mathcal{N}_{s,i}^k$, and robot-target measurement $\mathbf{z}_{R_iT}^k$ (if the target is detected by robot i) and generate the corresponding correction pairs $(\mathbf{s}_{R_{il}}^k, \mathbf{y}_{R_{il}}^k), (\mathbf{s}_{R_iT}^k, \mathbf{y}_{R_iT}^k), (\tilde{\mathbf{s}}_{R_iT}^k, \tilde{\mathbf{y}}_{R_iT}^k)$ using (6), (7), (10).
5.2	Send $(\tilde{\mathbf{s}}_{R_iT}^k, \tilde{\mathbf{y}}_{R_iT}^k), (\bar{\mathbf{p}}_{T_i}^k, \bar{\mathbf{x}}_{T_i}^k)$ to robot $j, i \in \mathcal{N}_{c,j}^k$.
5.3	Receive $(\tilde{\mathbf{s}}_{R_jT}^k, \tilde{\mathbf{y}}_{R_jT}^k), (\bar{\mathbf{p}}_{T_j}^k, \bar{\mathbf{x}}_{T_j}^k)$ from robot $j, j \in \mathcal{N}_{c,i}^k$.
5.4	Calculate the posterior robot estimate pair $(\hat{\mathbf{p}}_i^k, \hat{\mathbf{x}}_i^k)$ using (8), (9), and the posterior target estimate pair $(\hat{\mathbf{p}}_{T_i}^k, \hat{\mathbf{x}}_{T_i}^k)$ using (11), (12), (13).

collision and frequently keep communication connectivity needs to be computed in a shorter time period. SACP is a function that is used to predict the estimate of the states $\bar{\mathbf{x}}_j^{k+t_1}, \bar{\mathbf{x}}_{T_j}^{k+t_1}$ and covariances $\bar{\mathbf{p}}_i^{k+t_2}, \bar{\mathbf{p}}_{T_i}^{k+t_2}$ at time $k + t_1$ and $k + t_2$, respectively, and will be shown in detail later.

Let $\mathbf{r}_i = [x_i, y_i]^T$ be the position part of the state $\hat{\mathbf{x}}_i$ in (1). For each robot i , the potential function $J_{R_{ij}}$ is defined as

$$J_{R_{ij}} = \begin{cases} \infty, & \|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\| \leq \underline{d}_i \\ -10\log(\frac{\|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\| - \underline{d}_i}{a_i}), & \underline{d}_i < \|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\| \leq \underline{d}_i + a_i \\ 0, & \underline{d}_i + a_i < \|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\| \leq \bar{d}_i - a_i \\ 10(\|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\| - (\bar{d}_i - a_i))^2, & \|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\| > \bar{d}_i - a_i \end{cases} \quad (15)$$

where $\bar{\mathbf{r}}_i = [\bar{x}_i, \bar{y}_i]^T$ is the prior estimate of $\mathbf{r}_i$, $\|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\|, j \in \mathcal{N}_{c,i} \cup \{T_i\}$, is the estimated robot-robot or robot-target distance, $\underline{d}_i$ and $\bar{d}_i$ are, respectively, the minimum and maximum acceptable distances that are used to avoid collisions and maintain communication connectivity, and a_i is the length of the non-zero interval. Note that the definition of $J_{R_{ij}}$ accommodates both robot-robot potential and robot-target potential.

According to the definition of the potential function $J_{R_{ij}}$, there are several properties that we would like to point out: 1) $J_{R_{ij}} = 0$, when $\|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\| \in [\underline{d}_i + a_i, \bar{d}_i - a_i]$, which means that only when other robots or the target moves too close to the minimum range or almost moves out of the

maximum acceptable range, this potential function will play a role. Otherwise, only the covariance terms take effect in the objective function in (14). 2) $J_{R_{ij}} \rightarrow \infty$, if $\|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\| \rightarrow \underline{d}_i$ or $\|\bar{\mathbf{r}}_i - \bar{\mathbf{r}}_j\| \rightarrow \infty$. The potential function will immediately give a large penalty (close to infinity) when the robot comes too close to its neighbors or the target to avoid collisions. In contrast, it gives a relatively soft and weak penalty when the robot moves far away from its neighbors or the target so as to maintain communication connectivity or not lose sight of the target, respectively.

The State and Covariance Prediction (SACP) process is first used to predict robot i 's estimates of its own and the target's states, and robot i 's estimates of each neighbor j 's state at time $k + t_1$ so as to calculate the potential function term $J_{R_{ij}}^{k+t_1}$. The process is consecutively used to predict robot i 's covariance $\bar{\mathbf{p}}_i^{k+t_2}$ and the target covariance $\bar{\mathbf{p}}_{T_i}^{k+t_2}$ at time $k + t_2$ which constitute the uncertainty minimization terms in the objective function in (14). The SACP process is shown in Table II.

Remark 1. Since $\mathbf{u}_j^{k-1}$ is the optimization variable for each robot j , robot i propagates its neighbor j 's state estimate with the neighbor's latest available control input $\mathbf{u}_j^{k-2}$ at time $k-2$. Robot i 's own state is propagated with $\mathbf{u}_i^k$. The target's state is propagated with $\mathbf{u}_T^{k-1}$, which is assumed to be known for each robot at time $k-1$.

Remark 2. In Step 2 in Table II, $^*\mathcal{N}_{s,i}^k$ and $^*\mathcal{I}_{c,i}^k$ are two predicted neighbor sets at time k in the prediction process. Since the real movement does not happen, robot i can only utilize the information from its communicating neighbors at time $k-1$. Thus $^*\mathcal{N}_{s,i}^k$ and $^*\mathcal{I}_{c,i}^k$ are the subsets of the known set $\mathcal{I}_{c,i}^{k-1}$, where $^*\mathcal{N}_{s,i}^k = \{i|(l, i) \in (^*\mathcal{E}_s^k \cap \mathcal{E}_c^{k-1}), \forall l \neq i, l \in \mathcal{V}\}$ and $^*\mathcal{I}_{c,i}^k = \{i|(l, i) \in (^*\mathcal{E}_c^k \cap \mathcal{E}_c^{k-1}), l \in \mathcal{V}\}$. Here $^*\mathcal{E}_s^k$ and $^*\mathcal{E}_c^k$ are, respectively, the predicted sensing and communication edge set at time k which are determined by the predicted estimates of the robots' states $\bar{\mathbf{x}}_j^k$ together with, respectively, the robots' sensing fields of view and communication radii. In addition, η_{iT}^k in (8a) is determined by $\bar{\mathbf{x}}_i^k$ and the estimate of the target's state $\bar{\mathbf{x}}_{T_i}^k$ together with the robots' sensing fields of view.

Remark 3. In Step 4 in Table II, each robot i propagates its own state estimate and its estimate of the target's state up to time $k+t_1$. For each of robot i 's communicating neighbors at time $k-1$ ($j \in \mathcal{N}_{c,i}^{k-1}$), robot i also propagates neighbor j 's state estimate up to time $k+t_1$. The potential function term in (14) would result in a larger penalty when robot i loses communication connectivity with its neighbor $j, j \in \mathcal{N}_{c,i}^{k-1}$, at time $k+t_1$. Instead, if the potential function term were given by $\beta_i \sum_{j \in ^*\mathcal{N}_{c,i}^{k+t_1} \cup \{T_i\}} J_{R_{ij}}^{k+t_1}$ in (22), then a smaller value would be obtained due to the removal of robot j in $^*\mathcal{N}_{c,i}^{k+t_1}$.

IV. SIMULATION

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

A. Simulation Setup

Consider the scenario where $M = 6$ robots and a target move on a surface. Here we adopt the widely used unicycle model for both robots and the target in the simulation. The robot pose $\mathbf{x}_i^k$ consists of the position (x_i^k, y_i^k) and the orientation θ_i^k in the global frame. The motion models (1) and (2) can be expressed as

$$\begin{aligned} x_i^k &= x_i^{k-1} + (v_i^{k-1} + w_{v_i}^{k-1})\delta t \cos(\theta_i^{k-1}), \\ y_i^k &= y_i^{k-1} + (v_i^{k-1} + w_{v_i}^{k-1})\delta t \sin(\theta_i^{k-1}), \\ \theta_i^k &= \theta_i^{k-1} + (\omega_i^{k-1} + w_{\omega_i}^{k-1})\delta t, \end{aligned} \quad (16)$$

where $i \in \{1, \dots, M\} \cup \{T\}$, $\delta t = 1$ s is the sampling interval, $\mathbf{u}_i^{k-1} = [v_i^{k-1}, \omega_i^{k-1}]^T$ represents the linear and

angular velocities as the input for robot i , and $\mathbf{w}_i = [w_{v_i}^{k-1}, w_{\omega_i}^{k-1}]^T$ represents process noises for the linear and angular velocities. For robot i , the input $\mathbf{u}_i^{k-1}$ is calculated by our AJLATT algorithm. The target's input is assumed to be constant with $v_T = 0.25$ m/s and $\omega_T = 0$. The target's input $\mathbf{u}_T = [v_T, \omega_T]^T$ is known by every robot i . The corresponding process noise $\mathbf{w}_i$, $i \in \{1, \dots, M\} \cup \{T\}$, is assumed to be white Gaussian, with the standard deviations for $w_{v_i}^{k-1}$ and $w_{\omega_i}^{k-1}$ as, respectively, $\sigma_{v_i}^{k-1} = \frac{\sqrt{2}}{2}\sigma_i^{k-1}$ and $\sigma_{\omega_i}^{k-1} = 2\sqrt{2}\sigma_i^{k-1}$, where σ_i^{k-1} is proportional to the linear velocity as $\sigma_i^{k-1} = 1\%v_i^{k-1}$ for each robot i , $i \in \{1, \dots, M\}$, and $\sigma_T^{k-1} = 3\%v_T^{k-1}$ for the target. Hence $\mathbf{Q}_i^{k-1}$, $i \in \{1, \dots, M\} \cup \{T\}$ defined after (1) and (2) is given as $\mathbf{Q}_i^{k-1} = \begin{bmatrix} (\sigma_{v_i}^{k-1})^2 & 0 \\ 0 & (\sigma_{\omega_i}^{k-1})^2 \end{bmatrix}$.

TABLE II: State and Covariance Prediction (SACP) by Robot i at $k-1$

One-step Propagation and Predicted Update:

- 1 Propagate each inclusive communicating neighbor j 's, $j \in \mathcal{I}_{c,i}^{k-1}$, estimates of its own and the target's states and covariances in one step.

$$\begin{aligned} \bar{\mathbf{x}}_i^k &= f_i(\bar{\mathbf{x}}_i^{k-1}, \mathbf{u}_i^{k-1}, 0), \bar{\mathbf{p}}_i^k = \rho_{Rp}(\bar{\mathbf{p}}_i^{k-1}, \bar{\mathbf{x}}_i^{k-1}, \mathbf{Q}_i^{k-1}), \\ \bar{\mathbf{x}}_j^k &= f_j(\bar{\mathbf{x}}_j^{k-1}, \mathbf{u}_j^{k-1}, 0), \quad j \in \mathcal{N}_{c,i}^{k-1} \\ \bar{\mathbf{p}}_j^k &= \rho_{Rp}(\bar{\mathbf{p}}_j^{k-1}, \bar{\mathbf{x}}_j^{k-1}, \mathbf{Q}_j^{k-1}), \quad j \in \mathcal{N}_{c,i}^{k-1} \\ \bar{\mathbf{x}}_{T_j}^k &= g(\bar{\mathbf{x}}_{T_j}^{k-1}, \mathbf{u}_T^{k-1}, 0), \quad j \in \mathcal{I}_{c,i}^{k-1} \\ \bar{\mathbf{p}}_{T_j}^k &= \rho_{Tp}(\bar{\mathbf{p}}_{T_j}^{k-1}, \bar{\mathbf{x}}_{T_j}^{k-1}, \mathbf{Q}_{T_j}^{k-1}), \quad j \in \mathcal{I}_{c,i}^{k-1} \end{aligned}$$

- 2 Generate one-step predicted correction terms $\mathbf{s}_{R_{ij}}^k$, $l \in \mathcal{N}_{s,i}^k, \mathbf{s}_{R_{iT}}^k, \mathbf{s}_{R_{jT}}^k, j \in \mathcal{I}_{c,i}^k$, using (6a), (7a), (10a).
- 3 Calculate one-step predicted posterior robot covariance $\hat{\mathbf{p}}_i^k$ using (8a), (9a), and one-step predicted posterior target covariance $\hat{\mathbf{p}}_{T_i}^k$ using (11a), (12a), (13a).

Propagation in t_1 and t_2 Planning Horizons:

- 4 Propagate robot i 's estimate of the target's state and corresponding covariance and each inclusive communicating neighbor j 's, $j \in \mathcal{I}_{c,i}^{k-1}$, estimates of its own state and corresponding covariance for the **potential function term** in t_1 planning horizon:
 $t = 0, \dots, t_1 - 1$,

$$\begin{cases} \bar{\mathbf{x}}_{T_i}^{k+t+1} = g(\bar{\mathbf{x}}_{T_i}^{k+t}, \mathbf{u}_T^{k-1}, 0), \\ \bar{\mathbf{p}}_{T_i}^{k+t+1} = \rho_{Tp}(\bar{\mathbf{p}}_{T_i}^{k+t}, \bar{\mathbf{x}}_{T_i}^{k+t}, \mathbf{Q}_{T_i}^{k+t}), \text{ if } t = 0 \\ \bar{\mathbf{p}}_{T_i}^{k+t+1} = \rho_{Tp}(\bar{\mathbf{p}}_{T_i}^{k+t}, \bar{\mathbf{x}}_{T_i}^{k+t}, \mathbf{Q}_{T_i}^{k+t}), \text{ if } t \geq 1 \\ \bar{\mathbf{x}}_i^{k+t+1} = f_i(\bar{\mathbf{x}}_i^{k+t}, \mathbf{u}_i^{k-1}, 0), \\ \bar{\mathbf{p}}_i^{k+t+1} = \rho_{Rp}(\bar{\mathbf{p}}_i^{k+t}, \bar{\mathbf{x}}_i^{k+t}, \mathbf{Q}_i^{k+t}), \text{ if } t = 0 \\ \bar{\mathbf{p}}_i^{k+t+1} = \rho_{Rp}(\bar{\mathbf{p}}_i^{k+t}, \bar{\mathbf{x}}_i^{k+t}, \mathbf{Q}_i^{k+t}), \text{ if } t \geq 1 \\ \bar{\mathbf{x}}_j^{k+t+1} = f_j(\bar{\mathbf{x}}_j^{k+t}, \mathbf{u}_j^{k-1}, 0), j \in \mathcal{N}_{c,i}^{k-1} \end{cases}$$

Compute $J_{R_{ij}}^{k+t_1}$ using $\|\bar{\mathbf{r}}_i^{k+t_1} - \bar{\mathbf{r}}_j^{k+t_1}\|$, $j \in \mathcal{N}_{c,i}^{k-1} \cup \{T_i\}$.

- 5 Propagate robot i 's estimates of its own and the target's states and corresponding covariances to obtain the **covariance terms** in t_2 planning horizon
 $t = t_1, \dots, t_2 - 1$,

$$\begin{aligned} \bar{\mathbf{x}}_i^{k+t+1} &= f_i(\bar{\mathbf{x}}_i^{k+t}, \mathbf{u}_i^{k-1}, 0), \bar{\mathbf{x}}_{T_i}^{k+t+1} = g(\bar{\mathbf{x}}_{T_i}^{k+t}, \mathbf{u}_T^{k-1}, 0), \\ \bar{\mathbf{p}}_i^{k+t+1} &= \rho_{Rp}(\bar{\mathbf{p}}_i^{k+t}, \bar{\mathbf{x}}_i^{k+t}, \mathbf{Q}_i^{k+t}), \\ \bar{\mathbf{p}}_{T_i}^{k+t+1} &= \rho_{Tp}(\bar{\mathbf{p}}_{T_i}^{k+t}, \bar{\mathbf{x}}_{T_i}^{k+t}, \mathbf{Q}_{T_i}^{k+t}). \end{aligned}$$

We assume that each robot has a limited communication range with a radius of $R_i = 30$ m and a limited field of view with $R_{min} = 2$ m and $R_{max} = 15$ m for the range and $\phi = 60^\circ$ for the angle of view. As for the measurement model, we consider an indoor application scenario in this work and assume that these robots do not have access to the absolute position measurement. The relative distance-bearing measurement model is adopted for each robot in our simulation. If robot i detects robot j at time instant k , then the relative measurement can be expressed as $\mathbf{z}_{R_{ij}}^k = \begin{bmatrix} \sqrt{(x_j^k - x_i^k)^2 + (y_j^k - y_i^k)^2} \\ \text{atan2}((y_j^k - y_i^k), (x_j^k - x_i^k)) - \theta_i^k \end{bmatrix} + \mathbf{v}_{R_{ij}}^k$, where $\mathbf{v}_{R_{ij}}^k$ is a zero-mean white Gaussian noise. The standard deviation of the distance noise is set to be 3% of the actual distance, and the standard deviation of the bearing noise equals to 1° . The same measurement model is used for the robot-to-target measurement $\mathbf{z}_{R_{iT}}^k$.

Since the absolute measurement is not available, we assume that each robot initializes its estimated pose $\hat{\mathbf{x}}_i^0$ with its true pose $\mathbf{x}_i^0$, and the initial pose covariance $\hat{\mathbf{p}}_i^0$ is set to $\hat{\mathbf{p}}_i^0 = 10^{-3}\mathbf{I}_3$. The initial estimate of the target's state obtained by each robot i , $\hat{\mathbf{x}}_{T_i}^0$, does not necessarily equal to the true initial state of the target $\mathbf{x}_T^0$. In our simulation, we set $\hat{\mathbf{x}}_T^0 = [10, 10, 0]$ while the true initial target state is $\mathbf{x}_T^0 = [10, 5, 0]$. Since we assume that an accurate initial target state is not available, we initialize the target covariance with relatively large uncertainty as $\hat{\mathbf{p}}_{T_i}^0 = \mathbf{I}_3$.

We compare the performance of the following two motion strategies: 1) *Random Motion (RM)*: In this case, each robot moves with a constant linear velocity of $v_i = 0.5$ m/s. Its angular velocity ω_i is uniformly chosen from an interval of $[-\frac{\pi}{5}, \frac{\pi}{5}]$ rad/s. These robots behave as in our previous work [5] except that there is no moving field boundary for them. By adopting the RM strategy, robots do not tend to pursue the target or maintain communication connectivity with other robots, which eventually results in worse localization and tracking performance than the proposed AJLATT algorithm as shown later. 2) *Optimization-based AJLATT (AJLATT-O)*: In this setting, the robots' linear and angular velocities are calculated by the optimization-based AJLATT algorithm in Section III. The parameters of the optimization objective

function are set to $\alpha_{R_i} = 0.2$, $\alpha_{T_i} = 1$, and $\beta_i = 1$. As for the parameters on robot i , we set the length of the non-zero interval as $a_i = 2$ m and $\bar{d}_i = 2$ m in (15) for both the robot-robot and robot-target potential functions. We also set $\bar{d}_i = 30$ m in (15) for the robot-robot potential function to make the robots maintain the communication with its neighbors, and $\bar{d}_i = 20$ m for the robot-target potential function to help keep sight of the target. The planning horizons are $t_1 = 4$ and $t_2 = 11$ for the SACP process. The control space of each robot i is set as $\mathcal{U} = \{(v_i, \omega_i) | v_i \in [0, 0.5] \text{ m/s}, \omega_i \in [-\frac{\pi}{5}, \frac{\pi}{5}] \text{ rad/s}\}$.

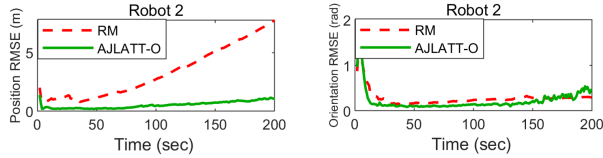


Fig. 1: Position and orientation estimate RMSE for the target on robot 2 (tracking).

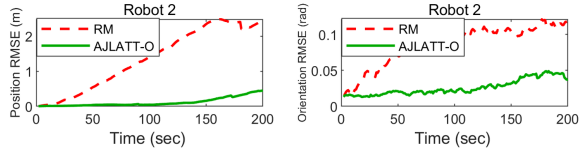


Fig. 2: Position and orientation estimate RMSE for robot 2 (localization).

We run 50 Monte Carlo simulations and use the root mean square error (RMSE) as the metric for accuracy to test the performance of these algorithms. Here we choose robot 2 as an example to show the performance of our proposed approach. Figure 1 shows robot 2's average target position and orientation estimate RMSE for target tracking. Figure 2 shows robot 2's own average position and orientation estimate RMSE for localization.

As shown in Figures 1 and 2, the robots using AJLATT-O algorithm achieve significant better performance than RM. The result confirms our expectation. The RM approach does not actively drive a one robot to observe the target and other robots or maintain the communication connectivity with its neighbors. As a result, the robot obtains fewer measurements and has fewer neighbors to exchange information, thus obtaining less accurate estimates of its own and the target's states compared with the proposed AJLATT-O algorithm. Hence the RM approach has worse performance.

V. CONCLUSION

In this paper, we have proposed an optimization-based algorithm to solve the AJLATT problem in a fully distributed (communication, estimation, planning) manner to drive a team of robots to actively track a target and localize itself so as to achieve better self-localization and target tracking performance. The proposed algorithm tries to find the optimal motion so that optimal robot localization and

target tracking performance can be achieved while collision avoidance and communication maintenance are considered at the same time. Monte-Carlo simulations are performed to illustrate the effectiveness of our approach.

REFERENCES

- [1] G. Huang, M. Kaess, and J. J. Leonard, "Consistent unscented incremental smoothing for multi-robot cooperative target tracking," *Robotics and Autonomous Systems*, vol. 69, pp. 52–67, 2015.
- [2] A. Ahmad, G. D. Tipaldi, P. Lima, and W. Burgard, "Cooperative robot localization and target tracking based on least squares minimization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013, pp. 5696–5701.
- [3] F. M. Mirzaei, A. I. Mourikis, and S. I. Roumeliotis, "On the performance of multi-robot target tracking," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 3482–3489.
- [4] N. Atanasov, R. Tron, V. M. Preciado, and G. J. Pappas, "Joint estimation and localization in sensor networks," in *Proceedings of the IEEE Conference on Decision and Control*, 2014, pp. 6875–6882.
- [5] P. Zhu and W. Ren, "Fully distributed joint localization and target tracking with mobile robot networks," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1519–1532, 2020.
- [6] Y. Cao and W. Ren, "Distributed coordinated tracking with reduced interaction via a variable structure approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 33–48, 2011.
- [7] T. Chung, J. Burdick, and R. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 2416–2422.
- [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] 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.
- [10] K. Zhou and S. I. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 678–695, 2011.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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, 2013.
- [16] 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.
- [17] S. Julier and J. K. Uhlmann, "General decentralized data fusion with covariance intersection," in *Handbook of multisensor data fusion*. CRC Press, 2017, pp. 339–364.