Enhancing Security in Multi-Robot Systems through Co-Observation Planning, Reachability Analysis, and Network Flow

Ziqi Yang, Roberto Tron *Member, IEEE*

***Abstract*—This paper addresses security challenges in multi-robot systems (MRS) where adversaries may compromise robot control, risking unauthorized access to forbidden regions. We propose a novel multi-robot optimal planning algorithm that integrates mutual observations and introduces *reachability* constraints for enhanced security. This ensures that, even with adversarial movements, compromised robots cannot breach forbidden regions without missing scheduled co-observations. The reachability constraint uses ellipsoidal over-approximation for efficient intersection checking and gradient computation. To enhance system resilience and tackle feasibility challenges, we also introduce *sub-teams*. These cohesive units replace individual robot assignments along each route, enabling redundant robots to deviate for co-observations across different trajectories, securing multiple sub-teams without requiring modifications. We formulate the *cross-trajectory co-observation* plan by solving a network flow coverage problem on the checkpoint graph generated from the original unsecured MRS trajectories, providing the same security guarantees against plan-deviation attacks. We demonstrate the effectiveness and robustness of our proposed algorithm, which significantly strengthens the security of multi-robot systems in the face of adversarial threats.**

***Index Terms*—Multi-robot system, cyber-physical attack, trajectory optimization, reachability analysis, network flow**

I. Introduction

Multi-robot systems (MRS) have found wide applications in various fields. While offering numerous advantages, the distributed nature and dependence on network communication render the MRS vulnerable to cyber threats, such as unauthorized access, malicious attacks, and data manipulation [1]. This paper addresses a specific scenario in which robots are compromised by attacker and directed to *forbidden regions*, which may contain security-sensitive equipment or human workers. Such countering deliberate deviations, termed *plan-deviation attacks*, are identified and addressed in previous studies [2]–[6]. As a security measure, we utilized onboard sensing capabilities of the robots to perform inter-robot co-observations and check for unusual behavior. These mutual observation establish a co-observation schedule alongside the path, ensuring that any attempts by a compromised robot to violate safety constraints (such as transgressing forbidden regions) would break the observation plan and be promptly detected.

A preliminary version of the paper was presented in [5], [6]. Extending the grid-world solution from [2] to continuous configuration spaces, we incorporate the co-observation planning as constraints in the alternating direction method of multipliers (ADMM)-based trajectory optimization solver to accomodate for more flexible objectives. In this paper, we incorporate additional reachability analysis during the planning phase, incorporating constraints based on sets of locations that agents could potentially reach, referred to as *reachability regions*, as a novel perspective to the existing literature. This approach enforces an empty intersection between forbidden regions and the reachability region during trajectory optimization, preventing undetected attacks if an adversary gains control of the robots. We utilize an ellipsoidal boundary to constrain the search space and formulate the ellipsoid as the reachability region constraint. We present a mathematical formulation of reachability regions compatible with the solver in [6] as a spatio-temporal constraint.

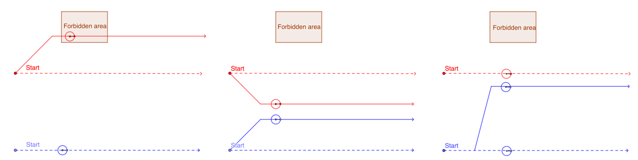
However, such plans are not guaranteed to exist, and the secured trajectory always come at the cost of overall system performance. As an extension of the previous works [5], [6], we also address the feasibility and optimality challenges. For a MRS with unsecured optimal trajectories (without security constraints), we introduce redundant robots and formed them into *sub-teams* with the original ones. These redundant robots are assigned the task of establishing additional co-observations, termed *cross-trajectory co-observations*, within and across different sub-teams. The proposed algorithm focuses on the movement plans for the additional robots, distinct from those dedicated to task objectives, indicating when they should stay with their current sub-team and when they should deviate to join another. This strategy allows sub-teams to preserve the optimal unsecured trajectories (as illustrated in Fig. 1) without requiring the entire *sub-team* to maintain close proximity to other teams during co-observation events. The cross-trajectory co-observation problem is transformed as a multi-agent path finding problem on roadmap (represented as directed graph) and solved as a network flow problem [7].

Related research. Trajectory planning problem of MRS systems remained as a subject of intense study for many decades, and optimization is a common approaches in such area. Optimization based approaches are customizable to a variable of constraints (e.g. speed limit, avoid obstacles) and task specifications (e.g. maximum surveillance coverage, minimal energy cost). In contrast to our work in Section II, many motion planning tasks require a non-convex constraint problem formulation, most contributors focused on convex problems, and only allow for a few types of pre-specified non-convex constraints through convexification [8] [9] [10]. Several optimization techniques like mixed integer programs with quadratic terms (MIQP) [11] and ADMM [12] have been used to reduce computational complexity, and to incorporate more complex non-convex constraints.

This project is supported by the National Science Foundation grant "CPS: Medium: Collaborative Research: Multiagent Physical Cognition and Control Synthesis Against Cyber Attacks" (Award number 1932162).

Ziqi Yang is with the Department of Systems Engineering, Boston University, Boston, MA 02215 USA (e-mail: zy259@bu.edu).

Roberto Tron is with the Faculty of Mechanical Engineering and Systems Engineering, Boston University, Boston, MA 02215 USA (e-mail: tron@bu.edu).



(a) Plan-deviation attack (b) Co-observation (c) Cross-trajectory co-observation

Fig. 1: 1a Attacker deviate the red robot into forbidden region.1b Red and blue robot tasked to co-observe each other during task resulting a new secured trajectory (solid lines) replacing the optimal ones (dashed line). 1c The blue team sends one robot to observe the red team (solid blue line) while having the rest of the robots following the optimal trajectory.

Graph-based search is another extensively explored approach in trajectory planning problems, such as formation control [13], [14] and Multi-Agent Path Finding (MAPF) problems [15]. In traditional MAPF formulations, environments are abstracted as graphs, with nodes representing positions and edges denoting possible transitions between positions and solved as a network flow problem [7], [16]. This formulation allows for the application of combinatorial network flow algorithms and linear program techniques, offering efficient and more flexible solutions to the planning problem.

Reachability analysis is essential for security and safety verification in cyber-physical systems (CPSs) [17], [18], often involving an over-approximation of reachable space to verify safety properties. Geometric methods like zonotopes and ellipsoids are commonly used to compactly enclose reachability sets [19]–[21]. For online safety assessments against cyber attacks, [22] computed reachable CPS states under attacks and compared them with a safe region based on state estimation. By incorporating an additional security measure that provides two secured states, the reachability region in our work is over-approximated using ellipsoids. This is inspired by the ellipsoidal *heuristic sampling domain* in [23] for the RRT* algorithm to simplify the sampling region between start and goal locations. Ellipsoids are also used in other path planning methods like Iterative Regional Inflation by Semidefinite programming (IRIS) [24], [25] and the Safe Flight Corridor (SFC) [26], [27]. Differs from reachability region that requires mapping all reachable states given several known states, the ellipsoids of IRIS and SFC focus on approximate the safe collision-free space rather than addressing security applications.

Paper contributions. Two main contributions have been presented.

- We present an innovative method to integrate reachability analysis into the ADMM-based optimal trajectory solver for multi-robot systems, preventing attackers from executing undetected attacks by simultaneously entering forbidden regions and adhering to co-observation schedules.
- We introduce additional robots to form *sub-teams* for both intra-sub-team and cross-sub-team co-observations. A new co-observation planning algorithm is formulated that can generate a resilient multi-robot trajectory with a co-observation plan that still preserve the optimal performance against arbitrary tasks. We also find the minimum redundant robots required for the security.

The rest of this paper is organized as follows. Section II introduces the ADMM-based optimal trajectory solver and introduce the security constraints. Section III introduces the enhanced security planning algorithm through cross-trajectory co-observation. Section IV concludes this article.

Notation. In this paper, we use non-bold symbols like to denote single-agent states (e.g. q_{ij}) and scalers, while bold symbols represent aggregated states involving single robot or multiple robots (e.g. $\mathbf{q}$).

II. Secured Multi-robot trajectory planning

We formulate the planning problem as an optimal trajectory optimization problem to minimize arbitrary smooth objective functions. We denote the trajectory as $\mathbf{q}_i = [q_{i0} \dots q_{iT}] \in \mathbb{R}^{m \times T}$ where $q_{ij} \in \mathbb{R}^m$ is the waypoint of agent i in a m dimensional state space and T is the time horizon. For a total of n_p robots, trajectories can be represented as an aggregated vector $\mathbf{q} = \mathrm{stack}(\mathbf{q}_1, \dots, \mathbf{q}_{n_p}) \in \mathbb{R}^{mn_p \times T}$, where $\mathrm{stack}(\cdot)$ denotes the vertical stacking operation. The overall goal is to minimize or maximize an objective function $\varPhi(\mathbf{q}) : \mathbb{R}^{nmT} \to \mathbb{R}$ under a set of nonlinear constraints described by a set Ω, which is given by the intersection of spatio-temporal sets given by traditional path planning constraintsand the security constraints (co-observation schedule, reachability analysis). Formally:

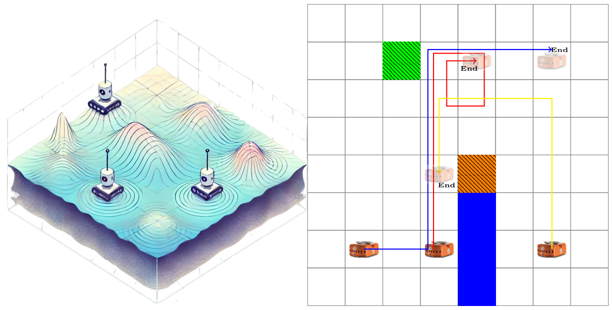
$$\begin{aligned} \min / \max \quad & \varPhi(\mathbf{q}) \\ \text{subject to} \quad & \mathbf{q} \in \Omega. \end{aligned} \tag{1}$$

To give a concrete example of the cost $\varPhi$ and the set Ω, we introduce a representative application that will be used for all the simulations throughout the paper.

Example 1. *Robots are tasked to navigate an unknown task space, collecting sensory data to reconstruct a field (see Figure 2a). The space is modeled as grid points, each with an associated value. The goal is to find paths that minimize uncertainty in reconstructing the field. Each grid point j is associated with a Kalman Filter (KF) [28] to track uncertainty through its covariance P_j, updated based on measurements taken by robots along the trajectory $\mathbf{q}$, where measurement quality, modeled by a Gaussian radial basis function, is higher near the robot. The optimization objective $\varPhi(\mathbf{q}) = \max_j P_j(\mathbf{q})$ is to minimize the maximum uncertainty (detailed in [6]).*

To incorporate the security against malicious deviations (through co-observation and reachability analysis) as constraints in the optimal trajectory planning problem, we define the security as:

Definition 1. *A multi-robot trajectory plan is secured against plan-deviation attacks if it ensures that any potential deviations to these forbidden regions will cause the corresponding robot to miss their next co-observation with other robots.*



(a) Map exploration task. (b) Grid world result.

Fig. 2: (a) Grid world example of a task planned in a 8×8 grid world. Blue grids are obstacles, orange and green grids are forbidden regions.

A. Preliminaries

1) Differentials: We define the differential of a map $f(x): \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ at a point x_0 as the unique matrix $\partial_x f \in \mathbb{R}^{d_1 \times d_2}$ such that

$$\frac{\mathrm{d}}{\mathrm{d}t} f\big(x(t)\big)\Big|_{t=0} = \partial_x f\big(x(0)\big)\dot{x}(0) \tag{2}$$

where $t \mapsto x(t) \in \mathbb{R}^{d_1}$ is a smooth parametric curve such that $x(0) = x$ with any arbitrary tangent $\dot{x}(0)$. For brevity we will use $\dot{f}$ for $\frac{\mathrm{d}}{\mathrm{d}t} f$ and $\partial_x f$ for $\frac{\partial f}{\partial x}$.

With a slight abuse of notation, we use the same notation $\partial_x f$ for the differential of a matrix-valued function with scalar arguments $f: \mathbb{R}^{d_1 \times d_3} \to \mathbb{R}^{d_2}$. Note that in this case (2) is still formally correct, although the RHS needs to be interpreted as applying $\partial_x f$ as a linear operator to $\dot{x}$.

2) Alternating Directions Method of Multipliers (ADMM): The basic idea of the ADMM-based solver introduced in [6] is to separate the constraints from the objective function using a different set of variables $\mathbf{z}$, and then solve separately in (1). More specifically, we rewrite the constraint $\mathbf{q} \in \Omega$ using an indicator function Θ, and include it in the objective function (details can be found in [6]). We allow $\mathbf{z} = D(\mathbf{q})$ to replicate an arbitrary function of the main variables $\mathbf{q}$ (instead of being an exact copy in general ADMM formulation), to transform constraint $\mathbf{q} \in \Omega$ to $D(\mathbf{q}) \in \mathcal{Z}$ to allow for an easier projection step in Equation (4b). In summary, we transform Equation (1) into

$$\begin{aligned} \max \quad & \Phi(\mathbf{q}) + \Theta(\mathbf{z}) \\ s.t. \quad & D(\mathbf{q}) - \mathbf{z} = 0 \end{aligned} \tag{3}$$

where $D(\mathbf{q}) = [D_1(\mathbf{q})^T, \ldots, D_l(\mathbf{q})^T]^T$ is a vertical concatenation of different functions for different constraints. This makes each constraint set $\mathcal{Z}_i$ independent and thus can be computed separately in later updating steps. $D_i(\mathbf{q})$ is chosen that the new constraint set $\mathcal{Z}_i$ becomes simple to compute which is illustrated in later secitons.

The update steps of the algorithm are then derived as [29]:

$$\mathbf{q}^{k+1} := \underset{\mathbf{q}}{\operatorname{argmin}}(\Phi(\mathbf{q}^k) + \frac{\rho}{2}\|D(\mathbf{q}) - \mathbf{z}^k + \mathbf{u}^k\|_2^2), \tag{4a}$$

$$\mathbf{z}^{k+1} := \Pi_{\mathcal{Z}}(D(\mathbf{q}^{k+1}) + \mathbf{u}^k), \tag{4b}$$

$$\mathbf{u}^{k+1} := \mathbf{u}^k + D(\mathbf{q}^{k+1}) - \mathbf{z}^{k+1}, \tag{4c}$$

where $\Pi_{\mathcal{Z}}$ is the new projection function to the modified constraint set $\mathcal{Z}$, $\mathbf{u}$ represents a scaled dual variable that, intuitively, accumulates the sum of primal residuals

$$\mathbf{r}^k = D(\mathbf{q}^{k+1}) - \mathbf{z}^{k+1}. \tag{5}$$

Checking the primal residuals alongside with the dual residuals

$$\mathbf{s}^k = -\rho(\mathbf{z}^k - \mathbf{z}^{k-1}) \tag{6}$$

after each iteration, the steps are reiterated until convergence when the primal and dual residuals are small, or divergence when primal and dual residual remains large after a fixed number of iterations.

Remark 1. *In practical application,* (4a) *is solved iteratively via nonlinear optimization solver like* fmincon*, while* (4b) *and* (4c) *have closed form solution.*

We now provide the functions $D(\mathbf{q})$, the sets $\mathcal{Z}$, and the corresponding projection operators $\Pi_{\mathcal{Z}}$ for security constraints including co-observation security constraints (Section II-B), and reachability constraints (Section II-E-Equation (34)). The latter are based on the definition of *ellipse-region-constraint* (Section II-C). Formulation of traditional path planning constraints like velocity constraints, convex obstacle constraints can be found in [6], thus are omitted in this paper.

B. Co-observation schedule constraint

The co-observation constraint ensures that two robots come into close proximity at scheduled times to observe each other's behavior. This constraint is represented as a relative distance requirement between the two robots at a specific time instant, ensuring they are within a defined radius to inspect each other or exchange data.

ADMM constraint 1 (Co-observation constraint)**.**

$$D(\mathbf{q}) = \overrightarrow{q_{aj}q_{bj}}, \tag{7}$$

$$\mathcal{Z} = \{\mathbf{z} \mid \|\mathbf{z}\| \leq d_{max}\}, \tag{8}$$

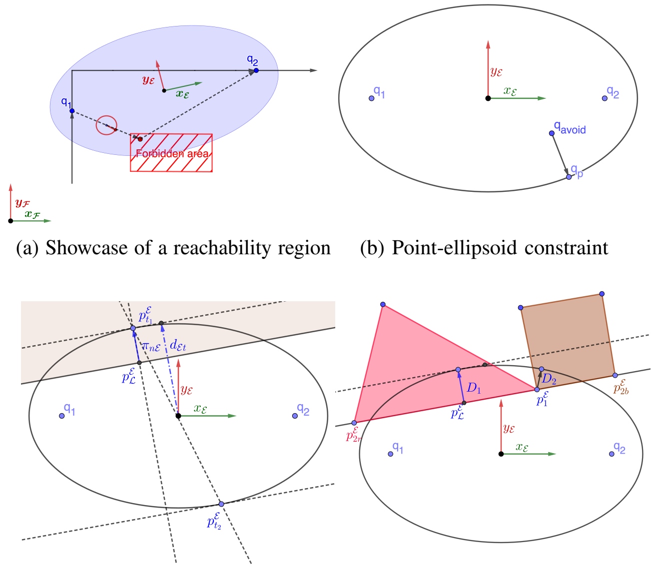
$$\Pi_{\mathcal{Z}}(z) = \begin{cases} d_{max}\frac{\mathbf{z}}{\|\mathbf{z}\|} & \text{if } \|\mathbf{z}\| > d_{max}, \\ \mathbf{z} & \text{otherwise}, \end{cases} \tag{9}$$

where a, b are the indices of the pair of agents required for a mutual inspection, d_{max} is the maximum distance between a pair of robots that the secure can be ensured through co-observation.

The locations q_{aj} and q_{bj} where the co-observation is performed are computed as part of the optimization.

C. Definition of ellipsoidal reachability regions

In this section, we define *ellipsoidal reachability regions* with respect to a pair of locations along a trajectory. We then use the ellipsoids to introduce various reachability constraints with respect to other geometric entities, such as points, lines, line segments, and polygons. These constraints, combined with the co-observation constraint 1, form the basis of the secured ADMM planning presented in Section II-A2.



(a) Showcase of a reachability region (b) Point-ellipsoid constraint

(c) Plane-ellipse constraint (d) Polygon-ellipse constraint

Fig. 3: (3a) Black line is the planned trajectory, q_1 and q_2 are two co-observed locations the robot are expected at given time t_1 and t_2, dashed lines show possible trajectory of a compromised robot during unobserved period. Axis of global frame $\mathcal{F}$ and canonical frame $\mathcal{F}_{\mathcal{E}}$ are shown as $x_{\mathcal{F}}, y_{\mathcal{F}}$ and $x_{\mathcal{E}}, y_{\mathcal{E}}$ respectively. (3b) For point-ellipsoid constraint, q_{avoid} is projected to the areas outside the ellipsoid to q_p. (3c) For plane-ellipsoid constraint, the projection is simplified to the point-ellipse constraint that projecting point p_L outside the ellipse to p_t. (3d) For convec-polygon-ellipsoid constraint, the projection either a plane-ellipse constraint D_1 (for red region) or a point-ellipse constraint D_2 (for brown region).

Definition 2. *Consider a robot i starting from q_1 at time t_1 and reaching q_2 at time t_2. The* reachability region *between t_1 and t_2 is defined as the set of points q' in the free configuration space for which there exists a kinematically feasible trajectory containing q'.*

For simplicity, in this paper we consider only a maximum velocity constraint v_{max}; the reachability region for q' can then be over-approximated by the following (the over-approximation is obtained by neglecting physical obstacles):

Definition 3. *The* reachability ellipsoid *$\mathcal{E}$ is defined as the region $\mathcal{E}(q_1, q_2, t_1, t_2) = \{\tilde{q} \in \mathbb{R}^n : d(q_1, \tilde{q}) + d(\tilde{q}, q_2) < 2a\}$, where $a = \frac{v_{max}}{2}(t_2 - t_1)$.*

This region is an ellipsoid because it represents the set of points q' whose sum of distances two *foci* q_1 and q_2 is less then $2a$, where a is the major radius of the ellipsoid.

The condition that a reachability ellipsoid $\mathcal{E}(q_1, q_2)$ does not intersect with any forbidden region is sufficient to secure the trajectory of the robot between q_1 and q_2 according to Definition 1: any deviation to a point p_o outside $\mathcal{E}(q_1, q_2)$ will cause the robot to miss a potentia observation at p_2, t_2. Definition 1 can then be restated as follows:

Definition 4. *A multi-robot trajectory is secured against plan-deviation attacks if there exist a co-observation plan such that the reachability region between each consecutive co-observation does not intersect with any forbidden regions.*

D. Transformation to canonical coordinates

To use the definition of reachability ellipsoid from the section above in a computational constraints for ADMM, we first apply a differentiable rigid body transformation to reposition the ellipse $\mathcal{E}$ from the global frame $\mathcal{F}$ to a canonical frame $\mathcal{F}_{\mathcal{E}}$, where the origin of $\mathcal{F}_{\mathcal{E}}$ is at the ellipsoid's center $o = \frac{1}{2}(q_1 + q_2)$, and the first axis of $\mathcal{F}_{\mathcal{E}}$ is aligned with the foci (see Figure 3a for an illustration). From this definition, a unit vector along the first axis of the ellipsoid in the frames $\mathcal{F}$ and $\mathcal{F}_{\mathcal{E}}$ is given by $\nu_{\mathcal{F}} = \frac{q_2 - q_1}{\|q_2 - q_1\|}$ and $\nu_{\mathcal{E}} = [1, 0, 0]^T$, respectively. The full transformation from coordinates $q^{\mathcal{E}} \in \mathcal{F}$ to $q \in \mathcal{F}$, and its inverse, are then parametrized by a rotation $R_{\mathcal{E}}^{\mathcal{F}}$ and a translation $o_{\mathcal{E}}^{\mathcal{F}}$ as (we drop the subscript and superscript from $R_{\mathcal{E}}^{\mathcal{F}}$ and $o_{\mathcal{E}}^{\mathcal{F}}$ to simplify the notation):

$$q = Rq^{\mathcal{E}} + o, \quad q^{\mathcal{E}} = R^{\mathrm{T}}(q - o). \tag{10}$$

where the rotation matrix $R = H(\nu_{\mathcal{F}}(q_1, q_2), \nu_{\mathcal{E}})$ is a *Householder rotation H* (a differentiable linear transformation describing the minimal rotation between the two vectors, see Appendix B for details). To simplify the notation, in the following, we will consider H to be a function of q_1, q_2 directly, i.e. $H(q_1, q_2)$. More details on (10) are given in Appendix A.

Reachability constraints are formulated with respect to different types of forbidden regions a point, a plane, a segment, and a convex polygon.

E. Point-ellipsoid reachability constraint

For a forbidden region in the shape of a single point q_{avoid} (Fig. 3b), the constraint is written as:

ADMM constraint 2 (Point-ellipsoid reachability constraint)**.**

$$D(\mathbf{q}) = \begin{cases} \pi_{p\mathcal{E}}(\mathbf{q}) - q_{avoid} & q_{avoid} \in \mathcal{E}, \\ 0 & \textit{otherwise}. \end{cases} \tag{11}$$

$$\mathcal{Z} = \{q \in \mathbb{R}^{nm} : \|D_p(\mathbf{q})\| = 0\}, \tag{12}$$

$$\Pi_{\mathcal{Z}}(\mathbf{z}) = 0, \tag{13}$$

where $\pi_{p\mathcal{E}}(q_{avoid}; q_1, q_2, a) = q_p$ is a projection function that returns a projected point q_p of q_{avoid} outside the ellipse, i.e., as the solution to

$$\pi_{p\mathcal{E}} = \underset{q_p \in \mathcal{E}^c}{\operatorname{argmin}} \quad \|q_{avoid} - q_p\|^2 \tag{14}$$

where $\mathcal{E}^c$ is the set complement of region $\mathcal{E}$.

For cases where $q_{avoid} \notin \mathcal{E}(q(t_2), q(t_1), r)$, $\pi_{p\mathcal{E}}(q_{avoid}) = q_{avoid}$. And for cases where $q_{avoid} \in \mathcal{E}(q_1, q_2, r)$, $D(q)$ needs to be projected to the boundary of the ellipse. In the canonical frame $\mathcal{F}_{\mathcal{E}}$, the projected point $q_p^{\mathcal{E}}$ can be written as:

$$q_p^{\mathcal{E}} = (I + sQ)^{-1} q_{avoid}^{\mathcal{E}} = S q_{avoid}^{\mathcal{E}} \tag{15}$$

where s can be solved as the root of the level set (39):

$$q_p^{\mathcal{E}\,\mathrm{T}} Q q_p^{\mathcal{E}} - 1 = q_{avoid}^{\mathcal{E}\,\mathrm{T}} Q'(s) q_{avoid}^{\mathcal{E}} - 1 = 0 \tag{16}$$

where

$$Q'(s) = S^{\mathrm{T}} Q S = \operatorname{diag}\left(\frac{a^2}{(s+a^2)^2}, \frac{b^2}{(s+b^2)^2}, \frac{b^2}{(s+b^2)^2}\right) \tag{17}$$

Detailed methods for computing s can be found in [5], [6], [30].

The point-to-ellipse projection function in $\mathcal{F}$ is then:

$$\begin{aligned}\pi_{p\mathcal{E}}(q) &= R^{-1}(q(t_1), q(t_2)) q_p^{\mathcal{E}} + o \\ &= R^{-1}(q(t_1), q(t_2)) S q_{avoid}^{\mathcal{E}} + o \\ &= R^{-1} S R (q_{avoid} - o) + o \end{aligned} \tag{18}$$

In our derivations, we consider only the 3-D case ($m = 3$); for the 2-D case, let $P = \begin{bmatrix} I & 0 \end{bmatrix} \in \mathbb{R}^{2\times 3}$: then $\pi_{p\mathcal{E}}^{2\mathrm{D}} = P\pi_{p\mathcal{E}}^{3\mathrm{D}}(P^{\mathrm{T}} q_{avoid}; P^{\mathrm{T}} q_1, P^{\mathrm{T}} q_2, a)$.

Proposition 1. *The differential of the projection operator $\pi_{p\mathcal{E}}(q_{avoid}; q_1, q_2, a)$ with respect to the foci q_1, q_2 is given by the following (using q as a shorthand notation for $q_{avoid}^{\mathcal{E}}$)*

$$\begin{aligned}\partial_{\left[\begin{smallmatrix} q_1 \\ q_2 \end{smallmatrix}\right]} \pi_{p\mathcal{E}} &= -2H[SH(q-o)]_\times U \\ &+ \big((q^{\mathrm{T}} \partial_s Q' q)^{-1} H^{-1} Q' q q^{\mathrm{T}} (4Q' H [q-o]_\times U \\ &+ 2Q' H \partial_q o - \partial_b Q' q q \partial_q b) - s H^{-1} S^2 \partial_b Q q \partial_q b\big) \\ &- 2H^{-1} S H [q-o]_\times U + (H^{-1} S H - I) \partial_q o \end{aligned} \tag{19}$$

Proof. See Appendix C. □

The differential of D_p is the same as the one for $\pi_{p\mathcal{E}}$.

F. Plane-ellipsoid reachability constraint

For a hyperplane shaped forbidden region $\mathcal{L}(q) = \{q \in \mathbb{R}^m : \mathbf{n}^{\mathrm{T}} q = d\}$ (Fig. 3c), the reachability constraint is $\mathcal{L} \cap \mathcal{E}(q_1, q_2, a) = \emptyset$. When transformed into the canonical frame, the hyperplane can be written as $\mathcal{L}^{\mathcal{E}}(q^{\mathcal{E}}) = \{q^{\mathcal{E}} \in \mathbb{R}^m : \mathbf{n}_{\mathcal{E}}^{\mathrm{T}} q^{\mathcal{E}} = d_{\mathcal{E}}\}$, with $\mathbf{n}_{\mathcal{E}} = H(q_1, q_2)\mathbf{n}, \quad d_{\mathcal{E}} = -\mathbf{n}^{\mathrm{T}} o + d$.

For every $\mathcal{L}^{\mathcal{E}}(q^{\mathcal{E}})$, there exist two planes that are both parallel to $\mathcal{L}$ and tangential to the ellipse (i.e. resulting in a unique intersection point Fig. 3c), $\mathcal{L}_1^{\mathcal{E}} = \{q^{\mathcal{E}} \in \mathbb{R}^m : \mathbf{n}_{\mathcal{E}}^{\mathrm{T}} q^{\mathcal{E}} = d_{\mathcal{E}t}\}$ and $\mathcal{L}_2^{\mathcal{E}} = \{q^{\mathcal{E}} \in \mathbb{R}^m : \mathbf{n}_{\mathcal{E}}^{\mathrm{T}} q^{\mathcal{E}} = -d_{\mathcal{E}t}\}$. The intersection point can be written as:

$$p_{t_1}^{\mathcal{E}} = \frac{d_{\mathcal{E}t} Q^{-1} n_{\mathcal{E}}}{n_{\mathcal{E}}^{\mathrm{T}} Q^{-1} n_{\mathcal{E}}} = \frac{Q^{-1} n_{\mathcal{E}}}{d_{\mathcal{E}t}}, \quad p_{t_2}^{\mathcal{E}} = -p_{t_1}^{\mathcal{E}}, \tag{20}$$

where $d_{\mathcal{E}t} = \sqrt{n_{\mathcal{E}}^{\mathrm{T}} Q^{-1} n_{\mathcal{E}}}$; intuitively, $d_{\mathcal{E}t}$ can be thought as a distance between the tangent plane $\mathcal{L}_1^{\mathcal{E}}$ (or $\mathcal{L}_2^{\mathcal{E}}$) and the origin (i.e., the center of the ellipse $\mathcal{E}$). The concept of *tangent interpolation point*s is introduced to characterize the relationship between the plane and the ellipsoid.

Definition 5. *The* tangent interpolation point *$p_{\mathcal{L}}^{\mathcal{E}} \in \mathcal{L}$ between the plane $\mathcal{L}$ and the ellipsoid $\mathcal{E}$ is defined on the plane by*

$$p_{\mathcal{L}}^{\mathcal{E}} = \frac{d_{\mathcal{E}} Q^{-1} n_{\mathcal{E}}}{n_{\mathcal{E}}^{\mathrm{T}} Q^{-1} n_{\mathcal{E}}}. \tag{21}$$

Intuitively, the point $p_{\mathcal{L}}^{\mathcal{E}}$ is the point on $\mathcal{L}$ which is closest to either $p_1^{\mathcal{E}}$ or $p_2^{\mathcal{E}}$. Note that when $d_{\mathcal{E}} = d_{\mathcal{E}t}$ or $d_{\mathcal{E}} = -d_{\mathcal{E}t}$, $p_{\mathcal{L}}^{\mathcal{E}} = p_{t_1}^{\mathcal{E}}$ or $p_{\mathcal{L}}^{\mathcal{E}} = p_{t_2}^{\mathcal{E}}$, respectively. When $d_{\mathcal{E}} \in [-d_{\mathcal{E}t}, d_{\mathcal{E}t}]$, the plane $\mathcal{L}$ and the ellipsoid $\mathcal{E}$ have at least one intersection, thus violating our desired reachability constraint.

With these definitions, the constraint can be written as:

ADMM constraint 3 (Plane-ellipsoid reachability constraint)**.**

$$D(\mathbf{q}) = H^{-1}(q_1, q_2)\pi_{\mathbf{n}_{\mathcal{E}}}^{\mathcal{E}}(\mathbf{q}) + o, \tag{22}$$

$$\mathcal{Z} = \{\mathbf{q} \in \mathbb{R}^{nm} : \|D_{\mathbf{n}}(\mathbf{q})\| = 0\}, \tag{23}$$

$$\Pi_{\mathcal{Z}}(\mathbf{z}) = \vec{0}, \tag{24}$$

where $\pi_{\mathbf{n}_{\mathcal{E}}}^{\mathcal{E}}(q)$ is the projection operator defined as:

$$\pi_{\mathbf{n}_{\mathcal{E}}}^{\mathcal{E}}(\mathbf{q}) = \begin{cases} p_{t_1}^{\mathcal{E}} - p_{\mathcal{L}}^{\mathcal{E}} & \text{if } d_{\mathcal{E}} \in [0, d_{\mathcal{E}t}], \\ p_{t_2}^{\mathcal{E}} - p_{\mathcal{L}}^{\mathcal{E}} & \text{if } d_{\mathcal{E}} \in [-d_{\mathcal{E}t}, 0), \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Proposition 2. *The differential of the projection function $\pi_{\mathbf{n}\mathcal{E}}^{\mathcal{E}}(\mathbf{q})$ with respect to the foci q_1 and q_2 is given by:*

$$\partial_q \pi_{\mathbf{n}\mathcal{E}}^{\mathcal{E}}(q) = \begin{cases} \partial_q p_{t1}^{\mathcal{E}} - \partial_q p_{\mathcal{L}}^{\mathcal{E}} & d_{\mathcal{E}} \in [0, d_{\mathcal{E}t}], \\ \partial_q p_{t2}^{\mathcal{E}} - \partial_q p_{\mathcal{L}}^{\mathcal{E}} & d \in [-d_{\mathcal{E}t}, 0), \\ 0 & \textit{otherwise.} \end{cases} \tag{26}$$

where

$$\begin{aligned}\partial_q p_{\mathcal{L}} = &\Big(-\frac{d_{\mathcal{E}t} n^{\mathrm{T}} \partial_q o - 2 d_{\mathcal{E}} \partial_q d_{\mathcal{E}t}}{d_{\mathcal{E}t}^3}\Big) Q^{-1} n_{\mathcal{E}} \\ &+ \frac{d_{\mathcal{E}} \partial_b Q^{-1} n_{\mathcal{E}} \partial_q b - 2 d_{\mathcal{E}} Q^{-1} H [n]_\times U}{d_{\mathcal{E}t}^2}, \end{aligned} \tag{27}$$

$$\partial_q p_1 = -\frac{Q^{-1} n_{\mathcal{E}} \partial_q d_{\mathcal{E}t}}{d_{\mathcal{E}t}^2} + \frac{\partial_b Q^{-1} n_{\mathcal{E}} \partial_q b - 2Q^{-1} H [n]_\times U}{d_{\mathcal{E}t}}. \tag{28}$$

Proof. See Appendix D. □

The differential of (22) can be written as:

$$\partial_q D_{\mathbf{n}\mathcal{E}} = -2H[\Pi^{\mathcal{E}}{}_{\mathbf{n}\mathcal{E}}]_\times M + H^{-1} \partial_q \Pi^{\mathcal{E}}{}_{\mathbf{n}\mathcal{E}}. \tag{29}$$

G. Line-segment-ellipse reachability constraint

As an intermediate step to consider polygon shaped forbidden regions, the reachability constraint for hyperplane segments is studied. For hyperplane $\mathcal{L}^{\mathcal{E}}(q^{\mathcal{E}}) = \{q^{\mathcal{E}} \in \mathbb{R}^m : \mathbf{n}_{\mathcal{E}}^{\mathrm{T}} q^{\mathcal{E}} = d_{\mathcal{E}}\}$ with endpoints $p_1^{\mathcal{E}}$ and $p_2^{\mathcal{E}}$; this segment is defined as:

$$\begin{bmatrix} (p_1^{\mathcal{E}} - p_2^{\mathcal{E}})^{\mathrm{T}} \\ (p_2^{\mathcal{E}} - p_1^{\mathcal{E}})^{\mathrm{T}} \end{bmatrix} p^{\mathcal{E}} \leq \begin{bmatrix} p_2^{\mathcal{E}\,\mathrm{T}} \\ -p_1^{\mathcal{E}\,\mathrm{T}} \end{bmatrix} (p_1^{\mathcal{E}} - p_2^{\mathcal{E}}), \quad \mathbf{n}_{\mathcal{E}}^{\mathrm{T}} q^{\mathcal{E}} = d_{\mathcal{E}}. \tag{30}$$

When the *tangent interpolation point* $p_{\mathcal{L}}^{\mathcal{E}}$ stays within the segment (i.e. red region in Fig. 3d), the constraint is a plane-ellipse constraint in Section II-F. Otherwise (i.e. brown region in Fig. 3d), the constraint is a point-ellipse constraint in Section II-E.

ADMM constraint 4 (Line-segment-ellipsoid reachability constraint)**.**

$$D(\mathbf{q}) = \begin{cases} D_{p_1}(\mathbf{q}) & (p_1^{\mathcal{E}} - p_2^{\mathcal{E}})^{\mathrm{T}}(p_{\mathcal{L}}^{\mathcal{E}} - p_2^{\mathcal{E}}) < 0 \\ D_{p_2}(\mathbf{q}) & (p_2^{\mathcal{E}} - p_1^{\mathcal{E}})^{\mathrm{T}}(p_{\mathcal{L}}^{\mathcal{E}} - p_1^{\mathcal{E}}) < 0 \\ D_{p_{\mathcal{L}}^{\mathcal{E}}}(\mathbf{q}) & \textit{otherwise} \end{cases} \tag{31}$$

$$\mathcal{Z} = \{\mathbf{q} \in \mathbb{R}^{nm} : \|D(\mathbf{q})\| = 0\}, \tag{32}$$

$$\Pi_{\mathcal{Z}}(\mathbf{z}) = 0, \tag{33}$$

where D_{p_1} and D_{p_2} are the point-ellipsoid constraint projection function (11) with respect to $p_1^{\mathcal{E}}$ and $p_2^{\mathcal{E}}$, and $D_{p_{\mathcal{L}}^{\mathcal{E}}}$ is the plane-ellipsoid constraint (22). with respect to frame $\mathcal{L}^{\mathcal{E}}$.

H. Convex-polygon-ellipse reachability constraint

For reachability constraints with respect to a convex polygon, the first step is to keep all segments that defines the hyperplane outside the reachability ellipsoid. Similar to (21), we define

ADMM constraint 5 (Convex-polygon-ellipsoid reachability constraint)**.**

$$D(\mathbf{q}) = \begin{bmatrix} D_{seg1}(\mathbf{q}) \\ D_{seg2}(\mathbf{q}) \\ \vdots \end{bmatrix} \tag{34}$$

$$\mathcal{Z} = \{\mathbf{q} \in \mathbb{R}^{nm} : \|D(\mathbf{q})\| = 0\}, \tag{35}$$

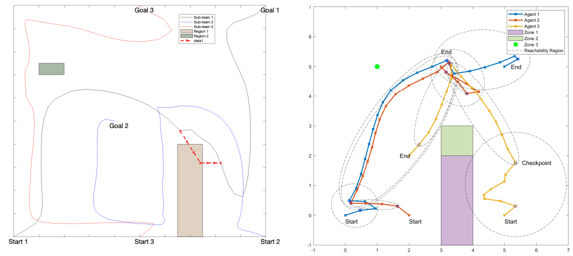
$$\Pi_{\mathcal{Z}}(\mathbf{z}) = 0, \tag{36}$$

where D_{seg} are the constraint functions for all line segments used to define the convex polygon region. This constraint needs to be supplemented with a convex obstacle constraint for the polygon (i.e. keep foci waypoints outside the region, introduced in [6]) to prevent cases where the ellipse is a subset of the region.

I. Secured planning results and limitation

We test the newly introduced security constraints with Example 1, illustrated in Fig. 4 in both simulations and on an experimental testbed, involving a 3-robot system assigned to explore a $10m \times 10m$ region with one obstacle, *Zone 1*, and two forbidden zones, *Zone 2* and *Zone 3*. The maximum velocity constraint $v_{max} = 0.5m/dt$, time horizon $T = 20$.

We employ the attack-proof MAPF(APMAPF) solver [2] on a 8 by 8 grid world with similar setup to generate a MAPF plan with a co-observation schedule in a grid-world application (Fig.2b). This result is transformed to a continuous configuration space and serves as the initial trajectory input for the ADMM solver with additional task function targeting the map exploration task. Co-observation schedules are set up using the APMAPF algorithm for two forbidden regions. Reachability constraints are added to ensure an empty intersection between all robots' reachability regions during co-observations and the forbidden regions. It is important to emphasize that an APMAPF solution does not guarantee existence, nor does it ensure a successful transition to the continuous configuration space. In this cases, while an APMAPF solution may be found, the secured, attack-proof solution becomes infeasible in the continuous setting because the robots' mobility is no longer restricted to adjacent grids. Additional security measures, such as stationary security cameras or surveillance robots, need to be incorporated. For instance, in this scenario, we deploy a security camera as additional checkpoint to observe agent 3 at time 8 to ensure security.



(a) Continuous world trajectory (b) Securd result

Fig. 4: Trajectory design of a map exploration task forthree-robot system based on result in grid world result Figure 2b. Zone 1 is obstacle, Zone 2 and Zone 3 are safe zones. Figure 4a shows the unsecured trajectory design in continuous world optimized with respect to a map exploration task where Zone 1 and 2 are considered together as one obstacle. A potential path that could break the security by entering region 1 is highlighted in dashline. Figure 4b shows the secured trajectory with the incorporation of co-observation and reachability constraints.

The simulation result, shown in Fig. 4b, displays reachability regions as black ellipsoids, demonstrating empty intersections with Zones 2 and 3. Explicit constraints between reachability regions and obstacles are not activated, assuming basic obstacle avoidance capabilities in robots. The intersections between obstacles and ellipsoids, as observed between agent 3 and Zone 1, are deemed tolerable. All constraints are satisfied, and agents have effectively spread across the map for optimal exploration tasks.

1) Limitations: Our solution demonstrates the potential of planning with reachability and co-observation to enhance the security of multi-agent systems.However, two primary challenges need to be addressed. Firstly, achieving a co-observation and reachability-secured plan is not always feasible, particularly when obstacles or restricted regions separate the robots, preventing them from establishing co-observation schedules or finding reachability-secured paths. For example, agent 3 in Fig. 4b requires additional security measures to create secured reachability areas. Secondly, security requirements can impact overall system performance, as illustrated by the comparison between Fig. 4a and Fig. 4b. The introduction of security constraints resulted in the top left corner remaining unexplored by any robots. This trade-off between security and system performance is particularly significant as system performance is a key factor in the decision of multi-agent system deployments. These challenges are further addressed in Section III, ensuring the effective integration of reachability and co-observation in securing multi-agent systems.

III. Cross-trajectory co-observation planning

To address the feasibility and performance trade-offs, we propose to form *sub-teams* on each route, and setup additional co-observations both within the sub-team and across different sub-teams. These *cross-trajectory co-observations* allow sub-teams to obtain trajectories that are closer to the optimal (as shown in Fig. 1), because they do not require the entire *sub-team* to meet with other teams for co-observations, thus providing more flexibility.

A. Problem overview

We start with an unsecured MRS trajectory with n_p routes $\{q_p\}_{p=1}^{N_p}$ (the ADMM-based planner in Section II without security constraints is used, but other planners are applicable). Here, the state $q_p(t), p \in \{\mathcal{I}_0, \dots, \mathcal{I}_p\}$ represents the reference position of the p-th sub-team at time $t \in \{0, \dots, T\}$. Introducing redundant robots into the MRS, we assume a total of $n > n_p$ robots are available and organized into *sub-teams* through a time-varying partition $\mathcal{I}(t) = \cup_p \mathcal{I}_p(t)$ where robots in each *sub-team* $\mathcal{I}_p$ share the same nominal trajectory. To ensure the fulfillment of essential tasks, at least one robot is assigned to adhere to the reference trajectory. The remaining redundant $n_p - n$ robots are strategically utilized to enhance security. These robots focus on co-observations either within their respective sub-teams, or when necessary, deviate and join another sub-team to provide necessary co-observations. The objective of this problem is to formulate a strategy for this new co-observation strategy, named *cross-trajectory co-observation* that the resulting MRS plan meet the security in Definition 4 while minimizing the number of additional robots required.

Our strategy is based on the following concept:

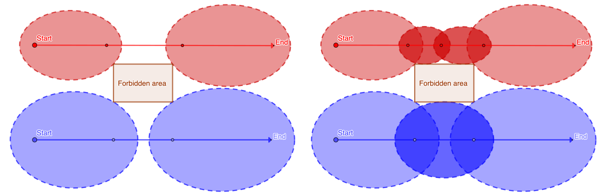
Definition 6. *The i-th* checkpoint $v_{pi} = (q_p, t_i)$ *for sub-team p is defined as a location $q_{pi} = q_p(t_i)$ and time t_i at which a robot is co-observed by its teammates, and where it can leave or join a new team.*

For simplicity, let $\mathcal{I}_{v_i}$ denote the sub-team to which v_i belongs. To ensure the security of the reference trajectory, the reachability region between consecutive checkpoints must avoid intersections with forbidden regions. This requirement can be formally stated as:

Remark 2. *A set of checkpoints $V_p = \{v_{p0}, \dots, v_{pT}\}$ (arranged in ascending order of $t_{v_{pi}}$) can secure the reference trajectory for sub-team p, if $\mathcal{E}(q_{v_{pi}}, q_{v_{p(i+1)}}, t_{v_{pi}}, t_{v_{p(i+1)}}) \cap F = \emptyset$ for every i, where F is the union of all forbidden regions.*

We model the planned trajectory $\{q_p\}_{p=1}^{N_p}$ as a directed *checkpoint graph* $Gq = (V_q, E_q)$, where $V_q = V_p \cup V_c$ are union of checkpoints V_p and connection verticies V_c which is introduced later in Section III, $E_q = E_t \cup E_c$ connect pairs of vertices if possible paths exist. Inspired by [7], additional robots in the sub-team are treated as flows in the checkpoint graph. This transforms the co-observation planning problem into a network multi-flow problem, solvable with general Mixed-Integer Linear Programming techniques.

This section presents the two components of our approach: constructing the checkpoint graph based on unsecured multi-robot trajectories, and the formulation and solution of the network multi-flow problem.



(a) (b)

Fig. 5: Checkpoint generation example

B. Rapidly-exploring Random Trees

To find edges between different nominal trajectories, we use the RRT* [31] algorithm to find paths from a waypoint on one trajectory to multiple destination points (i.e., reference trajectories of other sub-teams, in our method). As a optimal path planning algorithm, RRT* returns the shortest paths between an initial location and points in the free configuration space, organized as a tree. We assume that the generated paths can be travelled in both directions (this is used later in our analysis). Key functions from RRT* that are also used during constructions of the checkpoint graph are

Cost(v) This function assigns a non-negative cost (total travel distance in our application) to the unique path from the initial position to v .

Parent(v) This is a function that maps the vertex v to $v' \in V$ such that $(v', v) \in E$.

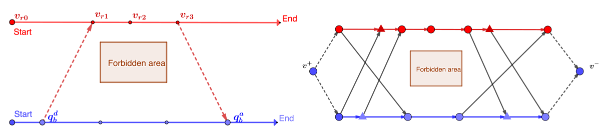
Our objective here is to ascertain the existence of feasible paths instead of optimize specific tasks. While the ADMM based solver Section II presented earlier offers a broader range of constraint handling, RRT*'s efficient exploration of the solution space, coupled with its ability to incorporate obstacle constraints, makes it a fitting choice for building the checkpoint graph.

C. Checkpoint graph construction

In this section, we define and search for security checkpoints and how to use RRT* to construct the checkpoint graph G_q. A heuristic approach is provided to locate the checkpoints on given trajectories (an optimal solution would likely be NP-hard, while the approach below works well enough for our purpose). We begin by adding the start and end locations $(q_p(t_0), t_0)$, $(q_p(t_2), t_2)$ to V_q, with $t_0 = 0$, $t_2 = T$. Then, we search for the largest $t_1 \in \{t_0 + 1, \dots, t_2\}$ such that the reachability ellipsoid between $q_p(t_0)$ and $q_p(t_1)$ has no overlap with any of the forbidden areas, i.e. $\mathcal{E}(q_p(t_0), q_p(t_1), t_0, t_1) \cap F = \emptyset$. Then $(q_p(t_1), t_1)$ is added to V_q. Simultaneously, we search backward to find $q_p(t_3)$ with the smallest $t_3 \in \{t_1, \dots, t_2 - 1\}$ that meets the reachability requirement $\mathcal{E}(q_p(t_2), q_p(t_3), t_2, t_3) \cap F = \emptyset$. Then $(q_p(t_3), t_3)$ is added to V_q. Afterwards, we set $t_0 \leftarrow t_1$, $t_2 \leftarrow t_3$ and repeat the same process. The search is stopped when $\mathcal{E}(q_p(t_0), q_p(t_2), t_0, t_2) \cap F = \emptyset$ or $t_0 > t_2$. A toy example is shown in Fig. 5.

1) Cross-trajectory edges: To enable co-observation cross different sub-team at checkpoints, we search for available connection paths (*cross-trajectory edges*) between checkpoints on different trajectories, allowing robots deviate from one sub-team to perform co-observation with a different sub-team.

Cross-trajectory edges $E_c = (v_1, v_2)$ define viable paths between two reference trajectories, where $\mathcal{I}_{v_1} \neq \mathcal{I}_{v_2}$ and at least one of v_1 and v_2 correspond to a security checkpoint $\cup_p V_p$. The cross-trajectory edges must also adhere to reachability constraints $\mathcal{E}(q_{v_1}, q_{v_2}, t_{v_1}, t_{v_2}) \cap F = \emptyset$ to ensure that no deviations into forbidden regions can occur during trajectory switches.



(a) Cross-trajectory edge generation (b) Full checkpoint graph

Fig. 6: (6a) Latest departure node q_b^d found for v_{r1} and earliest arrival node q_b^a found for v_{r3}. (6b) Eexample of a full checkpoint graph where round vertices are checkpoint generated through the heuristic search and triangle vertices are arrival and departure nodes added with the cross-trajectory edges. Virtual source v^+ and sink v^- are added to be used later in planning problem.

We first find trees of feasible paths between each security checkpoint and all other trajectories using position information alone (ignoring, for the moment, any timing constraint). More precisely, through $\texttt{RRT}^\star$, we find all feasible, quasi-optimal paths between one security checkpoint location $q_{v_p}(t_{v_p})$ where $v_p \in V_p$ for sub-team $\mathcal{I}_p$, and all the waypoints $\{q_r(t_{r_i})\}$ on any reference trajectory for a different sub-team $\mathcal{I}_r$. Then, to prune these trees, we consider the time needed to physically travel from one trajectory to the other while meeting other robots at the two endpoints. This is done by calculating the minimal travel time $t_{\text{path}} = \texttt{Cost}(q)/v_{max}$ for a robot to traverse each path. Two types of connecting nodes can be found.

Arrival nodes are waypoints on $\{q_r\}$ where robots from sub-team $\mathcal{I}_p$, deviating at v_p, can meet with sub-team $\mathcal{I}_r$ at $(q_r(t_{r_i}), t_{r_i})$. For these, $\texttt{RRT}^\star$ must have found a path from v_p to $q_r(t_r^a)$ with $t_p + t_{\text{path}} < t_{r_i}$, and $\mathcal{E}(q_r(t_{r_i}), q_p, t_{r_i}, t_p) \cap F = \emptyset$.
For each trajectory $r \neq p$, we define the *earliest arrival node* from v_p to $v_{ea} = ({}_r(t_r^a), t_r^a)$ as the arrival node characterized by the minimum t_r^a discovered.

Departure nodes are the waypoints on $\{q_r\}$ such that a robot from sub-team $\mathcal{I}_r$, deviating from $(q_r(t_{r_i}), t_{r_i})$, can meet with robots in sub-team $\mathcal{I}_p$ at v_p. For these nodes, $\texttt{RRT}^\star$ must find a path from v_p to $q_r(t_{r_i})$ for v_p if $t_p > t_{r_i} + t_{\text{path}}$ and $\mathcal{E}(q_p, q_r(t_{r_i}), t_p, t_{r_i}) \cap F = \emptyset$.
For each trajectory $r \neq p$, we define the *latest departure node* $v_{ld} = (q_r(t_r^d), t_r^d)$ as the departure node characterized by the maximum t_r^d discovered.

For each $v_p \in V_p$, all the latest departure and earliest arrival nodes are added as vertices $V_q = V_p \cup \{v_{ea}, v_{ld}\}$, and the corresponding paths are added as cross-trajectory edges to E_q. Examples are shown in Fig. 6a.

2) In-trajectory edges: We arrange all $V_q = \cup_p \{v_p^0, \ldots, v_p^T\}$ found in Sections III-C and III-C1 in ascending order of $t_{v_p^i}$. We then add to E_q the *in-trajectory edges* $\{(v_p^i \rightarrow v_p^{i+1})\}$ obtained by connecting all consecutive checkpoints $v_p^i \in V_q$ with the checkpoints v_p^{i+1} that follow them in their original trajectories. Examples are shown in Fig. 6b.

D. Co-observation planning problem

In this section, we formulate the cross-trajectory planning problem as a network multi-flow problem, and solve it using mixed-integer linear programs (MILP). We assume that n_p robots are dedicated (one in each sub-team) to follow the reference trajectory (named *reference robots*). The goal is to plan the routes of the $n - n_p$ additional *cross-trajectory robots* dedicated to cross-trajectory co-observations, and potentially minimize the number of cross-trajectory robots needed.

Remark 3. *Note that we assign fixed roles to robots for convenience in explaining the multi-flow formulation. In practice, after a cross-trajectory robot joins a team, it is considered interchangable, and could switch roles with the reference robot of that trajectory.*

To formulate the problem as a network multi-flow problem, we augment the checkpoint graph G_q to a flow graph $G = (V, E)$. The vertices of the new graph G are defined as $V = V_q \cup \{v^+, v^-\}$, where v^+ is a *virtual source* node, and v^- is a *virtual sink* node. We add directed edges from v^+ to all the start vertices, and from all end vertices to v^-, with v_p^0 and v_p^T representing the start and end vertices of sub-team $\mathcal{I}_p$. The edges of the new graph are defined as $E = E_q \cup \{(v_+, v_p^0)\}_p \cup \{(v_p^T, v_-)\}_p$.

The path of a robot k all starts from v^+ and ends at v^-, and are represented as a flow vector $\mathbf{f}^k = \{f_{ij}^k\}$, where $f_{ij}^k \in \{1, 0\}$ is an indicator variable representing whether robot k's path contains the edge $v_i \rightarrow v_j$. The planning problem can be formulated as a path cover problem on G_q, i.e., as finding a set of paths $F = [\mathbf{f}_1, \ldots, \mathbf{f}_\mathcal{K}]$ for cross-trajectory robots such that every checkpoint in $\cup_p V_p$ is included in at least one path in F (to ensure co-observation at every checkpoint as required by Definition 4).

Technically, we can always create a trivial schedule that involves only co-observations between members of the same team; this, however, would make the solution more vulnerable in the case where multiple agents are compromised in the same team. While in this paper we explicitly consider only the single attacker scenario, multi-attacks can be potentially handled by taking advantage of the *decentralized blocklist protocol* introduced in [3]. For this reason, we setup the methods presented below to always prefer *cross-trajectory co-observation* when feasible.

Finally, edges from the virtual source and to the virtual sink should have zero cost, to allow robots to automatically get assigned to the starting point that is most convenient for the overall solution (lower cost when taking cross-trajectory edges). These requirements are achieved with the weights for edges

$(v_i, v_j) \in E$ defined as:

$$w_{i,j} = \begin{cases} -w_t & \mathcal{I}_{v_i} = \mathcal{I}_{v_j}, (v_i, v_j) \in E_q \\ w_c & \mathcal{I}_{v_i} \neq \mathcal{I}_{v_j}, (v_i, v_j) \in E_q \\ 0 & (v_i, v_j) \in E / E_q \end{cases} \tag{37}$$

where $w_c > w_t$.

With the formulation, the planning problem is written as an optimization problem, where the optimization cost balances between the co-observation performance and total number of flows (cross-trajectory robots) needed:

$$\min_F \sum_k^{\mathcal{K}} \sum_{(+i) \in E} f_{+i}^k - \rho \sum_k^{\mathcal{K}} \sum_{(ij) \in E} w_{ij} f_{ij}^k \tag{38a}$$

$$s.t. \sum_{\{h:(hi) \in E\}} f_{hi}^k = \sum_{\{j:(ij) \in E\}} f_{ij}^k, \forall k, \forall v \in V_q / \{v^+, v^-\} \tag{38b}$$

$$\sum_k^{\mathcal{K}} \sum_{\{i:(ij) \in E\}} f_{ij}^k \geq 1, \forall v_j \in \{V_p^s\} \tag{38c}$$

$$f_{ij}^k \in \{0, 1\} \, \forall (ij) \in E \tag{38d}$$

where, for convenience, we used (ij) to represent the edge (v_i, v_j), and $(+i)$ to represent the edge (v^+, v_i).

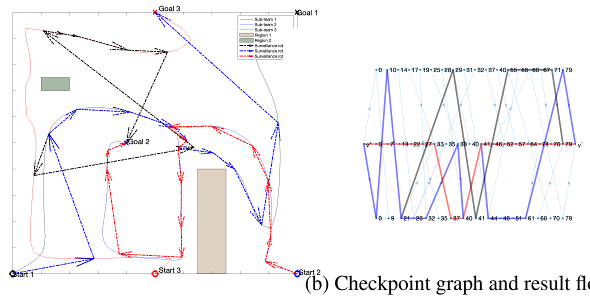
The first term in the cost (38a) represents the total number of robot used, and the second term represents the overall co-observation performance (defined as the total number of cross-trajectory edges taken by all flows beyond the regular trajectory edges); the constant ρ is a manually selected penalty parameter to balance between the two terms. (38b) is the flow conservation constraint to ensures that the amount of flow entering and leaving a given node v is equal (except for v^+ and v^-). (38c) is the flow coverage constraints to ensures that all security checkpoints $\{V_p^s\}$ have been visited co-observed. The security graph G_q is acyclic, making this problem in complexity class P, thus, can be solved in polynominal time [32].

E. Co-observation performance

Notice that problem (38) is guaranteed to have a solution for $\mathcal{K} = N_p$ where all $\mathcal{K}$ robots follows the reference trajectory ($f_{ij}^k = 1, \forall \mathcal{I}_{v_i} = \mathcal{I}_{v_j} = k$). It is possible for the resulting flows to have a subset of flows $F_e \in F$ that is empty, i.e. $f_{ij} = 0, \forall (v_i, v_j) \in E$, $f \in F_e$; these flow will not increase the cost and can be discarded from the solution.

The constant ρ selects the trade-off between the number of surveillance robots and security performance. An increase in robots generally enhancing security via cross-trajectory co-observations; this also increases the complexity of the coordination across robots. In order to identify the minimum number of robots necessary, we propose an iterative approach where we start with $\mathcal{K} = 1$ and gradually increases it until F contains an empty flow, indicating the point where further robot additions do not improve performance.

Remark 4. *The value of row ρ is upper bounded such that the second term for each single flow in* (38a) *is always smaller than one, i.e., $\rho \sum_{(ij) \in E} w_{ij} f_{ij}^k \leq 1$. Otherwise, the iteration continues indefinitely as adding additional flow introduce a negative term $\sum_{(+i) \in E} f_{+i}^k - \rho \sum_{(ij) \in E} w_{ij} f_{ij}^k = 1 - \rho \sum_{(ij) \in E} w_{ij} f_{ij}^k < 0$ which always makes the cost* (38a) *smaller.*



(a) Cross-trajectory co-observation plan

(b) Checkpoint graph and result flows

Fig. 7: Figure 7a showcase an unsecured 4-robot map exploration task. The cross-trajectory co-observation plan is shown as dotted arrows on top of the original plan in Figure 7a. The checkpoint graph and resulting flows are shown in Figure 7b.

F. Result and simulation

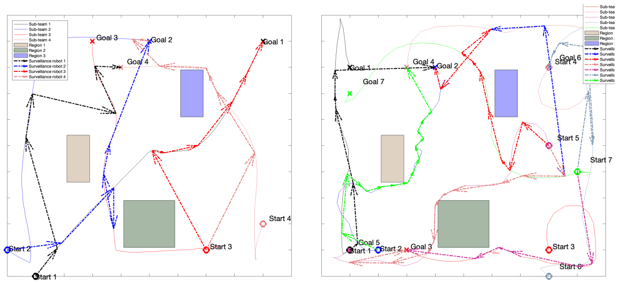
We first test the proposed method for the example application in Figures 4 and 4b, using the same setup in Section II-I. Using the parameters $w_c = 10$, $w_t = 1$ and $\rho = 0.01$, the result returns a total of $\mathcal{K} = 3$ surveillance robots with cross-trajectory plan shown in Figure 7. The flows derived from the solution of the optimization problem (38) are highlighted in the graph Figure 7b, where each horizontal line represents the original trajectory of a sub-team and the number on each vertex v_i represents the corresponding time t_i. The planning result in the workspace is shown in Figure 7a as dash-dotted arrows with the same color used for each flow in Figure 7b. Compared with the result in Figure 4b, it is easy to see that there is a significant improvement in map coverage in Figure 7a. At the same time, unsecured deviations to the forbidden regions of robots are secured through cross-trajectory observations.

The problem shows no solution for $\mathcal{K} \leq 2$. For cases $\mathcal{K} = 3$, problem return the optimal result as shown in Figure 8. If we further increase $\mathcal{K} > 3$, we do not get a better result; instead, the planner will return four flows with the rest $\mathcal{K} - 3$ flows empty.

We then test the proposed method for a 4-team and 7-team case with result shown in Fig. 8, where four trajectories are provided for a map exploration task with no security related constraints (co-observation schedule and reachability). We have a $10m \times 10m$ task space, three forbidden regions (rectangle regions in Figure 8a) and robots with a max velocity of $0.5m/dt$.

IV. Summary

This paper introduces security measures for protecting Multi-Robot Systems (MRS) from plan-deviation attacks. We establish a mutual observation schedule to ensure that any deviations into forbidden regions break the schedule, triggering detection.



(a) 4-agent co-observation plan (b) 7-agent co-observation plan

Fig. 8: Cross-trajectory co-observation result of a 4 sub-teams task (Figure 8a) and a 7 sub-teams task Figure 8b

Co-observation requirements and reachability analysis are incorporated into the trajectory planning phase, formulated as constraints for the optimal problem. In scenarios where a secured plan is infeasible or enhanced task performance is necessary, we propose using redundant robots for cross-trajectory co-observations, offering the same security guarantees against plan-deviation attacks. However, the time-sensitive and proximity-dependent nature of co-observation necessitates a focus on collision avoidance during planning. Future work will integrate collision avoidance and explore dynamic duty assignments within sub-teams, increasing the challenge for potential attackers.

APPENDIX

A. Transformation of reachability ellipsoid to canonical frame

The ellipse $\mathcal{E}$ expressed in $\mathcal{F}_\mathcal{E}$ is given by $\mathcal{E}^\mathcal{E} = \{q^\mathcal{E} \in \mathbb{R}^m : d(q_1^\mathcal{E}, q^\mathcal{E}) + d(q^\mathcal{E}, q_2^\mathcal{E}) < 2a\}$, with foci $q_1^\mathcal{E}, q_2^\mathcal{E}$ in $\mathcal{F}_\mathcal{E}$ defined as $q_1^\mathcal{E} = \begin{bmatrix} c & 0 & 0 \end{bmatrix}^\mathrm{T}, q_2^\mathcal{E} = \begin{bmatrix} -c & 0 & 0 \end{bmatrix}^\mathrm{T}$, and semi-axis distance $c = \frac{\|q_2 - q_1\|}{2}$.

Definition 7. *The* reachability ellipsoid $\mathcal{E}$ in the canonical frame *is defined as as the zero level set of the quadratic function*

$$E^\mathcal{E}(q^\mathcal{E}) = {q^\mathcal{E}}^\mathrm{T} Q q^\mathcal{E} - 1 \tag{39}$$

where

$$Q = \operatorname{diag}(a^{-2}, b^{-2}, b^{-2}), \tag{40}$$

and $b = \sqrt{a^2 - c^2}$. *The ellipse parameters* a, b *represent the lengths of the major axes.*

Lemma 1. *The original ellipse* $\mathcal{E}$ *in* $\mathcal{F}$ *can be expressed as the zero level set of the quadratic function*

$$E^\mathcal{F}(q) = (q - o_\mathcal{E})^\mathrm{T} H^\mathrm{T} Q H (q - o_\mathcal{E}) - 1 \tag{41}$$

Proof. The claim follows by substituting (10) into (39), and from the definition of R and o. □

B. Householder rotations

We define a differentiable transformation to a canonical ellipse that is used in the derivation of the reachability constraints in Sections II-C and II-H. This transformation includes a rotation derived from a modified version of Householder transformations [33]. We call our version of the operator a *Householder rotation*. In this section we derive Householder rotations and their differentials for the 3-D case; the 2-D case can be easily obtained by embedding it in the $z = 0$ plane. zyang $\nu_\mathcal{F}$ is defined in main content, i changed it to ν_1 and ν_2

Definition 8. *Let* ν_1 *and* ν_2 *be two unitary vectors* ($\|\nu_1\| = \|\nu_2\| = 1$). *Define the normalized vector* $u = \frac{\nu_1 + \nu_2}{\|\nu_1 + \nu_2\|}$, *the* Householder rotation $H(\nu_1, \nu_2)$ *is defined as*

$$H(\nu_1, \nu_2) = 2uu^\mathrm{T} - I. \tag{42}$$

Here H is a rotation mapping ν_1 to ν_2, as shown by the following.

Proposition 3. *The matrix* H *has the following properties:*

1) *It is a rotation, i.e.*
   a) $H^\mathrm{T} H = I$;
   b) $\det(H) = 1$.
2) $\nu_2 = H\nu_1$.

Proof. For subclaim *1)*a:

$$H^\mathrm{T} H = H^2 = 4uu^\mathrm{T} uu^\mathrm{T} - 4uu^\mathrm{T} + I^2 = I, \tag{43}$$

since $u^\mathrm{T} u = 1$.

For subclaim *1)*b, let $U = \begin{bmatrix} u & u_1^\perp & u_2^\perp \end{bmatrix}$, where $u_1^\perp, u_2^\perp$ are two orthonormal vectors such that $I = UU^\mathrm{T} = uu^\mathrm{T} + u_1^\perp (u_1^\perp)^\mathrm{T} + u_2^\perp (u_2^\perp)^\mathrm{T}$; then, substituting I in (42), we have that the eigenvalue decomposition of H is given by

$$H = U \operatorname{diag}(1, -1, -1) U^\mathrm{T}. \tag{44}$$

Since the determinant of a matrix is equal to the product of the eigenvalues, $\det(H) = 1$.

For subclaim *2)*, first note that $Hu = 2uu^\mathrm{T} u - u = u$. It follows that the sum of ν_1 and ν_2 is invariant under H:

$$H(\nu_1 + \nu_2) = Hu\|\nu_1 + \nu_2\| = u\|\nu_1 + \nu_2\| = \nu_1 + \nu_2, \tag{45}$$

and that their difference is flipped under H:

$$H(\nu_1 - \nu_2) = 2uu^\mathrm{T}(\nu_1 - \nu_2) - (\nu_1 - \nu_2)^2 = -(\nu_1 - \nu_2). \tag{46}$$

Combining (45) and (46) we obtain

$$H\nu_1 = \frac{1}{2}\big(H(\nu_1 + \nu_2) + H(\nu_1 - \nu_2)\big) = \nu_2 \tag{47}$$

□

We compute the differential of H implicitly using its definition (2). We use the notation $[v]_\times : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ to denote the matrix representation of the cross product with the vector v, i.e.,

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \tag{48}$$

such that $[v]_\times w = v \times w$ for any $w \in \mathbb{R}^3$.

Proposition 4. *Let* $\nu_1(t)$ *represent a parametric curve. Then*

$$\dot{H} = H[-2M\dot{\nu}_\mathcal{F}]_\times, \tag{49}$$

where the matrix $M \in \mathbb{R}^{3\times 3}$ is given by

$$M = [u]_\times \frac{\left(I - uu^\mathrm{T}\right)\left(I - \nu_1\nu_1^\mathrm{T}\right)}{\|u'\|\|\nu_1\|}. \tag{50}$$

Proof. From the definition of H in (42), we have

$$\dot{H} = 2(\dot{u}u^\mathrm{T} + u\dot{u}^\mathrm{T}) \tag{51}$$

Recall that $\dot{u} = \frac{1}{\|u'\|}(I - uu^\mathrm{T})\dot{u}'$ (see, for instance, [34]), which implies $(I - uu^\mathrm{T})\dot{u}' = \dot{u}'$. It follows that $\dot{u}$ flips sign under the action of H^T:

$$\begin{aligned} H^\mathrm{T}\dot{u} &= (2uu^\mathrm{T} - I)\frac{(I - uu^\mathrm{T})}{\|u'\|}\dot{u}' \\ &= \frac{1}{\|u'\|}(2uu^\mathrm{T} - I - 2uu^\mathrm{T}uu^\mathrm{T} + uu^\mathrm{T})\dot{u}' \\ &= -\frac{1}{\|u'\|}(I - uu^\mathrm{T})\dot{u}' = -\dot{u} \end{aligned} \tag{52}$$

Inserting $HH^\mathrm{T} = I$ in (51), we have

$$\begin{aligned} \dot{H} =& 2HH^\mathrm{T}(\dot{u}u^\mathrm{T} + u\dot{u}^\mathrm{T}) = 2H(-\dot{u}u^\mathrm{T} + u\dot{u}^\mathrm{T}) \\ =& -2H[[u]_\times \dot{u}]_\times \\ =& -2H\left[[u]_\times \frac{\left(I - uu^\mathrm{T}\right)\left(I - \nu_1\nu_1^\mathrm{T}\right)}{\|u'\|\,\|\nu_1\|}\dot{\nu}_\mathcal{F}\right]_\times \\ =& -2H[M\dot{\nu}_\mathcal{F}]_\times, \end{aligned} \tag{53}$$

which is equivalent to the claim. □

C. Proof of proposition 1

To make the notation more compact, we will use $\partial_q f$ instead of $\partial_{\left[\begin{smallmatrix} q_1 \\ q_2 \end{smallmatrix}\right]} f$ for the remainder of the proof. The differential of (18) can be represented as:

$$\begin{aligned} \dot{\pi}_{p\mathcal{E}} =& \dot{H}^{-1}SH(q_{avoid} - o) + H^{-1}\dot{S}H(q_{avoid} - o) \\ &+ H^{-1}S\dot{H}(q_{avoid} - o) + (H^{-1}SH - I)\dot{o} \end{aligned} \tag{54}$$

where

$$\begin{aligned} \dot{S} =& -S^2(Q\dot{s} + s\dot{Q}) \\ =& -S^2(Q\partial_q s\dot{q} - \partial_b Q\partial_q b\dot{q}) \end{aligned} \tag{55}$$

where

$$\partial_b Q = 2\frac{s}{b^3}\,\mathrm{diag}\{0, 1, 1\} \tag{56}$$

To compute the derivative $\partial_q \pi$, we need the expression of $\partial_q b$, $\partial_q o$ and $\partial_q s$; the first two can be easily derived using the equations above:

$$\partial_q b = \frac{1}{4b}\left[q_1 - q_2, q_2 - q_1\right]^\mathrm{T} \tag{57}$$

$$\partial_q o = \left[I/2, I/2\right]^\mathrm{T} \tag{58}$$

In order to get $\partial_q s$, we use the fact that $F\big(s(q)\big) = 0$ for all q; hence $F\big(\tilde{q}(t)\big) \equiv 0$, and $\partial_q F = 0$. We then have:

$$0 = \dot{F} = 2q^\mathrm{T}Q'\dot{q} + q^\mathrm{T}\partial_s Q' q\dot{s} + q^\mathrm{T}\partial_b Q' q\dot{b} \tag{59}$$

where

$$\partial_s Q' = -\,\mathrm{diag}\left(\frac{2a^2}{(s + a^2)^3}, \frac{2b^2}{(s + b^2)^3}, \frac{2b^2}{(s + b^2)^3}\right). \tag{60}$$

By moving term $\dot{s}$ to the left-hand side we can obtain:

$$\begin{aligned} \dot{s} &= (q^\mathrm{T}\partial_s Q' q)^{-1}(2q^\mathrm{T}Q'\dot{q} + q^\mathrm{T}\partial_b Q' q\dot{b}) \\ &= (q^\mathrm{T}\partial_s Q' q)^{-1}(-4q^\mathrm{T}Q'H[U\dot{q}]_\times(q_{avoid} - o) \\ &\quad - 2q^\mathrm{T}Q'H\dot{o} + q^\mathrm{T}\partial_b Q' q\dot{b}) \\ &= (q^\mathrm{T}\partial_s Q' q)^{-1}(-4q^\mathrm{T}Q'H[q_{avoid} - o]_\times U\dot{q} \\ &\quad - 2q^\mathrm{T}Q'H\dot{o} + q^\mathrm{T}\partial_b Q' q\dot{b}) \end{aligned} \tag{61}$$

The second term of equation (54) turns into:

$$\begin{aligned} &H^{-1}\dot{S}H(q_{avoid} - o) = -H^{-1}Q'q\dot{s} - sH^{-1}S^2\partial_b Q q\dot{b} \\ &= \big((q^\mathrm{T}\partial_s Q' q)^{-1}H^{-1}Q'qq^\mathrm{T}(4Q'H[q_{avoid} - o]_\times U \\ &+ 2Q'H\partial_q o - \partial_b Q' qq\partial_q b) - sH^{-1}S^2\partial_b Q q\partial_q b\big)\dot{q} \end{aligned} \tag{62}$$

Thus equation (54) could be written as:

$$\begin{aligned} \dot{\pi}_{p\mathcal{E}} =& \big(-2H[SH(q_{avoid} - o)]_\times U \\ &+ \big((q^\mathrm{T}\partial_s Q' q)^{-1}H^{-1}Q'qq^\mathrm{T}(4Q'H[q_{avoid} - o]_\times U \\ &+ 2Q'H\partial_q o - \partial_b Q' qq\partial_q b) - sH^{-1}S^2\partial_b Q q\partial_q b\big) \\ &- 2H^{-1}SH[q_{avoid} - o]_\times U \\ &+ (H^{-1}SH - I)\partial_q o\big)\dot{q}, \end{aligned} \tag{63}$$

from which the claim follows.

D. Proof of proposition 2

We first need to derive $\dot{d}_\mathcal{E}$ and $\dot{d}_{\mathcal{E}t}$

$$\dot{d}_\mathcal{E} = -n^\mathrm{T}\partial_q o\dot{q} \tag{64}$$

$$\begin{aligned} \dot{d}_{\mathcal{E}t} =& (\dot{n}_\mathcal{E}^\mathrm{T}Q^{-1}n_\mathcal{E} + n_\mathcal{E}^\mathrm{T}\dot{Q}^{-1}n_\mathcal{E} + n_\mathcal{E}^\mathrm{T}Q^{-1}\dot{n}_\mathcal{E})/\sqrt{n_\mathcal{E}^\mathrm{T}Q^{-1}n_\mathcal{E}} \\ =& (\sqrt{n_\mathcal{E}^\mathrm{T}Q^{-1}n_\mathcal{E}})^{-1}\left(-2n^\mathrm{T}H[Q^{-1}n_\mathcal{E}]_\times U\right. \\ &\left.+ n_\mathcal{E}^\mathrm{T}\partial_b Q^{-1}n_\mathcal{E}\partial_q b - 2n_\mathcal{E}Q^{-1}H[n]_\times U\right)\dot{q} \end{aligned} \tag{65}$$

Next, we need to derive $\dot{p}_{t1}$, $\dot{p}_{t2}$ and $\dot{p}_\mathcal{L}$. Since $p_\mathcal{L}$ could be written as

$$p_\mathcal{L} = \frac{d_\mathcal{E}Q^{-1}n_\mathcal{E}}{d_{\mathcal{E}t}^2}, \tag{66}$$

we have

$$\begin{aligned} \dot{p}_\mathcal{L} = &\Bigg((-\frac{d_{\mathcal{E}t}n^\mathrm{T}\partial_q o - 2d_\mathcal{E}\partial_q d_{\mathcal{E}t}}{d_{\mathcal{E}t}^3})Q^{-1}n_\mathcal{E} \\ &+ \frac{d_\mathcal{E}\partial_b Q^{-1}n_\mathcal{E}\partial_q b - 2d_\mathcal{E}Q^{-1}H[n]_\times U}{d_{\mathcal{E}t}^2}\Bigg)\dot{q} \end{aligned} \tag{67}$$

$$\begin{aligned} \dot{p}_1 = &\Bigg(-\frac{Q^{-1}n_\mathcal{E}\partial_q d_{\mathcal{E}t}}{d_{\mathcal{E}t}^2} \\ &+ \frac{\partial_b Q^{-1}n_\mathcal{E}\partial_q b - 2Q^{-1}H[n]_\times U}{d_{\mathcal{E}t}}\Bigg)\dot{q} \end{aligned} \tag{68}$$

subtracting $\dot{q}$ from (67) and (68), we can derive the result shown in (21)

References

[1] M. Brunner, H. Hofinger, C. Krauß, C. Roblee, P. Schoo, and S. Todt, "Infiltrating critical infrastructures with next-generation attacks," *Fraunhofer Institute for Secure Information Technology (SIT), Munich*, 2010.

[2] K. Wardega, R. Tron, and W. Li, "Resilience of multi-robot systems to physical masquerade attacks," in *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2019, pp. 120–125.

[3] K. Wardega, M. von Hippel, R. Tron, C. Nita-Rotaru, and W. Li, "Byzantine resilience at swarm scale: A decentralized blocklist protocol from inter-robot accusations," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 1430–1438.

[4] ——, "Hola robots: Mitigating plan-deviation attacks in multi-robot systems with co-observations and horizon-limiting announcements," *arXiv preprint arXiv:2301.10704*, 2023.

[5] Z. Yang and R. Tron, "Multi-agent trajectory optimization against plan-deviation attacks using co-observations and reachability constraints," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 241–247.

[6] ——, "Multi-agent path planning under observation schedule constraints," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6990–6997.

[7] J. Yu and S. M. LaValle, "Multi-agent path planning and network flow," in *Algorithmic Foundations of Robotics X: Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics*. Springer, 2013, pp. 157–173.

[8] X. Liu and P. Lu, "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 750–765, 2014.

[9] R. Van Parys and G. Pipeleers, "Online distributed motion planning for multi-vehicle systems," *2016 European Control Conference, ECC 2016*, pp. 1580–1585, 2016.

[10] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[11] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 477–483.

[12] J. Bento, N. Derbinsky, J. Alonso-Mora, and J. S. Yedidia, "A message-passing algorithm for multi-agent trajectory planning," in *Advances in neural information processing systems*, 2013, pp. 521–529.

[13] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on robotics and automation*, vol. 20, no. 3, pp. 443–455, 2004.

[14] J. Hu, P. Bhowmick, and A. Lanzon, "Distributed adaptive time-varying group formation tracking for multiagent systems with multiple leaders on directed graphs," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 140–150, 2019.

[15] R. Stern, N. Sturtevant, A. Felner, S. Koenig, H. Ma, T. Walker, J. Li, D. Atzmon, L. Cohen, T. Kumar *et al.*, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Proceedings of the International Symposium on Combinatorial Search*, vol. 10, 2019, pp. 151–158.

[16] J. Yu and S. M. LaValle, "Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1163–1177, 2016.

[17] H. Guéguen, M.-A. Lefebvre, J. Zaytoon, and O. Nasri, "Safety verification and reachability analysis for hybrid systems," *Annual Reviews in Control*, vol. 33, no. 1, pp. 25–36, 2009.

[18] D. Ding, Q.-L. Han, X. Ge, and J. Wang, "Secure state estimation and control of cyber-physical systems: A survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 176–190, 2020.

[19] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis," in *International workshop on hybrid systems: Computation and control*. Springer, 2000, pp. 202–214.

[20] N. M. B. Lakhal, L. Adouane, O. Nasri, and J. B. H. Slama, "Interval-based solutions for reliable and safe navigation of intelligent autonomous vehicles," in *2019 12th International Workshop on Robot Motion and Control (RoMoCo)*. IEEE, 2019, pp. 124–130.

[21] M. Maiga, N. Ramdani, L. Travé-Massuyès, and C. Combastel, "A comprehensive method for reachability analysis of uncertain nonlinear hybrid systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2341–2356, 2015.

[22] C. Kwon and I. Hwang, "Reachability analysis for safety assurance of cyber-physical systems against cyber attacks," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2272–2279, 2017.

[23] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2997–3004.

[24] R. Deits and R. Tedrake, "Computing large convex regions of obstacle-free space through semidefinite programming," in *Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2015, pp. 109–124.

[25] A. Ray, A. Pierson, and D. Rus, "Free-space ellipsoid graphs for multi-agent target monitoring," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6860–6866.

[26] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017.

[27] D. Fan, Q. Liu, C. Zhao, K. Guo, Z. Yang, X. Yu, and L. Guo, "Flying in narrow spaces: Prioritizing safety with disturbance-aware control," *IEEE Robotics and Automation Letters*, 2024.

[28] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[30] D. Eberly, "Distance from a point to an ellipse, an ellipsoid, or a hyperellipsoid," 2013. [Online]. Available: https://www.geometrictools.com/

[31] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, no. 2, pp. 267–274, 2010.

[32] S. Ntafos and S. Hakimi, "On path cover problems in digraphs and applications to program testing," *IEEE Transactions on Software Engineering*, vol. SE-5, no. 5, pp. 520–529, 1979.

[33] A. S. Householder, "Unitary triangularization of a nonsymmetric matrix," *Journal of the ACM (JACM)*, vol. 5, no. 4, pp. 339–342, 1958.

[34] R. Tron and K. Daniilidis, "Technical report on optimization-based bearing-only visual homing with applications to a 2-d unicycle model," 2014.