

Resilience in multi-robot target tracking through reconfiguration

Ragesh K. Ramachandran, Nicole Fronda and Gaurav S. Sukhatme

Abstract—We address the problem of maintaining resource availability in a networked multi-robot system performing distributed target tracking. In our model, robots are equipped with sensing and computational resources enabling them to track a target's position using a Distributed Kalman Filter (DKF). We use the trace of each robot's sensor measurement noise covariance matrix as a measure of sensing quality. When a robot's sensing quality deteriorates, the system's communication graph is modified by adding edges such that the robot with deteriorating sensor quality may share information with other robots to improve the team's target tracking ability. This computation is performed centrally and is designed to work without a large change in the number of active communication links. We propose two mixed integer semi-definite programming formulations (an 'agent-centric' strategy and a 'team-centric' strategy) to achieve this goal. We implement both formulations and a greedy strategy in simulation and show that the team-centric strategy outperforms the agent-centric and greedy strategies.

I. INTRODUCTION AND RELATED WORK

Target tracking using a robot team is of interest in many civilian and military applications. Consequently, this has led to a burgeoning interest in studying target tracking problems using robots [1]–[4]. Even though each robot has a limited view of the environment, multiple robots could perform the target tracking task collaboratively by exchanging information among themselves. A necessary condition for this to succeed is the existence of an underlying communication network. Collaborative distributed tracking avoids the need for a central data fusion center which combines the data collected by the robots to estimate the state of the target. A common approach for multi-robot target tracking in a distributed manner is through a distributed Kalman filter [2]. We follow this method in this paper.

We envision a scenario in which a team of robots tracks and estimates the state of a target while running a distributed Kalman filter on board. For simplicity we assume there is a single sensor (e.g. a camera) on each robot that is used to obtain target information. The robot team is monitored by a base station which intervenes in the team's activities only if a robot in the team experiences a deterioration in sensing quality. We use the trace of the sensor's measurement noise covariance matrix as the measure of sensing quality. Our focus is on the strategies to mitigate the effect of a sensor quality deterioration on a robot (and hence on the team's tracking performance) by appropriately modifying

This work was supported in part by the Army Research Laboratory as part of the Distributed and Collaborative Intelligent Systems and Technology (DCIST) Collaborative Research Alliance (CRA). The authors are with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089, USA rageshku|nfronda|gaurav@usc.edu

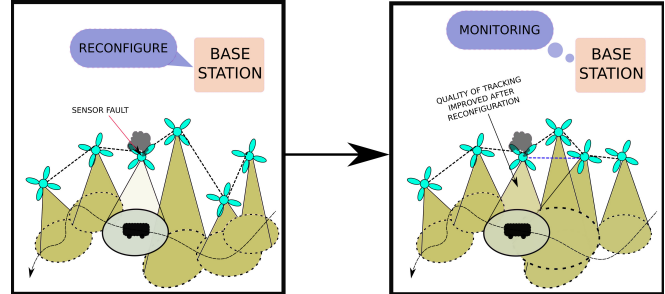


Fig. 1: A setting for resilient target tracking.

the topology of the underlying communication network and target state measurement fusion weights (Figure 1). In a control theoretic sense, the state estimation error of the DKF depends upon the observability of the underlying communication network [5]. Understanding and quantifying the effect of network topology on a network's observability properties [6]–[8] is well studied. These findings form the basis of our proposed strategy to employ communication network topology reconfiguration to mitigate the effect of sensor quality deterioration in target tracking performance. Although, we restrict ourselves to single target tracking in this paper, our technique can be applied to multi-target tracking assuming that robots can identify and discern individual targets.

The framework discussed here extends the abstract framework described in [9] by adapting it to multi-robot target tracking. Similar to [9], we propose a two-stage strategy to alleviate the impact of a robot's sensor quality deterioration on the team's performance. Our framework is in two stages. The first stage updates the team's communication network topology such that the new topology is adjacent to the original. It also computes an optimal set of new measurement fusion weights. Specifically, we optimize a monotonic function of the trace of the covariance matrix of the target state estimation error. The second stage generates a set of coordinates implementing communication network topology while maximizing the coverage of the robots over the target's domain. The framework considered here only accounts for sensor quality deterioration and not for the situation where the sensor fails completely. However, a near-complete sensor failure can be modeled as a sensor with very high measurement noise covariance. We also assume that the sensor measurement noise is always characterized by a zero mean probability distribution. Finally, we assume that robots can estimate their sensor quality. Since there exist techniques for sensor fault detection [10] and degradation estimation [11], [12], this assumption does not impose undue restrictions on the applicability of our method.

Stage one of our method uses mixed integer semi-definite

programs (MISDPs) to formulate and solve the problem of fabricating a communication graph topology and associated information fusion weights that improves the tracking performance of the team. We consider two MISDPs: *agent-centric configuration generation (ACCG)* and *team-centric configuration generation (TCCG)*. The first MISDP (ACCG) maximizes the trace of the inverse of target state estimation error covariance matrix (TSEECM) associated with the robot that experienced the sensor quality degradation. The second one (team-centric) minimizes the average of the trace of TSEECM over all the robots. Our procedure in stage two solves an optimization problem to compute the coordinates of the robots that maximize the coverage of the robots over the target's configuration space under communication constraints dictated by the graph topology computed in stage one. Although, resilience in multi-robot systems is well studied [13], the idea of resilience through reconfiguration during task execution is recent. This paper introduces the idea of resilience by reconfiguration into multi-robot target tracking.

II. NOTATION AND PRELIMINARIES

For any positive integer $z \in \mathbb{Z}^+$, $[z]$ denotes the set $\{1, 2, \dots, z\}$. $\|\cdot\|$ denotes the standard Euclidean 2-norm and the induced 2-norm for vectors and matrices. $\|\mathbf{M}\|_F$ is the Frobenius norm of the matrix $\mathbf{M} \in \mathbb{R}^{m_1 \times m_2}$. $\text{Tr}(\mathbf{M})$ is the trace of matrix \mathbf{M} . $\mathbf{1}^{m_1}$ and $\mathbf{0}^{m_1}$ are the vector of ones and zeros with appropriate dimensions. We drop the superscripts whenever the dimensions of the vectors or matrices are clear from the context. We use $\bar{\mathbf{0}}_i^n$ to denote a vector of zeros with one as its i^{th} entry. For any vector \mathbf{v} , $\text{Diag}(\mathbf{v})$ is the matrix with \mathbf{v} along its diagonal. Conversely, $\text{diag}(\mathbf{M})$ yields the vector containing the diagonal elements of the matrix \mathbf{M} . \mathbf{M}^\top or $(\mathbf{M})^\top$ is the transpose of \mathbf{M} . \mathcal{S}_+^m denotes the space of $m \times m$ symmetric positive semi-definite matrices. A weighted undirected graph with non negative edge weights \mathcal{G} is defined using the triplet $(\mathcal{V}, \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}, \mathbf{A} \in \mathbb{R}_{\geq 0}^{|\mathcal{V}| \times |\mathcal{V}|})$, where \mathbf{A} is the weighted adjacency matrix of the graph. $\bar{\mathcal{E}} = (\mathcal{V} \times \mathcal{V}) \setminus \mathcal{E}$ is the edge complement of \mathcal{G} . A matrix \mathbf{M} is doubly stochastic if its rows and columns sum to unity [14]. Mathematically, $\mathbf{M}\mathbf{1}^n = \mathbf{1}^n$ and $\mathbf{1}^\top \mathbf{M} = \mathbf{1}^\top$. One is an eigenvalue of a doubly stochastic matrix and all other eigenvalues of a doubly stochastic matrix have a magnitude less than one ([15], Lemma 2.9).

III. PROBLEM FORMULATION

We consider a team of n robots whose labels belong to $\{1, 2, \dots, n\}$. The team is tasked with tracking a moving target of interest for a time period of T epochs. It is assumed that the moving target is confined to a compact Euclidean space \mathcal{D} . We refer to the robot team that tracks this moving target as the *tracker team* and the robots as *trackers*. The robot with label i is indicated as ρ^i . Let X^i denote the triplet position vector $[x_i, y_i, z_i] \in \mathbb{R}^3$ of ρ^i , then the set $\{X_{[n]}\}$ contains the positions of all trackers. Also, X_{target} represents the position vector of the target. We assume that the trackers are equipped with localization capabilities which enable them to localize with reasonable accuracy in the environment.

We consider the scenario where the tracker team performs distributed target tracking, thus trackers need to communicate among themselves during the tracking task.

Let the dynamic undirected graph $\mathcal{G}[k] = (\mathcal{V}, \mathcal{E}[k], \mathbf{A}_u[k])$ model the communication network of the tracker team at the k^{th} time step ($k \in [T]$). Note that we use time step, time and epoch interchangeably in this paper. The node set \mathcal{V} is isomorphic to the tracker team label set $[n]$. An edge (i, j) is included in the edge set $\mathcal{E}[k]$ if ρ^i communicates with ρ^j at time k . We denote the communication range of trackers as $d_{mc} > 0$. The neighbor set of node i in $\mathcal{G}[k]$ is defined as $\mathcal{N}_{(i)}[k] = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}[k]\}$. The interaction between nodes in a graph can also be represented using an unweighted adjacency matrix. The adjacency matrix of $\mathcal{G}[k]$ is denoted by $\mathbf{A}_u[k]$.

A. Distributed Kalman filter for tracking

We describe the distributed Kalman filter computation employed by the robots to collectively track a target maneuvering in an environment according to some known dynamics. The main advantage of employing a distributed (as opposed to a centralized) Kalman filter is that it eliminates the need for a central data fusion center [2], [5], [16], [17]. We follow the formulation in [5], [18]. The target dynamics are described by the standard linear state space equation

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{u}_k + \mathbf{w}_k, \quad (1)$$

where $\mathbf{x}_k \in \mathbb{R}^{s_a}$ and $\mathbf{u}_k \in \mathbb{R}^{u_a}$ are the state and the input vectors of the target respectively. $\mathbf{F}_k \in \mathbb{R}^{s_a \times s_a}$ and $\mathbf{G}_k \in \mathbb{R}^{s_a \times u_a}$ are the state transition matrix and input matrix of appropriate dimensions respectively. $\mathbf{w}_k \in \mathbb{R}^{s_a}$ is the zero mean normally distributed random vector with the covariance matrix $\mathbf{Q}_k \in \mathbb{R}^{s_a \times s_a}$ ($\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$). Each tracker can obtain measurements about the state of the target. The sensing region of ρ^i is a disc of radius $d_{sen}^i > 0$ centered around the robot. The measurement model of ρ^i in its sensing region is $\mathbf{z}_k^i = \mathbf{H}_k^i \mathbf{x}_k + \mathbf{v}_k^i$ where $\mathbf{z}_k^i \in \mathbb{R}^{m_i}$ and $\mathbf{H}_k^i \in \mathbb{R}^{s_a \times m_i}$ are the measurement vector and output matrix of ρ^i , respectively. Also, $\mathbf{v}_k^i \in \mathbb{R}^{m_i}$ is a zero mean Gaussian sensing noise vector with a covariance matrix $\mathbf{R}_k^i \in \mathbb{R}^{m_i \times m_i}$ modeling the sensor noise characteristics of ρ^i .

The distributed Kalman filter algorithm consists primarily of two steps: 1) *individual update* and 2) *consensus update*. In the individual update step, the robots that receive a measurement estimate the state of the tracked agent using the standard Kalman filter equations. Mathematically, if ρ^i received a measurement \mathbf{z}_k^i then it makes an estimate about the state of the tracked target using the following equations.

Prediction:

$$\hat{\mathbf{x}}_{k|k-1}^i = \mathbf{F}_k \hat{\mathbf{x}}_{k-1}^i + \mathbf{G}_k \mathbf{u}_{k-1} \quad (2)$$

$$\mathbf{P}_{k|k-1}^i = \mathbf{F}_k \mathbf{P}_{k-1}^i (\mathbf{F}_k)^\top + \mathbf{Q}_k \quad (3)$$

Local innovation or measurement update:

$$\mathbf{K}_k^i = \mathbf{P}_{k|k-1}^i (\mathbf{H}_k^i)^\top \left(\mathbf{H}_k^i \mathbf{P}_{k|k-1}^i (\mathbf{H}_k^i)^\top + \mathbf{R}_k^i \right)^{-1} \quad (4)$$

$$\hat{\mathbf{x}}_{k,0}^i = \hat{\mathbf{x}}_{k|k-1}^i + \mathbf{K}_k^i (\mathbf{z}_k^i - \mathbf{H}_k^i \hat{\mathbf{x}}_{k|k-1}^i) \quad (5)$$

$$\mathbf{P}_{k,0}^i = \mathbf{P}_{k|k-1}^i - \mathbf{K}_k^i \mathbf{H}_k^i \mathbf{P}_{k|k-1}^i \quad (6)$$

The local target state estimate $\hat{\mathbf{x}}_{k,0}^i$ of ρ^i is fused with the local target state estimate of other trackers in the tracker team in a distributed fashion via a consensus protocol. This distributed information fusion is performed in the consensus step of the distributed Kalman filter. Each tracker in the tracker team computes the information vector $\mathbf{q}_k^i(0) = (\mathbf{P}_{k,0}^i)^{-1} \hat{\mathbf{x}}_{k,0}^i$ and the associated information matrix $\mathbf{\Omega}_k^i(0) = (\mathbf{P}_{k,0}^i)^{-1}$ prior to the consensus step. Once these quantities are computed, each robot in the tracker team exchanges information with its neighbors. Ultimately this results in consensus on a refined state estimate of the tracked target. The robots exchange information as follows:

$$\mathbf{\Omega}_k^i(l+1) = \sum_{j \in \mathcal{N}_{(i)}[k] \cup i} [\bar{\mathbf{A}}[k]]_{i,j} \mathbf{\Omega}_k^j(l) \quad (7)$$

$$\mathbf{q}_k^i(l+1) = \sum_{j \in \mathcal{N}_{(i)}[k] \cup i} [\bar{\mathbf{A}}[k]]_{i,j} \mathbf{q}_k^j(l), \quad (8)$$

where $[\bar{\mathbf{A}}[k]]_{i,j}$ is the (i,j) entry of a doubly stochastic matrix $\bar{\mathbf{A}}[k]$ with the same structure as the unweighted adjacency matrix $(\mathbf{A}_u[k])$ of $\mathcal{G}[k]$ except for the diagonal elements. Specifically, $\bar{\mathbf{A}}[k]$ is non-zero along its diagonal and its off-diagonal elements are non-zero if and only if the corresponding elements of $\mathbf{A}_u[k]$ are unity. In theory, information vectors and matrices of the robots converge to a common quantity only when l tends to infinity. However, it is known that this consensus protocol has an exponential rate of convergence [15]. As a result, a reasonable level of consensus on the information vector and matrix can be achieved by propagating Equation 8 and Equation 7 for a sufficient number of consensus steps L . We note that, the consensus update is assumed to happen at a much faster time scale compared to Equation 1. Since the L steps consensus process results in a fusion of all the measurements obtained by the team, the posteriori estimate of the target's state vector $\hat{\mathbf{x}}_k^i$ and the posteriori estimation error covariance matrix \mathbf{P}_k^i can be determined using $\hat{\mathbf{x}}_k^i = (\mathbf{\Omega}_k^i(L))^{-1} \mathbf{q}_k^i(L)$ and $\mathbf{P}_k^i = (\mathbf{\Omega}_k^i(L))^{-1}$ respectively.

B. Tracking under Sensor Quality Deterioration

As mentioned in Section I, we consider the problem of mitigating the effect of sensor quality deterioration on target tracking performance through appropriate reconfiguration of the tracker team. Next we give a precise definition of tracker team reconfiguration and sensor quality deterioration.

We term the tuple $(\mathcal{G}[k], \bar{\mathbf{A}}[k])$ as the *configuration* of the tracker team at the k^{th} time step and denote it by $\mathcal{C}[k]$. $\bar{\mathbf{A}}[k]$ is a doubly stochastic matrix whose elements are used to perform the information fusion computations outlined in Equation 8 and Equation 7 for the consensus step. During tracking operation for a time period T , let n_f detrimental events occur independently to random trackers in the tracker team. We assume that each event results in some sensor quality deterioration. At time k , we say that ρ^i 's sensor quality is deteriorated if the trace of the measurement noise covariance matrix associated with the sensor \mathbf{R}_k^i has increased with respect to $Tr(\mathbf{R}_{k-1}^i)$. In other words, if

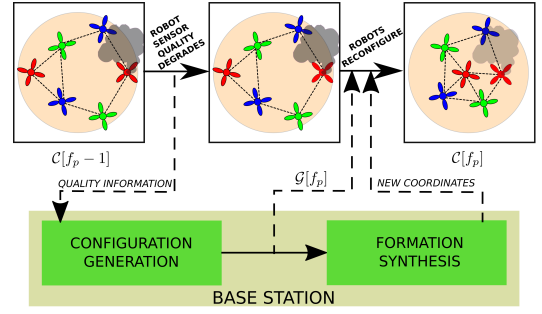


Fig. 2: Basic outline of our approach. When a robot experiences sensor quality degradation, *configuration generation* selects edges to modify the communication graph. Then, *formation synthesis* assigns robots to physical locations that support the desired graph topology. $Tr(\mathbf{R}_k^i) > Tr(\mathbf{R}_{k-1}^i)$, then ρ^i sensor quality deteriorated at time k . Recall that, we assume the sensor is unbiased even after its quality deteriorates. Consider a sequence set $\mathcal{F} = [f_1, f_2, \dots, f_p, \dots, f_{n_f}]$, where f_p indicate the time step when the p^{th} sensor fault occurred. We specify that $\mathcal{C}[f_p - 1]$ is the configuration of the tracker team before the p^{th} detrimental event occurred. We now formally define the problems studied in this paper. The first problem (Problem 1) deals with reconfiguration of the tracker team such that target tracking performance is optimal in some reasonable sense. The second problem addresses the issue of realizing the graph topology in 3D space while maximizing the tracker team's coverage over the \mathcal{D} .

Problem 1. Configuration generation or reconfiguration: Given that ρ^i experienced sensor quality deterioration at some time f_p , $\mathbf{R}_{f_p}^i$ is the sensor noise covariance matrix immediately after the deterioration event and $\mathcal{C}[f_p - 1]$ is the tracker configuration prior to the event determine a new configuration $\mathcal{C}[f_p]$ such that,

- 1) $\mathcal{G}[f_p]$ is a connected graph,
- 2) $\|\mathbf{A}_u[f_p] - \mathbf{A}_u[f_p - 1]\|_F \leq 2 \times n_e$, where $n_e \in \mathbb{Z}^+$ is the number of edges that may be modified in $\mathcal{G}[f_p - 1]$ to obtain $\mathcal{G}[f_p]$ and
- 3) tracking performance is optimized.

Problem 2. Formation synthesis: Given a tracker team configuration $\mathcal{C}[f_p]$, generate coordinates that best realize the given configuration and maximize the tracker team's coverage over \mathcal{D} , subject to constraints. We defer the details of this problem to Section IV-B.

Graph connectivity is an essential requirement for any distributed computation over a network and thus is enforced in Problem 1 [15]. The second condition enables the user to control the communication load on the generated configuration by tuning the parameter n_e . Finally, the third condition ensures good tracking performance.

IV. METHODOLOGY

We now describe our strategies for solving Problem 1 and Problem 2. A base station monitors the activities of the tracker team. Only when sensor deterioration occurs, the base station directs the tracker team to reconfigure to a new formation which it computes using the available

information. Following the framework in our previous work [9], the base station uses a two-step procedure to compute a new configuration and generate a set of robot coordinates which realize the computed configuration. As in [9], we refer to these steps as *configuration generation* and *formation synthesis*. The base station decision making process is shown in Figure 2. The configuration generation and formation synthesis steps are solutions to Problem 1 and Problem 2.

A. Configuration Generation

We propose two strategies for tackling the configuration generation problem. As mentioned earlier, we allude to these strategies as *agent-centric configuration generation (ACCG)* and *team-centric configuration generation (TCCG)*. Both these strategies amount to solving a mixed integer semi-definite program. Network topology design problems are often addressed by posing them as MISDPs [19], [20]. In the agent-centric configuration method, the idea is to find a new configuration such that the trace of the information matrix (inverse of TSEECM) associated with the deteriorated tracker in the tracker team is maximized. The team-centric method minimizes the average of the trace of TSEECMs associated with all the trackers. Since the optimization over the covariance matrices resulting from $L > 1$ consensus steps is a very hard problem to solve, we use the TSEECM obtained after one step consensus to formulate and solve the MISDP for both strategies. Before we describe the details of the MISDP formulations, we state a theorem (proof omitted here for brevity) which serves as the basis for the connectivity inequality constraint in MISDPs.

Theorem 1. *If a graph containing self loops at every node is equipped with a weighted adjacency matrix \mathbf{A} which is doubly stochastic then any graph isomorphic to this graph with or without self loops is connected if and only if*

$$\frac{1}{n}\mathbf{1}\mathbf{1}^\top + \mathbf{I} \succ \mathbf{A} \quad (9)$$

1) **ACCG:** The following MISDP models our agent-centric configuration generation approach:

$$\begin{aligned} & \underset{\substack{\mathbf{A} \in \mathcal{S}_+^n, \mu \in \mathbb{R}_{>0}, \\ \Pi \in \{0,1\}^{n \times n}}}{\text{maximize}} & (\bar{\mathbf{0}}_i^n)^\top \mathbf{A} \begin{bmatrix} \text{Tr}(\Omega_{f_p}^1(0)) \\ \text{Tr}(\Omega_{f_p}^2(0)) \\ \vdots \\ \text{Tr}(\Omega_{f_p}^n(0)) \end{bmatrix} \end{aligned} \quad (10)$$

$$\text{subject to } \mathbf{A} \cdot \mathbf{1}^n = \mathbf{1}^n \quad (11)$$

$$\frac{1}{n}\mathbf{1}\mathbf{1}^\top + (1 - \mu)\mathbf{I} \succeq \mathbf{A}, \mu \ll 1 \quad (12)$$

$$\text{diag}(\Pi) = \mathbf{1}^n \quad (13)$$

$$\Pi = \Pi^T \quad (14)$$

$$\mathbf{A} = \mathbf{A}^T \quad (15)$$

$$[\mathbf{A}]_{i,i} > 0 \quad \forall i \in [n] \quad (16)$$

$$[\mathbf{A}]_{i,j} \geq 0 \quad \forall (i,j) \in [n]^2, i \neq j \quad (17)$$

$$[\mathbf{A}]_{i,j} \leq \Pi_{i,j} \quad \forall (i,j) \in [n]^2, i \neq j \quad (18)$$

$$\|\Pi - \bar{\mathbf{A}}[f_p - 1]\|_F^2 \leq 2 \times n_e. \quad (19)$$

The decision variables \mathbf{A} and Π model the doubly stochastic matrix used for the consensus protocol and the adjacency matrix of the generate configuration respectively. Constraint 11 and Constraint 15 to Constraint 18 ensures that \mathbf{A} is a doubly stochastic matrix that is structurally equivalent to Π . In the light of Theorem 1, Constraint 12 enforces the generated configuration to possess a connected graph. Finally, Constraint 19 encodes the second condition in Problem 1 into the MISDP. If i represents the label of the robot whose sensor quality deteriorated at f_p , then with some simple algebraic manipulation it is easy to see that Equation 10 is equal to $\text{Tr}(\Omega_{f_p}^i(1))$ or $\text{Tr}((\mathbf{P}_{f_p}^i(1))^{-1})$.

2) **TCCG:** Consider the following MISDP formulation encoding the team-centric configuration generation strategy which minimizes $\frac{1}{n} \sum_i \text{Trace}(\mathbf{P}_{f_p}^i(1))$, where $\mathbf{P}_{f_p}^i(1) = (\Omega_{f_p}^i(1))^{-1}$.

$$\begin{aligned} & \underset{\substack{\mathbf{A} \in \mathcal{S}_+^n, \mu \in \mathbb{R}_{>0}, \\ \Pi \in \{0,1\}^{n \times n}, \bar{\Delta} \in \mathcal{S}_+^{n \times s_a}, \\ \Delta_1, \Delta_2, \dots, \Delta_n \in \mathcal{S}_+^{s_a}}}{\text{minimize}} & \frac{1}{n} \text{Tr}(\bar{\mathbf{P}}) \text{ subject to } \begin{bmatrix} \bar{\mathbf{P}} & \mathbf{I} \\ \mathbf{I} & \bar{\Delta} \end{bmatrix} \succeq 0 \end{aligned} \quad (20)$$

$$\mathbf{A} \otimes \mathbf{I} \begin{bmatrix} \Omega_{f_p}^1(0) \\ \Omega_{f_p}^2(0) \\ \vdots \\ \Omega_{f_p}^n(0) \end{bmatrix} = \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_n \end{bmatrix} \quad (21)$$

Where $\bar{\Delta}$ is a block diagonal matrix with $\{\Delta_1, \Delta_2, \dots, \Delta_n\}$ along its diagonal and $\mathbf{A} \otimes \mathbf{I}$ results in the Kronecker product [14] between \mathbf{A} and the identity matrix of the equal size. Constraint 21 is essentially Equation 7 for $L = 1$ written compactly as a single equation for the whole tracker team. Therefore, Δ_i should match the information matrix $\Omega_{f_p}^i(1)$. Note that Constraints 11-19 are also required for TCCG. The following lemma (proof omitted here for brevity) proves that minimizing Equation 20 minimizes $\frac{1}{n} \sum_i \text{Trace}(\mathbf{P}_{f_p}^i(1))$.

Lemma 1. *The $\frac{1}{n} \text{Tr}(\bar{\mathbf{P}})$ is an upper bound on $\frac{1}{n} \sum_i \text{Trace}(\mathbf{P}_{f_p}^i(1))$*

B. Formation Synthesis

We now describe a procedure to assign a physical location to each robot which maximizes the non-overlapping coverage of the space. We also impose constraints so that connected robot pairs remain within communication distance d_{mc} of each other, and that the distance between all robot pairs exceeds d_s to ensure that no two robots collide. An additional constraint is added to ensure that robots which were tracking the target at f_p have the target within their sensing range d_{sen} after reconfiguration.

We define coverage of the space as the total area covered by all robot sensing regions minus the overlapping area covered by two or more robot sensing regions. This produces the following constrained optimization problem:

$$\max_{\{X_{[n]}\}} \pi \sum_{i \in \mathcal{V}} \left((d_{sen}^i)^2 - \sum_{j \in \mathcal{V} \neq i} \frac{(2d_{sen}^i - \|X_i - X_j\|)^2}{2} \right) \quad (22)$$

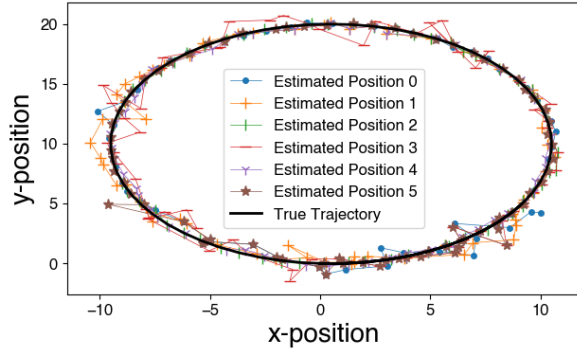


Fig. 3: Comparison of the target agent's true trajectory and its estimate by each robot in a tracking team of six, labeled 0-5.

$$\text{subject to } d_s \leq \|X_i - X_j\| \leq d_{mc} \quad \forall (i, j) \in \mathcal{E} \quad (23)$$

$$d_s \leq \|X_i - X_j\| \quad \forall (i, j) \in \bar{\mathcal{E}} \quad (24)$$

$$B^{\min} \leq X_i \leq B^{\max} \quad \forall i \in V \quad (25)$$

$$\|X_i - X_{target}\| \leq d_{sen}^i \quad \forall i \in \mathcal{V}_{f_p} \quad (26)$$

where d_{sen}^i is the radius of the circular field of vision of tracker, ρ^i , and $B^{\min}, B^{\max} \in \mathbb{R}^3$ are the minimum and maximum extents of an axis-aligned bounding box, with the operator \leq applied elementwise in Equation 25. $\mathcal{V}_{f_p} \subseteq \mathcal{V}$ is the labels of the subset of trackers tracking the target at f_p .

We solve the formation synthesis optimization problem Equation 22 - Equation 26 following the approach described in [9] by simply updating the objective to Equation 22 and incorporating the new constraint Equation 26.

V. SIMULATION

To validate our approach, we conducted multiple simulation experiments of a robot team tracking a target agent following a circular trajectory. We modeled the target trajectory as a Dubins car [21]. In our simulation, the target's state vector $\mathbf{x}_k = [x_k, y_k, \theta_k, v_k]$ describes its position, orientation and velocity at each time step k . The parameters for the linearized discrete dynamics (Equation 1) of the target are

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{G}_k = \begin{bmatrix} \cos \theta_k dt & 0 \\ \sin \theta_k dt & 0 \\ 0 & dt \\ 1 & 0 \end{bmatrix} \quad (27)$$

and input vector $\mathbf{u}_k = [\tilde{v}_k, \tilde{\theta}_k]$. We held \tilde{v}_k and $\tilde{\theta}_k$ constant throughout the simulation. This resulted in the target following a circular path.

For the distributed Kalman filter, we initiate the same \mathbf{H}_k^i and \mathbf{R}_k^i for each robot in the tracker team. We used $L = 15$ for the consensus step. Parameters chosen for the configuration generation and formation synthesis problems were $n_e = 1$, $d_s = 5$, $d_{mc} = 10$, and $d_{sen}^i = 20 \forall i \in [n]$, with a bounding box of $x \in [-100, 100]$, $y \in [-100, 100]$, and $z \in [-100, 100]$.

The target dynamics and distributed Kalman filter were implemented in Python. For the agent-centric approach, the MISDP problem was solved using Python with PICOS as the optimization problem modeling interface and MOSEK as the semi-definite programming solver. For the team-centric approach, the MISDP problem was solved in MATLAB with YALMIP as the modeling interface. In both approaches, the simulated annealing technique for formation synthesis was implemented in Python.

To simulate deteriorating sensor quality for a robot i , we modified its covariance matrix \mathbf{R}_k^i by adding a random positive definite matrix. We generated various deterioration event sequences for robot teams of $n \in \{5, 6, 7\}$ where a random robot was chosen at every f time step of the simulation to experience sensor deterioration. Figure 3 shows an overhead view of a single simulation trial. The true trajectory of the target and the estimated trajectory of the target by each tracker in the team are shown. Figure 4 shows screenshots from a separate trial showing different deterioration instances before and after reconfiguration. A video of this trial is included in the supplementary material.

For comparison of the ACCG and TCCG, we simulated 20 deterioration event sequences for each size robot team.

We also compared both ACCG and TCCG against a *greedy* strategy in which a single edge is added connecting the robot whose sensing quality has deteriorated to the robot with lowest $Tr(\mathbf{P}_{f_p-1})$ at the time of the deterioration event. Each trial was initialized with a line graph.

To quantify performance, we plotted the *maximum* $Tr(\mathbf{P})$ and *maximum estimation error* of the robot team for each strategy at each time step over all trials. In our plots \mathbf{P} is the covariance matrix of a tracker after completing 15 consensus steps of the distributed Kalman filter, or $\mathbf{P}_{f_p}^i(15)$. This differs from $\mathbf{P}_{f_p}^i(1)$, which we optimized in the MISDP formulations. Figure 5 shows the results of these simulations.

While the *greedy* strategy appears to perform best in the early steps of the simulation, TCCG outperforms both ACCG and *greedy* strategies for larger teams as the simulation progresses and sensor quality deteriorates. Interestingly, ACCG performed poorly even compared to the simple *greedy* approach for all team sizes.

To simulate a worst case scenario, we additionally ran simulations in which the same robot experienced sensor deterioration at each time step. Under these conditions, TCCG outperforms ACCG and the greedy strategy at most time steps, as seen in Figure 6, for all simulated team sizes.

VI. CONCLUSION

We describe a novel method that enables a team of robots engaged in tracking a target to reconfigure themselves in response to deterioration in the sensing quality in one of the team members. The reconfigured team alleviates the effect of sensor quality deterioration on team performance. We proposed two methods (both based on optimizing MISDPs) to generate the new configuration. The methods (agent-centric) and the other method (team-centric) were validated in simulation and compared to each other. They were also

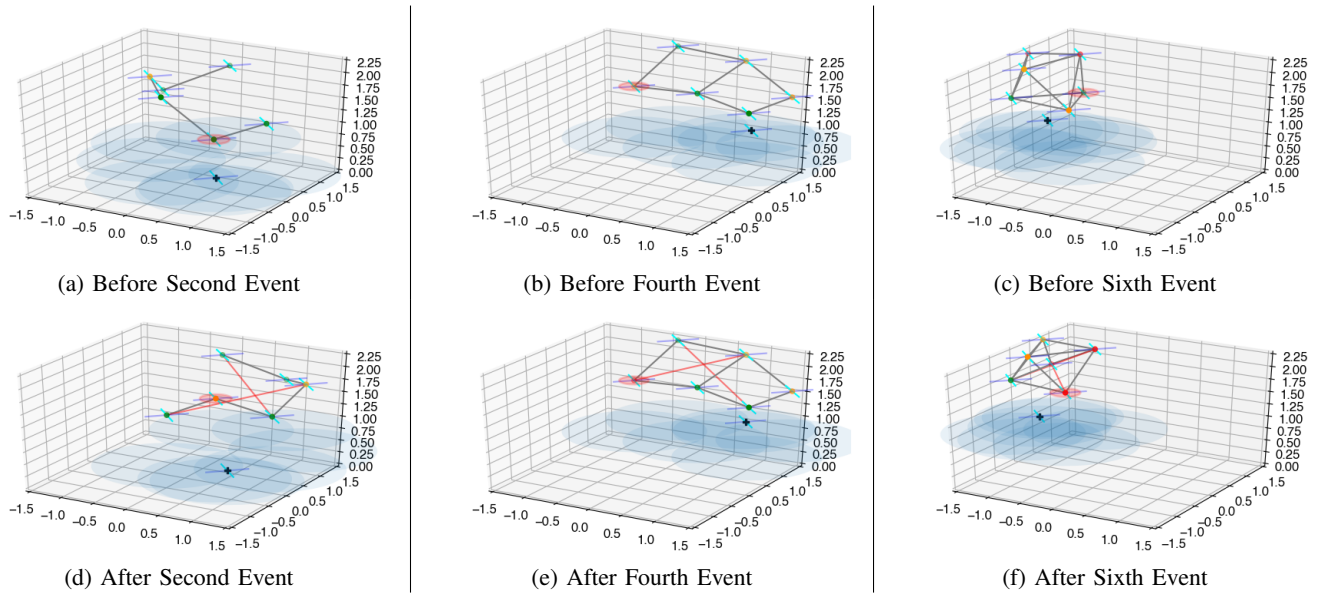


Fig. 4: Screenshots of a simulation in which a six quadrotor team tracks another quadrotor flying below them. Each quadrotor's field of vision is colored in light blue. The figures in the top row depict the formation of the quadrotors before the occurrence of a sensor deterioration event. The corresponding figures on the bottom portray the formation after 1) sensor deterioration is detected, 2) a new communication edge is chosen, and 3) the robots move to their new locations. The quadrotor which experienced sensor deterioration is enveloped using a filled red circle. The quadrotor colors vary from green to red indicating good to poor sensor quality. For visualization purposes, we set $n_e = 2$ for this simulation to illustrate multiple edges added for deterioration events. We also set $d_s = 0.5$, $d_{mc} = 1.0$, and $d_{sen}^i = 1.0 \forall i \in [n]$, with a bounding box of $x \in [-1.5, 1.5]$, $y \in [-1.5, 1.5]$, and $z \in [1.6, 2.25]$.

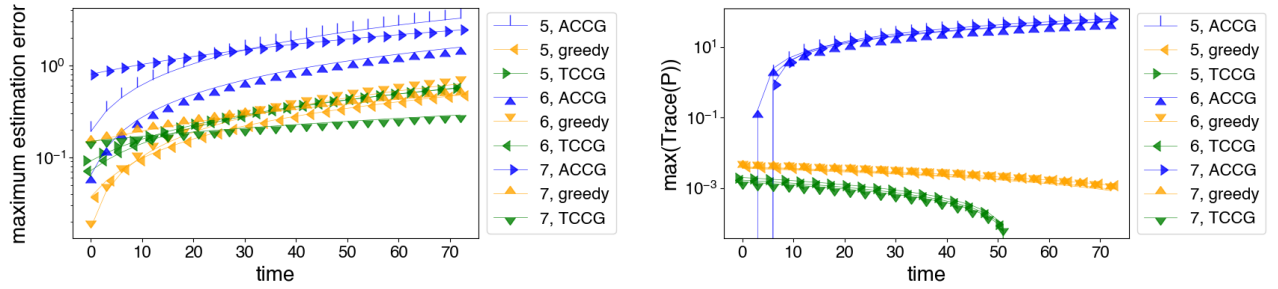


Fig. 5: The comparison (log scale) of *maximum estimation error* and *maximum $Tr(\mathbf{P})$* for all simulations and all team sizes between **ACCG** (blue), **TCCG** (green), and the *greedy* approaches (orange). **TCCG** performs better over time in terms of estimation error for teams of $n = 6$ and $n = 7$. **TCCG** also performs best in terms of the trace of the covariance matrix for all team sizes.

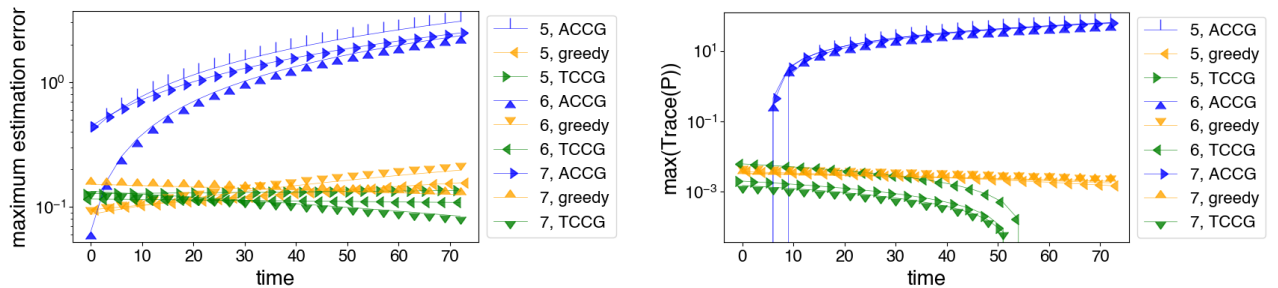


Fig. 6: A comparison of *maximum estimation error* and *maximum $Tr(\mathbf{P})$* between the three different strategies for simulations in which only a single node experienced sensor deterioration. Here, **TCCG** performs best for all simulated team sizes.

compared to a greedy strategy. The team-centric approach outperforms both the agent-centric and greedy approaches. We are currently working on validating our approach on our

multi-robot testbed [22]. We are also extending our approach to handle multi-robot multi-target tracking scenarios where the number of targets are unknown.

REFERENCES

- [1] 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.
- [2] R. Olfati-Saber and P. Jalalkamali, "Collaborative target tracking using distributed kalman filtering on mobile sensor networks," in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 1100–1105.
- [3] R. K. Williams and G. S. Sukhatme, "Observability in topology-constrained multi-robot target tracking," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 1795–1801.
- [4] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1540–1553, 2017.
- [5] Q. Liu, Z. Wang, X. He, and D. Zhou, "On kalman-consensus filtering with random link failures over sensor networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2701–2708, 2017.
- [6] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," in *2014 American Control Conference*, June 2014, pp. 3287–3292.
- [7] D. Leitold, Á. Vathy-Fogarassy, and J. Abonyi, "Controllability and observability in complex networks—the effect of connection types," *Scientific reports*, vol. 7, no. 1, p. 151, 2017.
- [8] R. K. Ramachandran and S. Berman, "The effect of communication topology on scalar field estimation by large networks with partially accessible measurements," in *2017 American Control Conference (ACC)*, May 2017, pp. 3886–3893.
- [9] R. K. Ramachandran, J. A. Preiss, and G. S. Sukhatme, "Resilience by reconfiguration: Exploiting heterogeneity in robot teams," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019, pp. 6518–6525.
- [10] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 3, p. 23, 2010.
- [11] S. Arosh, S. Nayak, S. Duttagupta, *et al.*, "Fitness function based sensor degradation estimation using h infinity filter," *Procedia Computer Science*, vol. 58, pp. 172–177, 2015.
- [12] L. Jiang, D. Djurdjanovic, J. Ni, and J. Lee, "Sensor degradation detection in linear systems," in *Engineering Asset Management*. Springer, 2006, pp. 1252–1260.
- [13] L. Guerrero-Bonilla, A. Prorok, and V. Kumar, "Formations for resilient robot teams," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 841–848, April 2017.
- [14] R. A. Horn and C. R. Johnson, Eds., *Matrix Analysis*. New York, NY, USA: Cambridge University Press, 1986.
- [15] F. Bullo, *Lectures on Network Systems*, 1st ed. Kindle Direct Publishing, 2019, with contributions by J. Cortes, F. Dorfler, and S. Martinez. [Online]. Available: <http://motion.me.ucsb.edu/book-1ns>
- [16] Olfati-Saber, "Distributed kalman filtering for sensor networks," in *Decision and Control, 2007 46th IEEE Conference on*, 2007.
- [17] F. Morbidi and G. L. Mariottini, "On active target tracking and co-operative localization for multiple aerial vehicles," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 2229–2234.
- [18] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based algorithms for distributed filtering," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 794–799.
- [19] M. Rafiee and A. M. Bayen, "Optimal network topology design in multi-agent systems for efficient average consensus," in *49th IEEE Conference on Decision and Control*. IEEE, 2010, pp. 3877–3883.
- [20] D. Xue, A. Gusrialdi, and S. Hirche, "A distributed strategy for near-optimal network topology design," in *21st International Symposium on Mathematical Theory of Networked and Systems (MTNS 2014)*, 2014.
- [21] S. M. LaValle, *Planning Algorithms*. New York, NY, USA: Cambridge University Press, 2006.
- [22] J. A. Preiss*, W. Hönig*, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304.