# Adaptive and Risk-Aware Target Tracking with Heterogeneous Robot Teams

Siddharth Mayya[1], Ragesh K. Ramachandran[2], Lifeng Zhou[1], Gaurav S. Sukhatme[2,3], Vijay Kumar[1]

*Abstract*— We consider a scenario where a team of robots with heterogeneous sensors must track a set of hostile targets which induce sensory failures on the robots. In particular, the likelihood of failures depends on the proximity between the targets and the robots. We propose a control framework that implicitly addresses the competing objectives of performance maximization and sensor preservation (which impacts the future performance of the team). Our framework consists of a predictive component—which accounts for the risk of being detected by the target, and a reactive component— which maximizes the performance of the team regardless of the failures that have already occurred. Based on a measure of the abundance of sensors in the team, our framework can generate aggressive and risk-averse robot configurations to track the targets. Crucially, the heterogeneous sensing capabilities of the robots are explicitly considered in each step, allowing for a more expressive risk-performance trade-off. Simulated experiments with induced sensor failures demonstrate the efficacy of the proposed approach.

## I. INTRODUCTION

Operating in disruptive and failure-prone environments is a fundamental requirement for achieving real-world autonomy in robotics systems [1]. These disruptions can either be adversarial—such as in defense and security applications [2], [3], [4], or non-adversarial—due to the complex and noisy nature of the real world [5]. In such settings, the deployment of multi-robot teams to accomplish tasks adds a layer of flexibility within the system, in the form of increased reliability via redundancy and reconfigurability [6].

Consequently, a significant amount of research effort has been dedicated towards the design of multi-robot coordination algorithms which account for the presence of disturbances, e.g. [7], [8], [9]. Broadly speaking, these approaches consist of two possible elements: a *predictive* risk-aware piece which generates solutions based on a model of the disturbance (typically interpreted as *robustness*) [10], or a *reactive* piece which adapts the underlying control algorithm to continue task execution regardless of the disruptions affecting the system (typically interpreted as *resilience*) [11]. The importance of both these aspects depends on the mission objectives and can change during operations—while the latter
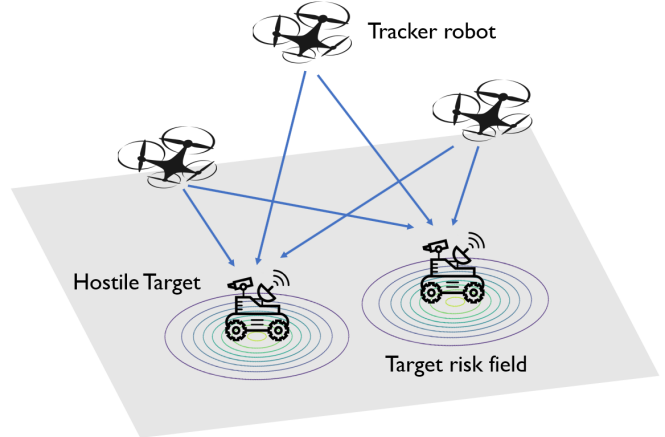
Fig. 1. A depiction of the multi-robot target tracking scenario considered in this paper. A team of robots equipped with heterogeneous noisy sensors tracks a set of mobile targets. The targets are moving and can induce failures in the sensors of the robots based on the proximity between them. Therefore, the robots must position themselves close to the targets to generate accurate estimates, but must simultaneously account for the risk of failures—which impact the quality of future measurements.

prioritizes task performance, the former generates potentially conservative solutions to prevent future disruptions.

These considerations are highly relevant in the context of *heterogeneous* multi-robot teams, where the different capabilities of the robots can be leveraged to magnify the efficacy of the robot team [12]. For example, if only a single robot in the multi-robot team carries an essential sensor required for the success of the task, this robot's actions should be more risk-averse compared to other robots which carry non-essential sensors. A pertinent question then is: *How should one systematically balance the objectives of risk-aversion and performance-focused adaptiveness in heterogeneous multi-robot teams?*

In the context of a target tracking application, this paper presents a framework which synthesizes adaptive and risk-aware control inputs for a team of robots equipped with heterogeneous sensing capabilities. We consider a team of robots tracking mobile adversarial targets in a failure-prone environment. The robots are equipped with different types of noisy sensors, and are tasked with estimating the targets' states. Furthermore, the targets are moving and can induce failures in the sensors of the robots based on the proximity between them (see Fig. 1). In such a scenario, the robots must not only position themselves in order to achieve a trade-off between risk and tracking quality, they must also

suitably adapt to failures that have rendered certain sensors dysfunctional.

We endow the robot team with a Kalman filter [1] which generates optimal estimates of the target locations along with a quantification of the uncertainty corresponding to each target's estimate. To embed risk-awareness in our framework, we introduce the notion of a *safety-aware observability Gramian* (SOG), which weighs the quality of future observations made by the multi-robot team with the risk of failures associated with making those observations. By encoding the heterogeneous sensors available to the robots within the SOG, we enable our framework to position robots based on the contribution of their sensors to the overall tracking performance.

Within our framework, the objectives of risk-aversion and tracking performance maximization are automatically traded off by considering the *sensing margin*—defined as the excess number of operational sensors when compared to the minimum required for collective observability of the targets tracking process. This enables our algorithm to prioritize performance when the sensing margin is large and switch to risk-averse behaviors as the sensing margin of the team decreases. As we demonstrate, when compared to a framework with no risk-awareness, our algorithm preserves the sensors of the robots for longer, resulting in an extended operational time horizon. Moreover, our approach presents the mission designer with an explicit mechanism to tune this trade-off between performance and risk-aversion.

The outline of the paper is as follows. Section II places our work in the context of existing literature. Section III introduces the formulation for the target tracking scenario along with some mathematical notations. Section IV presents the centralized optimization program whose solution generates adaptive and risk-aware configurations for the robot team. Section V presents the results of simulated experiments, and Section VI concludes the paper.

## II. LITERATURE REVIEW

Robust control and planning methods typically account for the risk of future failures and adversarial attacks occurring in the system when generating solutions [10], [13]. Techniques such as $\mathcal{H}_\infty$ control [10] and robust tube MPC [13] have been specifically designed for ensuring operability under "worst-case" disturbances or failures. Recently, similar traits have been investigated to develop robust coordination and planning algorithms in multi-robot systems. For instance, Zhou et al. [14], [15] devised robust target tracking algorithms for robot teams to counter worst-case denial-of-service (DoS) failures or attacks that can make robots fail or compromise their tracking sensors. Their algorithms guarantee a provably close-to-optimal team performance even though some robots or their sensors malfunction. Along this line, robust information collection algorithms [16] were developed while accounting for worst-case robot failures or attacks when a team of robots are jointly planning paths to gather data in the environment. In contrast to these approaches which might generate conservative solutions, [17] adopts a probabilistic

approach, and presents a risk-aware coordination algorithm which addresses the issue of random sensor failures in the problem of sensor placement for environmental monitoring.

As disruptions are inevitable during the long duration operations of robot teams, mechanisms to maintain performance despite failures have been developed [18], [19], [20], [21]. To this end, Saulnier et al. [19] presented a resilient formation control algorithm that achieves the formation despite deceptive signals from non-cooperative team members. Similarly, Ramachandran et al. [20] designed an algorithm to reconfigure the communication network of the robot team in order to compensate for resource failures. This algorithm was then utilized to adapt to failures of sensing resources in a multi-target tracking scenario [18]. A resilient event-driven approach was devised by Song et al. [21] to compensate for the robot failures when a team of robots are tasked to completely explore or cover an environment. Particularly, they designed a game-theoretic approach to trigger the reassignment of the functional robots, e.g., either continuing the coverage task in their pre-assigned workspace or being reassigned to the workspace of a failed robot.

In contrast to these works which focus on either risk-awareness or adaptiveness, we present an optimization framework that integrates *both* predictive and reactive control paradigms in the context of multi-robot multi-target tracking with heterogeneous sensing resources. In particular, our framework systemically exhibits a tradeoff between the performance maximization and the risk-aversion based on the abundance of resources available within the team.

## III. PROBLEM SETUP

This section formally introduces the heterogeneous sensing model of the multi-robot team and sets up the target tracking problem. Following this, we introduce the target-induced sensor failure models, which will be used in Section IV to imbue risk-awareness into the proposed controller.

### A. Notation

In this paper, capital or small letters, bold small letters and bold capital letters represent scalars, vectors and matrices, respectively. Calligraphic symbols are used to denote sets. For any positive integer $z \in \mathbb{Z}^+$, $[z]$ denotes the set $\{1, 2, \cdots, z\}$. $\|\cdot\|_p$ denotes the $p$-norm and the induced $p$-norm for vectors and matrices respectively. We drop the subscript on the norm when referring to the 2-norm. Additionally, $\|\mathbf{M}\|_F \triangleq \sqrt{trace(\mathbf{M}^T\mathbf{M})}$, outputs the Frobenius norm of a matrix $\mathbf{M}$. $\mathbf{1}^{m_1}$ and $\mathbf{1}^{m_1 \times m_2}$ denotes the vector and matrix of ones of appropriate dimension, respectively. $\mathbf{I}_m$ denotes the identify matrix of size $m \times m$. The operator $|\cdot|$ gives the length of a vector applied on a vector and cardinality of a set when operated on a set. Given a vector $\mathbf{v}$, $Diag(\mathbf{v})$ represents a diagonal matrix with elements of $\mathbf{v}$ along its diagonal. Also, $\text{Tr}(\mathbf{M})$ results in the trace of $\mathbf{M}$. We use $\mathbf{M}(i, j)$ to denote the $i, j$ element of $\mathbf{M}$. Given a set of matrices $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$, $[\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n]$ and $[\mathbf{M}_1; \mathbf{M}_2; \ldots, \mathbf{M}_n]$ represent the matrices obtained through horizontal and vertical concatenation of the given matrices

respectively. Since vectors and scalars are special cases of matrices, the definitions apply to them too. Furthermore, $\mathbf{M}_1 \oplus \mathbf{M}_2 \oplus \ldots \oplus \mathbf{M}_n$ yields a block diagonal matrix constructed from the given set of matrices. Similarly, the operation $\mathbf{M}_1 \otimes \mathbf{M}_2$ denotes the Kronecker product between the matrices.

### B. Heterogeneity Model

Consider a team of $N$ robots engaged in the target tracking task, indexed by the set $\mathcal{R} := [N]$. Let $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^p$ denote the state of robot $i$, and $\mathbf{u}_i \in \mathcal{U} \subseteq \mathbb{R}^q$ denote its control input, which steers the state according to the following control-affine dynamics:

$$\dot{\mathbf{x}}_i = f(\mathbf{x}_i) + g(\mathbf{x}_i)\mathbf{u}_i, \tag{1}$$

where $f$ and $g$ are continuously differentiable vector fields.

The robots are equipped with a set of heterogeneous sensors for tracking targets in the environment. Let $U$ denote the number of distinct types of sensors available within the team. Let $\boldsymbol{\Gamma} \in \{0,1\}^{N \times U}$ denote the *sensor matrix* which describes the distribution of sensors over the robot team:

$$\boldsymbol{\Gamma}_{ij} = \begin{cases} 1, & \text{if robot } i \text{ possesses sensor } j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

### C. Target Tracking

The robots are tasked with tracking a set of $M$ targets in the environment, which are indexed by the set $\mathcal{T} := [M]$. Let $\mathbf{e}_j \in \mathcal{X} \subseteq \mathbb{R}^p$ denote the state of target $j \in \mathcal{T}$, whose motion in the environment is described by the following linear dynamics:

$$\dot{\mathbf{e}}_j = \mathbf{A}_j \mathbf{e}_j + \mathbf{B}_j \mathbf{u}_j^e + \mathbf{w}_j^e, \tag{3}$$

where $\mathbf{A}_j$ and $\mathbf{B}_j$ are the process and control matrices, $\mathbf{w}_e^j$ is the zero-mean independent Gaussian process noise with a covariance matrix $\mathbf{Q}_j \in \mathbb{R}^{p \times p}$, and $\mathbf{u}_j^e \in \mathcal{U} \subseteq \mathbb{R}^q$ is the control input of target $j \in \mathcal{T}$. Stacking the target states and target control inputs into the vectors $\mathbf{e}$ and $\mathbf{u}^e$, respectively, we get:

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{B}\mathbf{u}^e + \mathbf{w}^e, \tag{4}$$

where $\mathbf{A} \triangleq \mathbf{A}_1 \oplus \ldots \oplus \mathbf{A}_M$, $\mathbf{B} \triangleq \mathbf{B}_1 \oplus \ldots \oplus \mathbf{B}_M$ and $\mathbf{w}^e \triangleq [\mathbf{w}_1^e; \ldots; \mathbf{w}_M^e]$. As described before, we let $\mathbf{w}^e \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{Q} \triangleq \mathbf{Q}_1 \oplus \mathbf{Q}_2 \oplus \ldots \oplus \mathbf{Q}_N$.

Within our framework, every robot $i$ makes an observation of every target $j$ according to the following linear observation model:

$$\mathbf{y}_{ij} = \mathbf{H}_{ij}\mathbf{e}_j + \boldsymbol{\nu}_{ij}, \quad i \in \mathcal{R}, \ j \in \mathcal{T}, \tag{5}$$

where $\mathbf{y}_{ij}$ is the measurement, $\boldsymbol{\nu}_{ij}$ is the measurement noise, and $\mathbf{H}_{ij}$ is the measurement matrix corresponding to robot $i$'s measurements of target $j$ and will depend on the sensor suite available to robot $i$.

Let $\boldsymbol{\gamma}_i$ denote the vector containing the ordered column indices of the sensor matrix corresponding to robot $i$,

$$\boldsymbol{\gamma}_i = [j \in [U] \mid \boldsymbol{\Gamma}_{ij} = 1]. \tag{6}$$

In this paper, we assume the sensors to be linear, and associate a one-dimensional output matrix corresponding to each sensor available in the team, denoted by the set $\mathcal{S} := \{\mathbf{h}_1, \ldots, \mathbf{h}_{|\mathcal{S}|}\}$. Consequently, the measurement matrix for robot $i$ associated with the process of taking the measurements about the state of target $j$ can be constructed simply using the set $\mathcal{S}$ and $\boldsymbol{\gamma}_i$,

$$\mathbf{H}_{ij} = [\mathbf{h}_{\boldsymbol{\gamma}_i(1)}, \ldots, \mathbf{h}_{\boldsymbol{\gamma}_i(|\boldsymbol{\gamma}_i|)}]^T. \tag{7}$$

We model the measurement noise $\boldsymbol{\nu}_{ij} \sim \mathcal{N}(0, \mathbf{R}_{ij})$ to be a zero-mean Gaussian process where covariance $\mathbf{R}_{ij}$ of the sensor noise is distance dependent, and exponentially increases with the distance between the robots and the targets,

$$\mathbf{R}_{ij}^{-1} = Diag([R_{ij1}^{-1}, R_{ij2}^{-1}, \ldots, R_{ij|\boldsymbol{\gamma}_i|}^{-1}]) \tag{8}$$

with

$$\mathbf{R}_{ijl}^{-1} = w_l \exp\left(-\lambda_l \|\mathbb{P}(\mathbf{x}_i) - \mathbb{P}(\mathbf{e}_j)\|\right), \ \forall l \in [|\boldsymbol{\gamma}_i|], \tag{9}$$

where $\lambda_l$ and $w_l$ determine the noise characteristics of each sensor $l$ in the sensor suite of robot $i$, and $\mathbb{P}$ is a projector operator that maps the state of the robot to it's position. While this paper chooses an exponential decay model to capture the sensor performance characteristics, any other continuously differentiable degradation function can be utilized instead.

In ensemble form, the measurements of all the targets by robot $i$ can be written as:

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{e} + \boldsymbol{\nu}_i, \ \forall i \in \mathcal{R}, \tag{10}$$

where $\mathbf{H}_i = I_M \otimes \mathbf{H}_{ij}$, and $\boldsymbol{\nu}_i = [\boldsymbol{\nu}_{i1}, \ldots, \boldsymbol{\nu}_{iM}]^T$. Here $I_M$ denotes the identity matrix of size $M$. The measurement equation for the team can be written as:

$$\mathbf{y} = \mathbf{H}\mathbf{e} + \boldsymbol{\nu}, \tag{11}$$

where $\mathbf{y} = [\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_N]$, $\mathbf{H} = [\mathbf{H}_1; \mathbf{H}_2; \ldots; \mathbf{H}_N]$ and $\boldsymbol{\nu} = [\boldsymbol{\nu}_1; \boldsymbol{\nu}_2; \ldots; \boldsymbol{\nu}_N]$. The team-wise measurement noise can be written as $\boldsymbol{\nu} \sim \mathcal{N}(0, \mathbf{R})$ where $\mathbf{R} = \mathbf{R}_{11} \oplus \mathbf{R}_{12} \oplus \cdots \oplus \mathbf{R}_{NM}$. Thus, given a sensor matrix $\boldsymbol{\Gamma}$, one can construct the corresponding robot output matrix and the team output matrix. These sensing models will be utilized in Section IV to compute the estimates of the target positions and regulate the tracking performance of the robots.

### D. Target-induced Risk Model

As discussed in the introduction, each target can detect and induce failures within the robot team. Towards this end, let $\phi_j : \mathcal{X} \to [0,1]$ represent the *target risk field* which models the risk of a robot being detected by target $j \in \mathcal{T}$. Formally, $\phi_j(\mathbf{x}_i)$ represents the probability of robot $i$ with state $\mathbf{x}_i$ being detected by target $j$. In addition, we define the *immunity field* $\pi_j : \mathcal{X} \to \mathbb{R}^+$ associated with $\phi_j(\mathbf{x}_i)$ as

$$\pi_j(\mathbf{x}_i) = -\log(\phi_j(\mathbf{x}_i)). \tag{12}$$

Note that, $\pi_j$ increases with the decrease in $\phi_j$. Thus, the immunity field $\pi_j$ encodes the safety of a robot in the vicinity

of target $j$. Intuitively, it quantifies the ability of robot $i$ to evade detection from target $j$ while residing at $\mathbf{x}_i$. Depending on the context, we refer to this model as either the risk model or the safety model. Furthermore, we assume that the functional form of the target risk fields are known to the robot team (see Section V for particular choices of these functions). The next section formulates the optimization program to track the targets in an adaptive and risk-aware manner.

## IV. ADAPTIVE AND RISK-AWARE TARGET TRACKING

Given the measurements of the targets as well as the risk model, this section first introduces the Kalman filter equations which will be used to measure the performance of the team. Subsequently, we introduce a safety-aware observability framework, which is then used to formulate the optimization program for generating the positions of the robots.

### A. Generating Target Estimates

As discussed above, the primary objective of the robots is to generate accurate estimates of the positions of the targets. We achieve this using a Kalman filter which generates an optimal estimate for the states of all targets $\hat{\mathbf{e}}$ along with a team-wise covariance matrix $\mathbf{P}$. Typically, the estimation by KF contains prediction and update steps. In the prediction step, we have,

$$\hat{\mathbf{e}}_- = \mathbf{A}\hat{\mathbf{e}}_0, \tag{13}$$

$$\mathbf{P}_- = \mathbf{A}\mathbf{P}_0\mathbf{A}^T + \mathbf{Q}, \tag{14}$$

where $\hat{\mathbf{e}}_-$ and $\mathbf{P}_-$ are (a priori) estimated states of the targets and the (a priori) team-wise covariance matrix at the next step, respectively; $\hat{\mathbf{e}}_0$ and $\mathbf{P}_0$ are the estimated states of the targets and the team-wise covariance matrix at the current step, respectively; and $\mathbf{Q}$ is the process noise covariance corresponding to all targets. Once the robots' measurements $\mathbf{y}$ (Eq. 11) are available, we have the KF update step as,

$$\hat{\mathbf{e}} = \hat{\mathbf{e}}_- + \mathbf{K}\widetilde{\mathbf{y}}, \tag{15}$$

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_-, \tag{16}$$

where $\hat{\mathbf{e}}$ and $\mathbf{P}$ are the (a posteriori) estimated states of the targets and the (a posteriori) team-wise covariance matrix at the next step; $\widetilde{\mathbf{y}} := \mathbf{y} - \mathbf{H}\hat{\mathbf{e}}_-$ is measurement pre-fit residual; $\mathbf{K} := \mathbf{P}_-\mathbf{H}^T(\mathbf{H}\mathbf{P}_-\mathbf{H}^T + \mathbf{R})^{-1}$ is the Kalman gain with $\mathbf{R}$ denoting the covariance matrix of the robot team measurement noise $\boldsymbol{\nu}$, as described in (11). Notably, the team-wise aposteriori covariance matrix $\mathbf{P}$ can be expressed as $\mathbf{P} = \mathbf{P}_1 \oplus \mathbf{P}_2 \oplus \ldots \oplus \mathbf{P}_M$ with $\mathbf{P}_j$ denoting the (a posteriori) covariance matrix corresponding to all robots' joint estimate of target $j \in \mathcal{T}$.

### B. Imparting Risk-aware Behaviors

In order to encode risk-awareness into our framework, we leverage the observability of the system of moving targets. More specifically, the observability Gramian associated with the multi-robot team tracking the targets is defined as: $\mathbf{O} \triangleq \sum_{k=0}^{T-1}(\mathbf{A}^T)^k\mathbf{H}^T\mathbf{H}(\mathbf{A})^k$, where $T \geq 0$ is a suitably chosen time horizon. The positive definiteness of $\mathbf{O}$ is a

necessary and sufficient condition for the observability of a deterministic linear system [22]. Intuitively, $\mathbf{O}$ measures the ease of estimating the initial state of a deterministic linear system from a given sequence of measurements on the state of the system. A measure of the observability Gramian matrix (such as the trace or the determinant) is typically used to measure the ease of estimating the initial state of the linear system [23].

In our scenario, proximity between the robots and the targets will prove detrimental to the ability of the robots to sense the targets in the future, due to the increasing risk of target-induced sensor failures. To encode this, we introduce a weighted observability Gramian, expressed as:

$$\mathbf{O}_{\boldsymbol{\Pi}} \triangleq \sum_{k=0}^{T-1}(\mathbf{A}^T)^k\mathbf{H}^T\boldsymbol{\Pi}\mathbf{H}(\mathbf{A})^k, \tag{17}$$

where $\boldsymbol{\Pi}$ is a positive definite matrix quantifying the relative safety of the different sensor-equipped robots. As the detection models described in Section III-D are purely dependent on the states of the targets and the robots, we can define $\boldsymbol{\Pi}$ as,

$$\boldsymbol{\Pi} = \oplus_{i=1}^{N}\left(\oplus_{j=1}^{M} I_{|\boldsymbol{\gamma}_i|} \otimes \pi_j(\mathbf{x}_i)\right) \tag{18}$$

where $\pi_j(\mathbf{x_i})$ is the immunity field with respect to robot $i$ and target $j$, as defined in (12). Note that, the block diagonal matrices $I_{|\boldsymbol{\gamma}_i|} \otimes \pi_j(\mathbf{x}_i)$ which compose the matrix $\boldsymbol{\Pi}$ represent the inverse risk associated with target $j$'s detection of robot $i$.

In other words, larger the diagonal entries of $\mathbf{O}_{\boldsymbol{\Pi}}$, larger is the ability of the robot team to obtain the targets' state measurements while evading detection by the targets. Thus, we refer to the matrix $\mathbf{O}_{\boldsymbol{\Pi}}$ as *safety aware observability gramian* (SOG) matrix. In the optimization problem described in Section IV, we bound the trace of $\mathbf{O}_{\boldsymbol{\Pi}}^{-1}$ from above to control the risk-averseness of the robot team.

### C. Optimization Problem

We propose solving the following optimization program to obtain the states of the robots for target tracking. In the following, we will describe the various contraints and symbols constituting the problem description.

**Adaptive and Risk-Aware Target Tracking** $\hspace{1em}$ (19)

$$\underset{\mathbf{x}, \boldsymbol{\delta}_1, \delta_2}{\text{minimize}} \; k_1(\Delta, \boldsymbol{\delta}_1) + k_2(\Delta, \delta_2) \tag{19a}$$

$$s.t. \; \|\mathbf{x}_i - \mathbf{x}_{i0}\| \leq d_m, \; \forall i \in \mathcal{R}, \tag{19b}$$

$$\|\mathbf{x}_i - \mathbf{x}_j\| \geq d_n, \; \forall i \neq j, \; \forall i,j \in \mathcal{R}, \tag{19c}$$

$$\text{Tr}(\mathbf{P}_j) \leq \boldsymbol{\rho}_{1,j} + \delta_{1,j}, \; \delta_{1,j} \geq 0, \; j \in \mathcal{T}, \tag{19d}$$

$$\text{Tr}(\mathbf{O}_{\boldsymbol{\Pi}}^{-1}) \leq \rho_2 + \delta_2, \; \delta_2 \geq 0, \tag{19e}$$

where $\mathbf{x}_{i0}$ is robot $i$'s state at the current time step, $\mathbf{x} := [\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_N^T]^T$ is the optimization variable representing the ensemble state of all robots at the next time step, $d_m$ is the maximum distance each robot $i$ can move between two consecutive time steps, and $d_n$ is the minimum inter-robot distance for safety. Furthermore, $\boldsymbol{\rho}_1 =$

$[\boldsymbol{\rho}_{1,1}, \boldsymbol{\rho}_{1,2}, \cdots, \boldsymbol{\rho}_{1,M}]^T$ and $\rho_2$ denote the user-defined upper bounds on the tracking error (encoded by $\mathbf{P}$) and team-level risk (encoded by $\mathbf{O}_{\boldsymbol{\Pi}}^{-1}$), respectively. We would like our adaptive framework to continue operations even when sensor failures might make it impossible to achieve desired specifications. To this end, we introduce slack variables $\boldsymbol{\delta}_1$ and $\delta_2$ where $\boldsymbol{\delta}_1 := [\boldsymbol{\delta}_{1,1}, \boldsymbol{\delta}_{1,2}, \cdots, \boldsymbol{\delta}_{1,M}]^T$ introduces a separate performance slack variable for each target.

We now describe the various constraints introduced in (19). Equation 19b limits the distance that each robot can travel between two consecutive steps. Equation 19c guarantees that the generated solutions maintain a minimum safety distance between the robots. Equation 19d nominally aims to bound the trace of the posterior covariance matrix $\mathbf{P}_j$ (16) for each target $j$ by a constant $\boldsymbol{\rho}_{1,j}$. However, sensor failures might imply that meeting this performance criteria is not longer possible. To this end, the slack variable $\boldsymbol{\delta}_1$ is introduced to ensure that the framework adapts to the current operating conditions. Similarly, the risk in the team tracking performance is captured by the inverse SOG $\mathbf{O}_{\boldsymbol{\Pi}}^{-1}$ and it is upper bounded by the sum of $\rho_2$ and the slack variable $\delta_2$ to ensure a desired level of safety (19e). It should be noted that $\mathbf{P}$ is computed directly using the update equation (16) and $\mathbf{O}_{\boldsymbol{\Pi}}$ is computed using the estimated state of the targets $\hat{\mathbf{e}}$ computed in (17).

### D. Objective Function Design

As described in the introduction, balancing the competing objectives of performance maximization and risk-aversion is a desirable feature of a resource-aware control framework. In this paper, we enable our framework to implicitly make this trade-off based on a quantification of the abundance of sensors available in the team. This is encoded within the cost functions $k_1$ and $k_2$ as described next.

First, we define the resource abundance $\Delta$ as the difference between the current sensor matrix $\boldsymbol{\Gamma}$ and the minimal heterogeneous resources required for collective observability of the targets $\boldsymbol{\Gamma}_{\min}$,

$$\Delta = \|\boldsymbol{\Gamma}\|_{\mathcal{F}} - \|\boldsymbol{\Gamma}_{\min}\|_{\mathcal{F}}, \tag{20}$$

where $\boldsymbol{\Gamma}_{\min}$ is defined as,

$$\boldsymbol{\Gamma}_{\min} = \arg\min_{\boldsymbol{\Gamma}} \|\boldsymbol{\Gamma}\|_{\mathcal{F}}^2$$
$$s.t. \ \det\left(\sum_{k=0}^{T-1} (\mathbf{A}^T)^k \mathbf{H}^T \mathbf{H} (\mathbf{A})^k\right) > 0. \tag{21}$$

Notably, (21) computes the smallest set of sensors which will yield collective observability of the system.

Generally speaking, with fewer heterogeneous resources (i.e., a lower $\Delta$), the robot team should lower its expectations on the tracking performance by choosing a larger $\|\boldsymbol{\delta}_1\|_1$ (in (19d)) and be more risk-averse by choosing a smaller $\delta_2$ (in (19e)). Conversely, with more heterogeneous resources, the robots should aim to achieve a higher tracking quality by choosing a smaller $\|\boldsymbol{\delta}_1\|_1$ and can be more risk-seeking by choosing a larger $\delta_2$. To this end, $k_1(\Delta, \boldsymbol{\delta}_1)$ belongs to a class of functions $f : [0, +\infty] \times [0, +\infty] \to [0, +\infty]$

that is monotone increasing in both $\|\boldsymbol{\delta}_1\|_1$ and $\Delta$. While, $k_2(\Delta, \delta_2)$ belongs to a class of functions $f : [0, +\infty] \times [0, +\infty] \to [0, +\infty]$ that is monotone increasing in $\delta_2$ and yet monotone decreasing in $\Delta$. Simple examples of $k_1(\Delta, \boldsymbol{\delta}_1)$ and $k_2(\Delta, \delta_2)$ can be

$$k_1(\Delta, \boldsymbol{\delta}_1) = w_1 \Delta \|\boldsymbol{\delta}_1\|_1, \tag{22}$$
$$k_2(\Delta, \delta_2) = w_2 \frac{1}{\Delta} \delta_2^2, \tag{23}$$

where $w_1, w_2$ are positive scalars. Notably, optimizing over such functions in the objective (19a) enables the robots to make adaptive and risk-aware decisions given the abundance of the heterogeneous resources they have. While this paper utilizes a simple instantiation of the functions $k_1$ and $k_2$, more complex functions can be designed for achieving fine-tuned behaviors.

### E. Computational Aspects

The optimization problem presented in (19) can be solved at discrete time intervals $t \in \mathbb{N}$ based on the updated estimates of the target states. Algorithm 1 illustrates the operations of the target tracking framework. Step 3 uses the Kalman filter to generate estimates of the target positions, $\hat{\mathbf{e}}$. Using this estimate, Step 4 generates new positions for the robots according the tradeoffs between performance maximization and risk-aversion. We assume that the robots drive to the configuration generated by the optimization program within this step (ensured by (19a)) [24]. Following this, the robots evaluate which of their sensors have failed due to proximity with the targets, and update the sensing margin $\Delta$ in Step 6. See Fig. 2 for a system diagram illustrating these operations.

---

**Algorithm 1** Adaptive and Risk-Aware Target Tracking
___
**Require:**
    Robot team heterogeneity specifications: $\boldsymbol{\Gamma}, \mathbf{H}$
    Target and Team Specs: $N, M, \phi_j, w_l, \lambda_l, d_m, d_n$
    Parameters: $\boldsymbol{\rho}_1, \rho_2, w_1, w_2$
1: Initialize: $t = 0$
2: **while** true **do**
3:     Update target position estimate $\hat{\mathbf{e}}$      ▷ (15)
4:     Execute the adaptive risk-aware controller      ▷ (19)
5:     Compile sensor failures and update $\boldsymbol{\Gamma}$
6:     Update sensing margin $\Delta$      ▷ (20)
7:     $t = t + 1$
8: **end while**

---

## V. SIMULATED EXPERIMENTS

In this section, we present a series of simulated experiments which demonstrate the ability of the proposed framework to track multiple targets in an adaptive and risk-aware manner using a team of robots with heterogeneous sensors. We simulate a team of ground robots operating in $\mathbb{R}^2$, with single integrator dynamics: $\dot{\mathbf{x}} = \mathbf{u}$, where $\mathbf{x}$ denotes the positions of the robots. The robots are equipped with
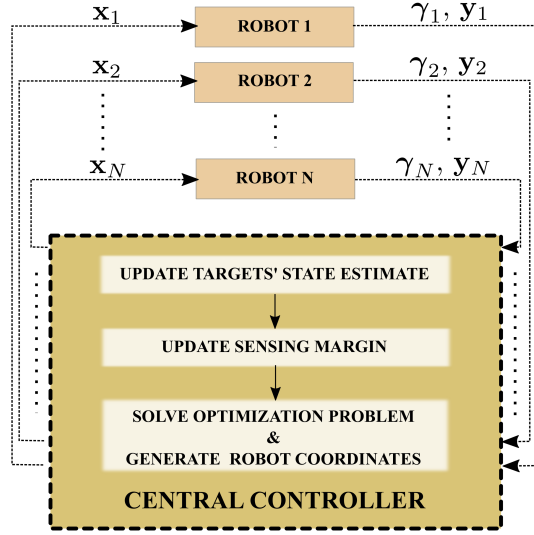
Fig. 2. Operations of the proposed adaptive and risk-aware target tracking strategy. A central controller estimates the targets' state and sensing margin based on the information received from the robots. Using the targets' state estimate and the sensing margin, the central controller solves the optimization problem depicted in (19) to generate new coordinates for the robots.

three different types of sensors, whose reduced measurement matrices are given as:

$$h_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \; h_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}, \; h_3 = \begin{bmatrix} 1 & 1 \end{bmatrix}. \quad (24)$$

We enable the centralized controller (Fig. 2) to run a Kalman filter using the noisy measurements obtained from the robots and update the posteriori covariance matrix $\mathbf{P}$.

The target risk field $\phi_j$ corresponding to each target $j \in \mathcal{T}$ is described by a Gaussian function centered around the target,

$$\phi_j(\mathbf{x}) = \frac{c_j}{2\pi|\Sigma_j|} \exp\left( -\frac{1}{2}(\mathbf{x} - \mathbf{e}_j)^T \Sigma_j (\mathbf{x} - \mathbf{e}_j) \right), \quad (25)$$

where $c_j$ determines the peak value of the field for target $j$, and $\Sigma_j$ determines the variance. The optimization problem given in (19) was solved using SNOPT [25] via pydrake [26].

### A. Two robots, two targets

For the first set of experiments, we consider a team of two robots, whose sensor matrix at deployment time is specified as $\mathbf{\Gamma} = \mathbf{1}^{2 \times 3}$. The target risk field parameters are chosen as $c_j = 3$ and $\Sigma_j = Diag([2.0, 2.0])$, $\forall j \in \mathcal{T}$. Sensor failures on the robots are simulated by flipping a biased coin proportional to the magnitude of the target risk field at the robot's location. While the true state of the targets $\mathbf{e}$ is used to simulate the failures, the robots compute $\mathbf{O}_{\mathbf{\Pi}}$ using the estimated state of the targets $\hat{\mathbf{e}}$ as they do not have access to the true target states.

The parameters for the sensor covariance matrices are chosen as $w_l = 1.80, \lambda_l = 0.1, \forall l \in [|\mathcal{S}|]$, and the target dynamics are set to $\mathbf{A}_j = Diag([1,1]), \mathbf{B}_j = Diag([1,1])$, $\forall j \in \mathcal{T}$. For the parameters ($d_m = 0.33, d_n = 2.0, \boldsymbol{\rho}_1 = [0.5, 0.5], \rho_2 = 0.1, w_1 = 1, w_2 = 100$), Fig. 3a

illustrates the motion trails of the robots (black dots) relative to the true target positions (red squares) and the estimated target positions (blue translucent squares) as the simulation progresses. As seen, the robots approach the targets while accounting for the risk of detection and failures (depicted by the shaded purple regions). Figure 4 illustrates how the
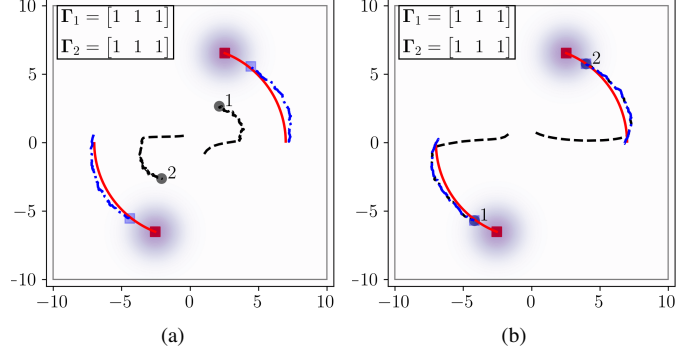


Fig. 3. Motion trails of the robots (black dots), mobile targets (red squares), and estimated target locations (blue translucent squares) executing the adaptive & risk-aware control framework presented in this paper. The target risk fields are illustrated by the shaded regions around the targets. (a) illustrates how the risk-aware constraint (19e) enables the robots to trade-off between current and future tracking performance by maintaining a distance from the targets. This can be contrasted with the behavior in (b) where the risk-aware constraint has been removed. Consequently, the robots position themselves very close to the targets to optimize for tracking performance but at the risk of higher sensor failures rates which detrimentally affects their future performance.

tracking error corresponding to each target—measured by $\text{Tr}(\mathbf{P}_j)$—increases as failures affect the system. It should be noted that, despite not meeting the desired maximum tracking error $\boldsymbol{\rho}_1$, the team continues to perform the task, thanks to the slack variable $\boldsymbol{\delta}_1$ introduced in constraint (19d).
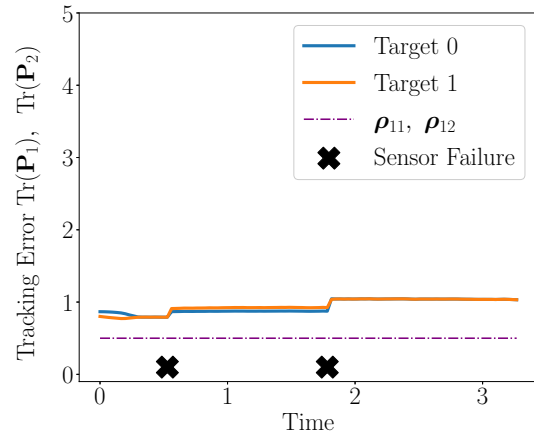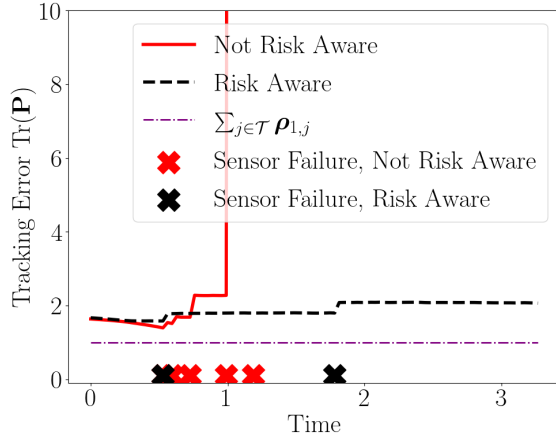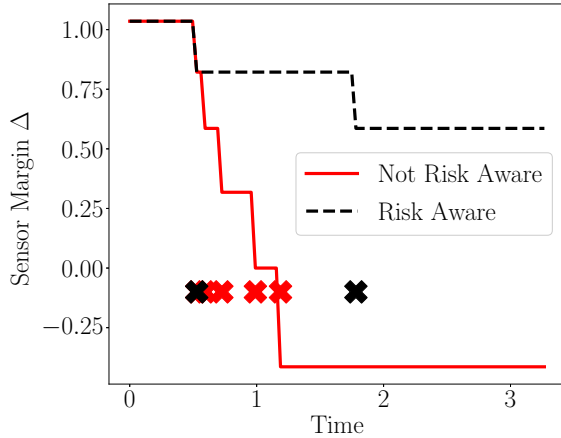


Fig. 4. Tracking error for a team of two robots executing the adaptive & risk-aware target tracking control presented in (19). As seen, the robots do not achieve their tracking performance objectives $\boldsymbol{\rho}_1$ due to the safety Gramian constraint imposed in (19e) as well as sensor failures induced by the targets. However, the adaptive nature of the performance optimization allows the team to continue tracking the targets while balancing the objectives of current performance and safe future operations.

To demonstrate the advantage of the control framework

introduced in this paper, we compare the performance of our controller with and without the SOG constraint given by (19e). In this scenario, the robots drive close to the targets in an attempt to obtain accurate estimates of the target positions (see Fig. 3b). Figure 5 compares the tracking performance and sensor margin over the course of a simulation, and indicates the times at which the sensor failures occurred. As seen, without accounting for the detrimental effects of future failures, the robots drive very close to the targets, and consequently experience a sequence of sensor failures proportional to the target risk field. These simulations demonstrate the ability of our risk-aware adaptive controller to not only continue operations in the face of sensor failures, but also operate for a longer period of time by accounting for future failures.

## B. Four robots, two targets

For the second set of experiments, we consider a team of 4 robots, which are equipped with the same types of three sensors described in (24). We assign the the following sensor matrix for the team at deployment time: $\gamma_1 = [1, 2, 3], \gamma_2 = [1, 2, 3], \gamma_3 = [3], \gamma_4 = [1, 2]$. For the parameters ($d_m = 0.33, d_n = 2.0, c_j = 1, \Sigma_j = Diag([4.0, 4.0]), \rho_1 = [0.45, 0.45], \rho_2 = 0.1, w_1 = 1, w_2 = 500$), Fig. 6 compares the performance of the algorithm with and without the risk-aware constraint (19e). In particular, we present the averaged results over 10 simulation runs to demonstrate the consistent performance of the developed framework. As seen, not only does the proposed framework ensure the longer operability of the robot team, the variance in the tracking performance is also lower when compared to the case with no risk-awareness.
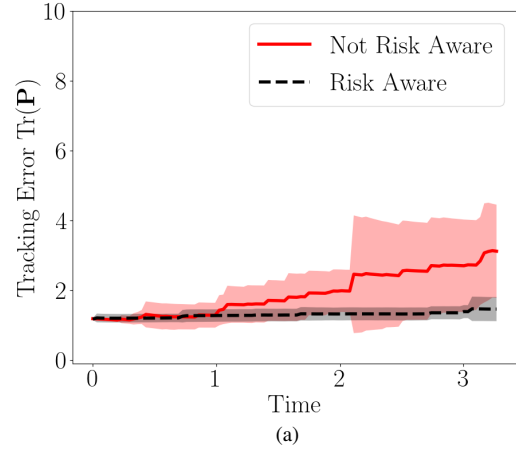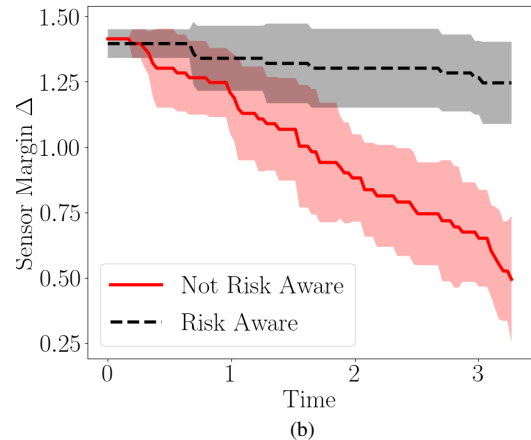


(a)



(b)

Fig. 5. Comparison of total tracking error with and without the risk aware controller encoded in constraint (19e). In the case where future performance is not accounted for, the robots drive very close to the target with the aim of meeting their current performance specifications. However, the larger number of sensor failures dramatically increases the tracking error of the robots (see (a)). This corresponds to a decreasing sensor margin as seen in (b). In contrast, the proposed risk-aware adaptive controller enables the multi-robot team to track the targets for a longer time horizon, and maintains a larger sensing margin $\Delta$ over the simulation time horizon.



(a)



(b)

Fig. 6. Performance of the proposed adaptive and risk-aware target tracking controller in the face of target-induced failures (4 robots, 2 targets). Similar to Fig. 5, the addition of the risk-aware component encoded by constraint (19e) enables the proposed controller to balance performance maximization with the quality of future measurements—allowing for better performance over a longer time horizon. These results were averaged over 10 simulation runs to demonstrate consistent performance. The solid lines denote the mean values, and the shaded region depicts the ±1 standard deviation. This is especially clear in (a) where the variance over the simulation runs is lower in the case of the risk-aware controller.

One of the salient features of the proposed framework is the automatic trade-off between performance maximization and risk-aversion which accounts for the excess of sensors available within the team, as defined in (20). Towards this end, Fig. 7 compares the team-wide tracking quality $\mathrm{Tr}(\mathbf{P})^{-1}$ and the safety metric $\mathrm{Tr}(\mathbf{O_\Pi})$ for varying sensor margin values. From Fig. 7 it is interesting to observe that the system prioritizes performance over safety when the abundance of sensors in the team is high, but flips this relation as the abundance reduces. Note that, to illustrate this point, the sensor margin $\Delta$ was not computed using (20) but was directly specified to the optimization program which then generated the robot configurations accordingly.
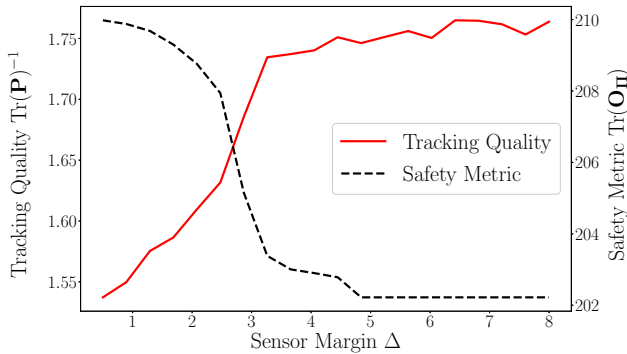


Fig. 7. The automatic trade-off between performance and risk-aversion within the proposed control framework as a function of the resource abundance within the team. Thanks to the encoding of sensing margin $\Delta$ within the cost functions $k_1$ and $k_2$ (see (23)), the optimization-based controller in (19) inherently prioritizes performance for high sensor margins, and prioritizes safety as the sensor margin decreases. This is reflected in the tracking quality and safety metric of the generated robot configurations.

## VI. Conclusion

This paper developed a framework to leverage the heterogeneous sensors in a robot team to track hostile targets while simultaneously accounting for current and future tracking performance. The cost functions $k_1$ and $k_2$, which were a function of the sensor abundance $\Delta$, are an implicit mechanism to trade-off between risk-aversion and performance maximization. While only a simple version of these functions were considered in this paper, future investigations might reveal a systematic mechanism to design these functions to obtain a desired outcome.

## References

[1] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[2] Efrat Sless, Noa Agmon, and Sarit Kraus. Multi-robot adversarial patrolling: facing coordinated attacks. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1093–1100, 2014.

[3] Venkatraman Renganathan and Tyler Summers. Spoof resilient coordination for distributed multi-robot systems. In *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 135–141. IEEE, 2017.

[4] Yoonchang Sung. *Multi-robot coordination for hazardous environmental monitoring*. PhD thesis, Virginia Tech, 2019.

[5] Eliahu Khalastchi and Meir Kalech. Fault detection and diagnosis in multi-robot systems: a survey. *Sensors*, 19(18):4019, 2019.

[6] Jorge Cortés and Magnus Egerstedt. Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, 10(6):495–503, 2017.

[7] Alphan Ulusoy, Stephen L Smith, Xu Chu Ding, and Calin Belta. Robust multi-robot optimal path planning with temporal logic constraints. In *2012 IEEE International Conference on Robotics and Automation*, pages 4693–4698. IEEE, 2012.

[8] M Bernardine Dias, Marc Zinck, Robert Zlot, and Anthony Stentz. Robust multirobot coordination in dynamic environments. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 3435–3442. IEEE, 2004.

[9] Wenhao Luo and Katia Sycara. Minimum k-connectivity maintenance for robust multi-robot systems. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7370–7377. IEEE, 2019.

[10] Kemin Zhou and John Comstock Doyle. *Essentials of robust control*, volume 104. Prentice hall Upper Saddle River, NJ, 1998.

[11] Luiz FO Chamon, Alexandre Amice, Santiago Paternain, and Alejandro Ribeiro. Resilient control: Compromising to adapt. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 5703–5710. IEEE, 2020.

[12] Yara Rizk, Mariette Awad, and Edward W Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–31, 2019.

[13] Shuyou Yu, Christoph Maier, Hong Chen, and Frank Allgöwer. Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems. *Systems & Control Letters*, 62(2):194–200, 2013.

[14] Lifeng Zhou, Vasileios Tzoumas, George J Pappas, and Pratap Tokekar. Resilient active target tracking with multiple robots. *IEEE Robotics and Automation Letters*, 4(1):129–136, 2018.

[15] Lifeng Zhou, Vasileios Tzoumas, George J Pappas, and Pratap Tokekar. Distributed attack-robust submodular maximization for multi-robot planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2479–2485. IEEE, 2020.

[16] Brent Schlotfeldt, Vasileios Tzoumas, Dinesh Thakur, and George J Pappas. Resilient active information gathering with mobile robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4309–4316. IEEE, 2018.

[17] Lifeng Zhou and Pratap Tokekar. An approximation algorithm for risk-averse submodular optimization. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 144–159. Springer, 2018.

[18] Ragesh K Ramachandran, Nicole Fronda, and Gaurav S Sukhatme. Resilience in multi-robot target tracking through reconfiguration. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4551–4557. IEEE, 2020.

[19] Kelsey Saulnier, David Saldana, Amanda Prorok, George J Pappas, and Vijay Kumar. Resilient flocking for mobile robot teams. *IEEE Robotics and Automation letters*, 2(2):1039–1046, 2017.

[20] Ragesh K Ramachandran, James A Preiss, and Gaurav S Sukhatme. Resilience by reconfiguration: Exploiting heterogeneity in robot teams. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6518–6525. IEEE, 2019.

[21] Junnan Song and Shalabh Gupta. Care: Cooperative autonomy for resilience and efficiency of robot teams for complete coverage of unknown environments under robot failures. *Autonomous Robots*, 44(3):647–671, 2020.

[22] Joao P Hespanha. *Linear systems theory*. Princeton university press, 2018.

[23] Vasileios Tzoumas. *Resilient submodular maximization for control and sensing*. PhD thesis, University of Pennsylvania, 2018.

[24] Li Wang, Aaron D Ames, and Magnus Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[25] Philip E. Gill, Walter Murray, Michael A. Saunders, and Elizabeth Wong. User's guide for SNOPT 7.7: Software for large-scale nonlinear programming. Center for Computational Mathematics Report CCoM 18-1, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2018.

[26] Russ Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019.