

Multi-Agent Trajectory Optimization Against Plan-Deviation Attacks using Co-Observations and Reachability Constraints

Ziqi Yang and Roberto Tron

Abstract—In this paper, we focus on using path planning and inter-agent measurements to improve the security of multi-robot systems against possible takeovers from cyber-attackers. We build upon recent trajectory optimization approaches where *introspective* measurement capabilities of the robots are used in an *co-observation schedule* to detect deviations from the preordained routes. This paper proposes additional constraints that can be incorporated in the previous trajectory optimization algorithm based on Alternating Direction Method of Multipliers (ADMM). The new constraints provide guarantees that a compromised robot cannot reach a designed safety zone between observations despite adversarial movement by the attacker. We provide a simulation showcasing the new components of the formulation in a multi-agent map exploration task with several safety zones.

Index Terms—Path planning, Trajectory optimization, Observation planning, ADMM, Cyber-Physical Security; Attack Detection, Reachability region.

I. INTRODUCTION

Multi-robots systems have been used in commercial applications to take advantage of their capability of performing a large variety of settings, such as warehouse goods transportation, unknown fields exploration, surveillance, etc. Companies that have been employing this technology for years include Amazon Robotics and Fetch Robotics. In the academic research setting, applications include aerial swarms for precision agriculture [1] and forest fire monitoring [2]. Current multi-robot systems are typically equipped with planning algorithms that are geared toward ensuring efficient navigation, collision avoidance, and safety for robot-human interactions. However, there exists an increased risk of hacking for these systems that rely on network communications [3]. As a result, it is important to introduce defense mechanisms that can enhance the cyber-security of multi-robot systems.

There exist some rich literature on multi-agent pathfinding (MAPF) problems [4], but only a few have taken safety requirements into consideration. [5] focus on physical layer resilient control, models the game for CPS security, and provides resilience through a game-theoretic approach. [6], [7] made use of the physics of wireless signals, information from sensing infrastructure and proximity graph of vehicles, leverage the physics of environment to defend attacks. Our paper took a similar approach of using physical layers to detect and mitigate attacks.

In this paper, we focus on the defense of *plan-deviation attacks*, previously introduced in [8], where compromised

robots intend to perform a deviation from the MAPF plan and enter forbidden regions. Previous work [8], [9] posit that the overall security of the system against this kind of attack can be enhanced by combining the detection capabilities of the agents (via their onboard sensors) with path planning. This idea uses an *introspection-based* security layer that detects malicious actions of compromised robots whenever a sufficiently significant deviation from the pre-planned routes happens. The robots observe each other according to an *co-observation schedule* along specific routes; the inter-agent observations are frequent enough such that any unexpected absence of an agent at the pre-planned observation time and location will trigger an alarm. On the one hand, the joint path and co-observation schedule planning problem has been tackled using Satisfiability-Modulo-Theory (SMT) [8], which leads to a computationally complete algorithm (i.e., it always finds a solution if one exists), but is limited to discrete abstraction (e.g., a four-connect grid map) and does not scale well in the number of agents or the size of the environment. On the other hand, the approach of [9] uses a trajectory optimization algorithm based on ADMM which can handle continuous spaces and can deal with a large variety type of path planning constraints and complex cost functions (e.g., field estimation uncertainty in map exploring tasks). The latter approach provides local optimality guarantees with respect to the planning cost but does not provide the same strong guarantees on the detection of agents that do not follow the planned routes.

A possible failure case is due to the case where the attacker changes a robot's path between two observations to reach a forbidden region while still satisfying all the observation times and locations from the co-observation schedule. More broadly, the problem of solving a path optimization problem with constraints based on the sets of locations that the agents could *potentially* reach, which we call *reachability regions*, has not received attention in the literature.

a) *Paper contributions*: In this paper, we formulate a way to enforce an empty intersection between forbidden region and reachability regions, such that if an attacker takes control of the robots, they cannot perform an undetected attack by entering forbidden regions and meeting co-observation schedules at the same time. We propose a mathematical formulation of reachability region that can be solved in trajectory optimization problems together with other spatio-temporal constraints. The constraints are formulated using an ellipsoidal bound of the reachability region. The idea is inspired by the *heuristic sampling domain* introduced by [10] in the context of the RRT* path planning algorithm; in that

case, an ellipsoidal bound is used to limit the search space with the initial and goal states fixed. In our case, we use a similar bound to optimize the location of the two states to exclude the forbidden region from the ellipsoidal region. As a secondary contribution, we introduce the concept of *Householder rotations* for defining differentiable rigid changes of coordinates.

b) Paper outline: We first review relevant prior work with particular attention to the ADMM formulation from [9] (Section II-B), which we use as our optimization framework. We then introduce *Householder rotations* (Section II-C), which are a modification of Householder transformations that we use for defining differentiable rigid changes of coordinates. Next, we define *reachability ellipsoids* and their transformation to a canonical frame (Section III); these sets are used to formulate four types of constraints, which consider reachability of a point, a plane, a segment, and a convex polygon (Section IV). Finally, we include a simulation that provides a demonstration of how our novel constraints can be incorporated into a path optimization application. Our algorithm is validated on a benchmark field exploration task where three robots are asked to minimize the overall sensing uncertainty on a vector field (estimated using Kalman filtering), using the co-observation scheduled generated by [8] as initialization.

II. PRELIMINARIES

In this section, we review various mathematical concepts that will provide the foundations and context for our novel constraints.

A. Differentials

We define the differential of a map $f(x) : \mathbb{R}^m \rightarrow \mathbb{R}^n$ at a point x_0 as the unique matrix $\partial_x f \in \mathbb{R}^{n \times m}$ such that

$$\left. \frac{d}{dt} f(x(t)) \right|_{t=0} = \partial_x f(x(0)) \dot{x}(0) \quad (1)$$

where $t \mapsto x(t) \in \mathbb{R}^n$ is a smooth parametric curve such that $x(0) = x$ with any arbitrary tangent $\dot{x}(0)$. For later part of this paper, we will use $\dot{f}$ for $\frac{d}{dt} f$ and $\partial_x f$ for $\frac{\partial f}{\partial x}$. The differentials $\partial_x f$ is derived through (1) having $\dot{f}$ divided by $\dot{x}$.

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 $f : \mathbb{R}^R \rightarrow \mathbb{R}^{m \times n}$. Note that in this case (1) is still formally correct, although semantically different.

B. Alternating Directions Method of Multipliers (ADMM)

In this section, we briefly review the path planning algorithm based on ADMM constraints discussed in [9]. This algorithm can deal with a variety of types of non-convex and non-smooth optimization constraints that are commonly encountered in trajectory optimization tasks once they are formulated in a certain format. The goal of the section is to lay the groundwork for proposing our novel constraint in the following sections.

Let $x_{ij} \in \mathbb{R}^m$ denote the position of agent $i \in \{1, \dots, n\}$ at the discrete-time index $j \in [0, \dots, T]$, with m being the

dimension of the workspace, n the number of agents, and T the planning time horizon. Let $\mathbf{x} = \text{stack}(\{x_{ij}\})$ denote the aggregated vector of all the agents' trajectories over the entire time interval. The goal of the algorithm is to optimize an objective function $\Phi(\mathbf{x})$ on a set Ω defined by nonlinear constraints. To solve the problem using ADMM, we introduce a new set of variables $\mathbf{z} = D(\mathbf{x})$ to translate the constraints Ω on x to (typically lower-dimensional) sets of constraints on $\mathbf{z}$, such that the projection operator on ζ can be easily implemented (e.g., a constrained distance between two agents is translated into their difference constrained to an origin-centered sphere). The new set of constraints ζ on $\mathbf{z}$ is incorporated using an indicator function Θ_ζ . The main problem is written as:

$$\begin{aligned} \min \quad & \Phi(\mathbf{x}) + \Theta_\zeta(\mathbf{z}) \\ \text{s.t.} \quad & D(\mathbf{x}) - \mathbf{z} = 0, \end{aligned} \quad (2)$$

where $D(\mathbf{x}) = [D_1(\mathbf{x})^T, \dots, D_l(\mathbf{x})^T]^T$ is a vertical concatenation of different functions for different constraints.

And having the update steps shown as:

$$\mathbf{x}^{k+1} := \underset{\mathbf{x}}{\operatorname{argmin}} (\Phi(\mathbf{x}^k) + \frac{\rho}{2} \|D(\mathbf{x}) - \mathbf{z}^k + \mathbf{u}^k\|_2^2) \quad (3a)$$

$$\mathbf{z}^{k+1} := \Pi_\zeta(D(\mathbf{x}^{k+1}) + \mathbf{u}^k) \quad (3b)$$

$$\mathbf{u}^{k+1} := \mathbf{u}^k + D(\mathbf{x}^{k+1}) - \mathbf{z}^{k+1}, \quad (3c)$$

Previous work in [9] considered the following types of constraints:

- (C1) *Velocity constraints:* The movement of each agent is constrained by a maximum distance in any direction over a single discrete time step. This constraint could be adapted also to enforce more refined dynamical models.
- (C2) *Convex obstacles:* We use convex polygons to model regions defining the boundary of the workspace, as well as solid obstacles that cannot be entered by agents (note that the latter represent a non-convex constraint set from the point of view of the optimization problem). Obstacles with non-convex shapes can be modeled using unions of the convex sets.
- (C3) *Waypoints with flexible deadlines:* We give locations that the agents need to visit at any point in a given time window (which can potentially contain a single time). These constraints can also be interpreted as static, trusted cameras that can complement the introspection constraints.
- (C4) *Introspection constraints:* In a given time window, agents are asked to remain at a certain distance from each other to detect each other's presence.

Constraint (C4) is the main mechanism that was provided in [9] for forcing agents to stay away from the forbidden region. However, this mechanism alone cannot provide strong guarantees in this regard. This motivates the introduction of our novel *ellipsoidal constraints* in Sections IV-A, IV-B. This novel constraint is defined with the assumption that the velocity constraints still hold under attack.

C. Householder rotations

To formulate the operator Π_ζ for our novel constraint, we will define a differentiable transformation of the constraint

set to a canonical form. This transformation will include a rotation that we derive from a modified version of Householder transformations [11]. With respect to the standard definition, our modification ensures that the final operator is a proper rotation (i.e., not a reflection). 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.

Definition 1: Let $\nu_{\mathcal{F}}$ and $\nu_{\mathcal{E}}$ be two unitary vectors ($\|\nu_{\mathcal{F}}\| = \|\nu_{\mathcal{E}}\| = 1$). Define the normalized vector u as

$$u' = \nu_{\mathcal{F}} + \nu_{\mathcal{E}}, \quad u = \frac{u'}{\|u'\|}. \quad (4)$$

The *Householder rotation* $H(\nu_{\mathcal{F}}, \nu_{\mathcal{E}})$ is defined as

$$H(\nu_{\mathcal{F}}, \nu_{\mathcal{E}}) = 2uu^T - I. \quad (5)$$

The main property of interest for our application is the fact that H is a rotation mapping $\nu_{\mathcal{F}}$ to $\nu_{\mathcal{E}}$, as shown by the following.

Proposition 1: The matrix H has the following properties:

- 1) It is a rotation, i.e.
 - a) $H^T H = I$;
 - b) $\det(H) = 1$.
- 2) $\nu_{\mathcal{E}} = H\nu_{\mathcal{F}}$.

Proof: For subclaim 1)a:

$$H^T H = H^2 = 4uu^T uu^T - 4uu^T + I^2 = I, \quad (6)$$

since $u^T u = 1$.

For subclaim 1)b, let $U = [u \ u_1^\perp \ u_2^\perp]$, where $u_1^\perp, u_2^\perp$ are two orthonormal vectors such that $I = UU^T = uu^T + u_1^\perp (u_1^\perp)^T + u_2^\perp (u_2^\perp)^T$; then, substituting I in (5), we have that the eigenvalue decomposition of H is given by

$$H = U \text{diag}(1, -1, -1) U^T. \quad (7)$$

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

For subclaim 2), first note that $Hu = 2uu^T u - u = u$. It follows that the sum of $\nu_{\mathcal{F}}$ and $\nu_{\mathcal{E}}$ is invariant under H :

$$H(\nu_{\mathcal{F}} + \nu_{\mathcal{E}}) = Hu\|\nu_{\mathcal{F}} + \nu_{\mathcal{E}}\| = u\|\nu_{\mathcal{F}} + \nu_{\mathcal{E}}\| = \nu_{\mathcal{F}} + \nu_{\mathcal{E}}, \quad (8)$$

and that their difference is flipped under H :

$$H(\nu_{\mathcal{F}} - \nu_{\mathcal{E}}) = 2uu^T(\nu_{\mathcal{F}} - \nu_{\mathcal{E}}) - (\nu_{\mathcal{F}} - \nu_{\mathcal{E}})^2 = -(\nu_{\mathcal{F}} - \nu_{\mathcal{E}}). \quad (9)$$

Combining (8) and (9) we obtain

$$H\nu_{\mathcal{F}} = \frac{1}{2}(H(\nu_{\mathcal{F}} + \nu_{\mathcal{E}}) + H(\nu_{\mathcal{F}} - \nu_{\mathcal{E}})) = \nu_{\mathcal{E}} \quad (10)$$

We compute the differential of H implicitly using the relation (1). We will use the notation $[v]_{\times} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ to denote the matrix representation of the cross product with the vector v , i.e.,

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \mapsto \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \quad (11)$$

such that $[v]_{\times} w = v \times w$ for any $w \in \mathbb{R}^3$. One can verify by direct computation the following property:

$$wv^T - vw^T = [[v]_{\times} w]_{\times}. \quad (12)$$

Proposition 2: Let $\nu_{\mathcal{F}}(t)$ represent a parametric curve. Then we have

$$\dot{H} = H[-2M\dot{\nu}_{\mathcal{F}}]_{\times} \quad (13)$$

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

$$M = [u]_{\times} \frac{(I - uu^T)(I - \nu_{\mathcal{F}}\nu_{\mathcal{F}}^T)}{\|u'\|\|\nu_{\mathcal{F}}\|}. \quad (14)$$

Proof: From the definition of $\dot{H}$ in (5), we have

$$\dot{H} = 2(\dot{u}u^T + u\dot{u}^T) \quad (15)$$

Recall that $\dot{u} = \frac{1}{\|u'\|}(I - uu^T)\dot{u}'$ (see, for instance, [12]), which implies $(I - uu^T)\dot{u}' = \dot{u}'$. It follows that $\dot{u}$ flips sign under the action of H^T :

$$\begin{aligned} H^T \dot{u} &= (2uu^T - I) \frac{(I - uu^T)}{\|u'\|} \dot{u}' \\ &= \frac{1}{\|u'\|} (2uu^T - I - 2uu^T uu^T + uu^T) \dot{u}' \\ &= -\frac{1}{\|u'\|} (I - uu^T) \dot{u}' = -\dot{u} \end{aligned} \quad (16)$$

Inserting $HH^T = I$ in (15), and using 12, we finally have

$$\begin{aligned} \dot{H} &= 2HH^T(\dot{u}u^T + u\dot{u}^T) = 2H(-\dot{u}u^T + u\dot{u}^T) \\ &= -2H[[u]_{\times} \dot{u}]_{\times} \\ &= -2H \left[[u]_{\times} \frac{(I - uu^T)(I - \nu_{\mathcal{F}}\nu_{\mathcal{F}}^T)}{\|u'\|\|\nu_{\mathcal{F}}\|} \dot{\nu}_{\mathcal{F}} \right]_{\times} \\ &= -2H[M\dot{\nu}_{\mathcal{F}}]_{\times}, \end{aligned} \quad (17)$$

which is equivalent to the claim. $\blacksquare$

III. ELLIPSOIDAL REACHABILITY CONSTRAINTS

In this section, we first provide the definition of an *ellipsoidal reachability region*. We then provide a differentiable map to transform such region in a canonical axis-aligned form where the operator Π_{ζ} and its differential can be obtained; these operators are then extended to the general case via the aforementioned transform. The overall goal is to define the functions $D(q)$, its differential, and the operator Π_{ζ} for *ellipsoidal reachability regions* with respect to single points in the forbidden regions that can then be used in the ADMM formulation reviewed in section II-B.

A. Definition of reachability region

The reachability region is defined as the set of locations $x(t)$ that a robot can reach between two given fixed positions:

Definition 2: The *reachability region* for two waypoints $x(t_1) = x_1, x(t_2) = x_2$ is defined as the sets of points x' in the workspace such that there exist a trajectory $x(t)$ where $x(t') = x', t_1 \leq t' \leq t_2$ and $x(t)$ satisfies the velocity constraint $d(x(t), x(t+1)) \leq v_{max}$.

This region can be analytically bound via an ellipsoid:

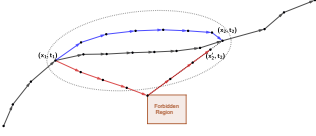


Fig. 1: The ellipse is a showcase of the reachability region. The black line is the trajectory of an agent, x_1 and x_2 are two locations this agent is expected at given time t_1 and t_2 , blue and red line are possible trajectories if the agent is compromised in after reaching x_1 . Agent that goes outside the reachability region like the red line could not be able to get back to x_2 at t_2 .

Definition 3: The *reachability ellipsoid* is the region $\mathcal{E}(x_1, x_2) = \{\tilde{x} \in \mathbb{R}^n : d(x_1, \tilde{x}) + d(\tilde{x}, x_2) < 2a\}$, where $a = \frac{v_{max}}{2}(t_2 - t_1)$.

The region $\mathcal{E}(x_1, x_2)$ is an ellipsoid with foci at x_1, x_2 , center $o_{\mathcal{E}} = \frac{1}{2}(x_1 + x_2)$, and the major radius equal to a . Let $c_{\mathcal{E}} = \frac{1}{2}\|x_1 - x_2\| = \|o_{\mathcal{E}} - x_1\|$ be the distance from the center to a foci.

The reachability ellipsoid is an over-approximation of the exact reachability region; the difference between the two is due to the discretization of the trajectory, and the fact that $\mathcal{E}$ does not consider the presence of obstacles.

B. Transformation to canonical coordinates

To simplify the problem, a canonical rigid body transformation is used to transform the ellipse $\mathcal{E}$ from a global frame $\mathcal{F}$ to a canonical frame $\mathcal{F}_{\mathcal{E}}$. The latter is defined such that the center of the ellipsoid is located at the origin and the foci are aligned with the first axis of $\mathcal{F}_{\mathcal{E}}$. Since the transformation depends on the two foci, i.e., two waypoints of an agent, the challenge here is to construct the transformation in a differentiable way.

For the convenience of derivation, we define the coordinate transformation from $\mathcal{F}$ to $\mathcal{F}_{\mathcal{E}}$ using a rotation $R_{\mathcal{E}}^{\mathcal{F}}$ and a translation $o_{\mathcal{E}}^{\mathcal{F}}$, which, to simplify the notation, from now on we simply refer to as R and o , respectively. The transformation of a point from the frame $\mathcal{F}_{\mathcal{E}}$ to the frame $\mathcal{F}$ and its inverse are given by the formulation:

$$x^{\mathcal{F}} = Rx^{\mathcal{E}} + o, \quad x^{\mathcal{E}} = R^T(x^{\mathcal{F}} - o). \quad (18)$$

The reverse transformation is $x^{\mathcal{E}} = R^T(x^{\mathcal{F}} - o)$.

We define $\nu_{\mathcal{F}}$ and $\nu_{\mathcal{E}}$ to represent the x -axis unitary vector of $\mathcal{F}_{\mathcal{E}}$ in the frames $\mathcal{F}$ and $\mathcal{F}_{\mathcal{E}}$, respectively. formally:

$$\nu'_{\mathcal{F}} = x_2 - x_1, \quad \nu_{\mathcal{F}} = \frac{\nu'_{\mathcal{F}}}{\|\nu'_{\mathcal{F}}\|}, \quad \nu_{\mathcal{E}} = [1, 0, 0]^T; \quad (19)$$

see Fig.1 for an illustration. Note that $\nu_{\mathcal{E}}$ is constant and, for the sake of clarity, we have suppressed the dependency of $\nu_{\mathcal{F}}$ on x_1, x_2 . We then define the rotation R using a Householder rotation, while from (18) we see that o represents the center

of $\mathcal{E}(x_1, x_2)$ expressed in $\mathcal{F}$, i.e.:

$$R = H(\nu_{\mathcal{F}}(x_1, x_2), \nu_{\mathcal{E}}(x_1, x_2)), \quad o = \frac{1}{2}(x_1 + x_2). \quad (20)$$

To simplify the notation, in the following we will consider H to be a function of x_1, x_2 directly, i.e. $H(x_1, x_2)$.

IV. REACHABILITY CONSTRAINTS VIA ELLIPSOIDS

In this section, we define constraints that consider the ellipsoid regions from the section above against different types of forbidden regions: a point, a plane, a segment, and a convex polygon. For each one, our goal is to define the function $D(x)$, the set ζ and the projection Π_{ζ} that can be incorporated in the ADMM optimization (2); we also include derivations for the differential $\partial_x D(x)$, which can be used to significantly speed up the optimization.

A. Point-ellipsoid constraint

Consider a forbidden region in the shape of a single point q_{avoid} . The goal is to design the trajectory $x(t)$ such that $q_{avoid} \notin \mathcal{E}(x_1, x_2)$. We first define a projection function $\pi_p(q_{avoid}; x_1, x_2, a) = q_p$, which returns a projected point q_p of q_{avoid} outside the ellipse, i.e., as the solution to

$$\begin{aligned} \min \quad & \|q_{avoid} - q_p\|^2 \\ \text{s.t.} \quad & q_p \in \mathcal{E}^c. \end{aligned} \quad (21)$$

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

Then, the constraint $q_{avoid} \notin \mathcal{E}(x(t_1), x(t_2))$ can be written as:

$$D(x) = \pi_p(q_{avoid}; x(t_1), x(t_2), r) - q_{avoid} = 0 \quad (22)$$

For cases where $q_{avoid} \notin \mathcal{E}(x(t_2), x(t_1), r)$, $\pi_p(q_{avoid}) = q_{avoid}$. And for cases where $q_{avoid} \in \mathcal{E}(x_1, x_2, r)$, $D(x)$ needs to be projected to the boundary of the ellipse, which is discussed below.

1) *Projection to the standard ellipse:* The ellipse $\mathcal{E}$ expressed in $\mathcal{F}_{\mathcal{E}}$ is given by $\mathcal{E}^{\mathcal{E}} = \{x^{\mathcal{E}} \in \mathbb{R}^m : d(x_1^{\mathcal{E}}, x^{\mathcal{E}}) + d(x^{\mathcal{E}}, x_2^{\mathcal{E}}) < 2a\}$, where the coordinates of the two foci $x_1^{\mathcal{E}}, x_2^{\mathcal{E}}$ in $\mathcal{F}_{\mathcal{E}}$ are

$$x_1^{\mathcal{E}} = [c \ 0 \ 0]^T, \quad x_2^{\mathcal{E}} = [-c \ 0 \ 0]^T, \quad (23)$$

with $c = \frac{\|x_2 - x_1\|}{2}$.

The ellipsoid $\mathcal{E}$ in the canonical frame can be described as the zero level set of the quadratic function

$$E^{\mathcal{E}}(x^{\mathcal{E}}) = x^{\mathcal{E}T} Q x^{\mathcal{E}} - 1 \quad (24)$$

where

$$Q = \text{diag}(a^{-2}, b^{-2}, b^{-2}), \quad (25)$$

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

The point to project, q_{avoid} , can be likewise expressed in $\mathcal{F}_{\mathcal{E}}$ as $q^{\mathcal{E}}_{avoid} = H(q^{\mathcal{F}}_{avoid} - o)$.

We now turn our attention to the problem of projecting $q^{\mathcal{E}}_{avoid}$ on the zero level set of $E^{\mathcal{E}}$ (i.e., the reachability ellipsoid in the canonical frame). The derivations below are loosely inspired by [13].

Let $q_p^\mathcal{E}$ be the point on the surface of the ellipsoid, i.e., $E^\mathcal{E}(q_p^\mathcal{E}) = 0$, corresponding to the projection of the point $q_{avoid}^\mathcal{E}$. Using Lagrange multipliers applied to the constrained optimization problem (21) (after transforming it in the canonical frame), one can show that the vector from a point to its projection, $q_{avoid}^\mathcal{E} - q_p^\mathcal{E}$, must be collinear with the gradient of $\mathcal{E}^\mathcal{E}$, i.e.

$$q_p^\mathcal{E} - q_{avoid}^\mathcal{E} = s \partial_q E^\mathcal{E}(q_p^\mathcal{E})^\top = s Q q_p^\mathcal{E} \quad (26)$$

for some scale $s \in \mathbb{R}$; thus $q_p^\mathcal{E}$ can be written as:

$$q_p^\mathcal{E} = (I + sQ)^{-1} q_{avoid}^\mathcal{E} = S q_{avoid}^\mathcal{E} \quad (27)$$

where $S = (I + sQ)^{-1}$. Using the fact that since $q_p^\mathcal{E}$ is a point on the ellipse, s can be solved as the root of the equation obtained by substituting (27) in $E^\mathcal{E}(x^\mathcal{E})$:

$$0 = F(s) = q_p^{\mathcal{E}\top} Q q_p^\mathcal{E} - 1 = q_{avoid}^{\mathcal{E}\top} Q'(s) q_{avoid}^\mathcal{E} - 1, \quad (28)$$

where

$$Q'(s) = S^\top Q S = \text{diag} \left(\frac{a^2}{(s+a^2)^2}, \frac{b^2}{(s+b^2)^2}, \frac{b^2}{(s+b^2)^2} \right) \quad (29)$$

Detailed methods for computing s can be found in [13].

Then the point-to-ellipse projection function can be represented as:

$$\begin{aligned} \pi_{p\mathcal{E}}(x) &= R^{-1}(x(t_1), x(t_2)) q_p^\mathcal{E} + o \\ &= R^{-1}(x(t_1), x(t_2)) S q_{avoid}^\mathcal{E} + o \\ &= R^{-1} S R (q_{avoid} - o) + o \end{aligned} \quad (30)$$

In our derivations, we consider only the 3-D case ($m = 3$); for the 2-D case, let $P = [I \ 0] \in \mathbb{R}^{2 \times 3}$: then $\pi_{p\mathcal{E}}^{2D} = P \pi_{p\mathcal{E}}^{3D}(P^\top q_{avoid}; P^\top x_1, P^\top x_2, a)$.

2) *ADMM constraints*: The corresponding constraint is written as

$$D_p(x) = \begin{cases} \pi_{p\mathcal{E}}(x) - q_{avoid} & q_{avoid} \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \quad (31)$$

and feasible set and projection function as:

$$\zeta = \{x \in \mathbb{R}^{nm} : \|D_p(x)\| = 0\}, \quad \Pi(D_p(x)) = 0 \quad (32)$$

3) Differential of the constraint:

Proposition 3: The differential of the projection operator $\pi_{p\mathcal{E}}(x_{avoid}; x_1, x_2, a)$ with respect to the foci x_1, x_2 is given by the following (where we use q as a shorthand notation for $q_{avoid}^\mathcal{E}$)

$$\begin{aligned} \partial_{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} \pi_{p\mathcal{E}} &= -2H[SH(q-o)]_\times U \\ &+ ((q^\top \partial_s Q' q)^{-1} H^{-1} Q' q q^\top (4Q' H[q-o]_\times U \\ &+ 2Q' H \partial_x o - \partial_b Q' q q \partial_x b) - sH^{-1} S^2 \partial_b Q q \partial_x b) \\ &- 2H^{-1} SH[q-o]_\times U + (H^{-1} SH - I) \partial_x o \end{aligned} \quad (33)$$

Proof: To make the notation more compact, we will use $\partial_x f$ instead of $\partial_{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} f$ for the remainder of the proof. The differential of (30) can be represented as:

$$\begin{aligned} \dot{\pi}_{p\mathcal{E}} &= \dot{H}^{-1} SH(q_{avoid} - o) + H^{-1} \dot{S} H(q_{avoid} - o) \\ &+ H^{-1} S \dot{H}(q_{avoid} - o) + (H^{-1} SH - I) \dot{o} \end{aligned} \quad (34)$$

where

$$\begin{aligned} \dot{S} &= -S^2(Q\dot{s} + s\dot{Q}) \\ &= -S^2(Q\partial_x s \dot{x} - \partial_b Q \partial_x b \dot{x}) \end{aligned} \quad (35)$$

where

$$\partial_b Q = 2 \frac{s}{b^3} \text{diag}\{0, 1, 1\} \quad (36)$$

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

$$\partial_x b = \frac{1}{4b} [x_1 - x_2, x_2 - x_1]^\top \quad (37)$$

$$\partial_x o = [I/2, I/2]^\top \quad (38)$$

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

$$0 = \dot{F} = 2q^\top Q' \dot{q} + q^\top \partial_s Q' q \dot{s} + q^\top \partial_b Q' q \dot{b} \quad (39)$$

where

$$\partial_s Q' = -\text{diag} \left(\frac{2a^2}{(s+a^2)^3}, \frac{2b^2}{(s+b^2)^3}, \frac{2b^2}{(s+b^2)^3} \right). \quad (40)$$

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

$$\begin{aligned} \dot{s} &= (q^\top \partial_s Q' q)^{-1} (2q^\top Q' \dot{q} + q^\top \partial_b Q' q \dot{b}) \\ &= (q^\top \partial_s Q' q)^{-1} (-4q^\top Q' H[U\dot{x}]_\times (q_{avoid} - o) \\ &\quad - 2q^\top Q' H \dot{o} + q^\top \partial_b Q' q \dot{b}) \\ &= (q^\top \partial_s Q' q)^{-1} (-4q^\top Q' H[q_{avoid} - o]_\times U \dot{x} \\ &\quad - 2q^\top Q' H \dot{o} + q^\top \partial_b Q' q \dot{b}) \end{aligned} \quad (41)$$

The second term of equation (34) turns into:

$$\begin{aligned} H^{-1} \dot{S} H(q_{avoid} - o) &= -H^{-1} Q' q \dot{s} - sH^{-1} S^2 \partial_b Q q \dot{b} \\ &= ((q^\top \partial_s Q' q)^{-1} H^{-1} Q' q q^\top (4Q' H[q_{avoid} - o]_\times U \\ &\quad + 2Q' H \partial_x o - \partial_b Q' q q \partial_x b) - sH^{-1} S^2 \partial_b Q q \partial_x b) \dot{x} \end{aligned} \quad (42)$$

Thus equation (34) could be written as:

$$\begin{aligned} \dot{\pi}_{p\mathcal{E}} &= (-2H[SH(q_{avoid} - o)]_\times U \\ &+ ((q^\top \partial_s Q' q)^{-1} H^{-1} Q' q q^\top (4Q' H[q_{avoid} - o]_\times U \\ &+ 2Q' H \partial_x o - \partial_b Q' q q \partial_x b) - sH^{-1} S^2 \partial_b Q q \partial_x b) \\ &- 2H^{-1} SH[q_{avoid} - o]_\times U \\ &+ (H^{-1} SH - I) \partial_x o) \dot{x}, \end{aligned} \quad (43)$$

from which the claim follows. $\blacksquare$

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

B. Plane-ellipsoid constraint

Consider a forbidden region in the shape of a hyperplane $\mathcal{L}(\tilde{x}) = \{\tilde{x} \in \mathbb{R}^n : \mathbf{n}^\top \tilde{x} = d\}$. The reachability constraint not can then be defined as $\mathcal{L} \cap \mathcal{E}(x_1, x_2, a) = \emptyset$. Using the transformation introduced in (III-B), all ellipse can be transferred into a standard one with the new hyperplane in the form of $\mathcal{L}^\mathcal{E}(\tilde{x}) = \{\tilde{x} \in \mathbb{R}^m : \mathbf{n}_\mathcal{E}^\top \tilde{x} = d_\mathcal{E}\}$, where,

$$\mathbf{n}_\mathcal{E} = H(x_1, x_2) \mathbf{n} \quad (44)$$

$$d_\mathcal{E} = -\mathbf{n}^\top o + d \quad (45)$$

1) *Projection via the tangent interpolation point:* For each $\mathcal{L}_\varepsilon$, there exist two planes $\mathcal{L}_1^\varepsilon = \{\tilde{x} \in \mathbb{R}^m : \mathbf{n}_\varepsilon^\top \tilde{x} = d_{\varepsilon t}\}$ and $\mathcal{L}_2^\varepsilon = \{\tilde{x} \in \mathbb{R}^m : \mathbf{n}_\varepsilon^\top \tilde{x} = -d_{\varepsilon t}\}$ that are parallel to $\mathcal{L}$ and tangent to the ellipse, where

$$d_{\varepsilon t} = \sqrt{\mathbf{n}_\varepsilon^\top Q^{-1} \mathbf{n}_\varepsilon}. \quad (46)$$

The corresponding tangent points are given as

$$p_1^\varepsilon = \frac{d_{\varepsilon t} Q^{-1} \mathbf{n}_\varepsilon}{\mathbf{n}_\varepsilon^\top Q^{-1} \mathbf{n}_\varepsilon} = \frac{Q^{-1} \mathbf{n}_\varepsilon}{d_{\varepsilon t}}, \quad (47)$$

$$p_2^\varepsilon = -p_1^\varepsilon. \quad (48)$$

We use these two tangent points to define a novel measure of displacement between a plane and an ellipse.

Definition 4: The *tangent interpolation point* is defined as

$$p_\mathcal{L}^\varepsilon = \frac{d_\varepsilon Q^{-1} \mathbf{n}_\varepsilon}{\mathbf{n}_\varepsilon^\top Q^{-1} \mathbf{n}_\varepsilon} \quad (49)$$

Note that $p_\mathcal{L}^\varepsilon \in \mathcal{L}$ and when $d_\varepsilon = d_{\varepsilon t}$ or $d_\varepsilon = -d_{\varepsilon t}$, $p_\mathcal{L}^\varepsilon = p_1^\varepsilon$ or $p_\mathcal{L}^\varepsilon = p_2^\varepsilon$, respectively. It is clear that when $d_\varepsilon \in [-d_{\varepsilon t}, d_{\varepsilon t}]$, the plane $\mathcal{L}$ and the ellipsoid $\mathcal{E}$ have at least one intersection, thus violating our desired reachability constraint. Using this fact, we define the following projection operator:

$$\pi_{\mathbf{n}\varepsilon}^\varepsilon(x) = \begin{cases} p_{t1}^\varepsilon - p_\mathcal{L}^\varepsilon & d_\varepsilon \in [0, d_{\varepsilon t}], \\ p_{t2}^\varepsilon - p_\mathcal{L}^\varepsilon & d_\varepsilon \in [-d_{\varepsilon t}, 0), \\ 0 & \text{otherwise.} \end{cases} \quad (50)$$

2) *ADMM constraints:* Transforming the projection π^ε back to $\mathcal{F}$ we have:

$$D_{\mathbf{n}}(x) = H^{-1}(x(t_1), x(t_2)) \pi_{\mathbf{n}\varepsilon}^\varepsilon(x) + o, \quad (51)$$

with the corresponding constraint written as

$$\zeta_{\mathbf{n}} = \{x \in \mathbb{R}^{nm} : \|D_{\mathbf{n}}(x)\| = 0\} \quad (52)$$

$$\Pi_{\mathbf{n}}(D_{\mathbf{n}}(x)) = 0 \quad (53)$$

3) *Gradients for the projection function:*

Proposition 4: The differential of the projection function $\Pi_{\mathbf{n}\varepsilon}^\varepsilon(x)$ with respect to the foci x_1 and x_2 is given by:

$$\partial_x \pi_{\mathbf{n}\varepsilon}^\varepsilon(x) = \begin{cases} \partial_x p_{t1}^\varepsilon - \partial_x p_\mathcal{L}^\varepsilon & d_\varepsilon \in [0, d_{\varepsilon t}], \\ \partial_x p_{t2}^\varepsilon - \partial_x p_\mathcal{L}^\varepsilon & d_\varepsilon \in [-d_{\varepsilon t}, 0), \\ 0 & \text{otherwise.} \end{cases} \quad (54)$$

where

$$\begin{aligned} \partial_x p_\mathcal{L} &= \left(-\frac{d_{\varepsilon t} n^\top \partial_x o - 2d_\varepsilon \partial_x d_{\varepsilon t}}{d_{\varepsilon t}^3} \right) Q^{-1} \mathbf{n}_\varepsilon \\ &\quad + \frac{d_\varepsilon \partial_b Q^{-1} \mathbf{n}_\varepsilon \partial_x b - 2d_\varepsilon Q^{-1} H[n]_\times U}{d_{\varepsilon t}^2}, \quad (55) \\ \partial_x p_1 &= -\frac{Q^{-1} \mathbf{n}_\varepsilon \partial_x d_{\varepsilon t}}{d_{\varepsilon t}^2} + \frac{\partial_b Q^{-1} \mathbf{n}_\varepsilon \partial_x b - 2Q^{-1} H[n]_\times U}{d_{\varepsilon t}}. \quad (56) \end{aligned}$$

Proof: We first need to derive $\dot{d}_\varepsilon$ and $\dot{d}_{\varepsilon t}$

$$\dot{d}_\varepsilon = -n^\top \partial_x o \dot{x} \quad (57)$$

$$\begin{aligned} \dot{d}_{\varepsilon t} &= (\dot{n}_\varepsilon^\top Q^{-1} \mathbf{n}_\varepsilon + n_\varepsilon^\top \dot{Q}^{-1} \mathbf{n}_\varepsilon + n_\varepsilon^\top Q^{-1} \dot{\mathbf{n}}_\varepsilon) / \sqrt{n_\varepsilon^\top Q^{-1} \mathbf{n}_\varepsilon} \\ &= (\sqrt{n_\varepsilon^\top Q^{-1} \mathbf{n}_\varepsilon})^{-1} (-2n^\top H[Q^{-1} \mathbf{n}_\varepsilon]_\times U \\ &\quad + n_\varepsilon^\top \partial_b Q^{-1} \mathbf{n}_\varepsilon \partial_x b - 2n_\varepsilon Q^{-1} H[n]_\times U) \dot{x} \quad (58) \end{aligned}$$

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

$$p_\mathcal{L} = \frac{d_\varepsilon Q^{-1} \mathbf{n}_\varepsilon}{d_{\varepsilon t}^2}, \quad (59)$$

we have

$$\dot{p}_\mathcal{L} = \left(\left(-\frac{d_{\varepsilon t} n^\top \partial_x o - 2d_\varepsilon \partial_x d_{\varepsilon t}}{d_{\varepsilon t}^3} \right) Q^{-1} \mathbf{n}_\varepsilon + \frac{d_\varepsilon \partial_b Q^{-1} \mathbf{n}_\varepsilon \partial_x b - 2d_\varepsilon Q^{-1} H[n]_\times U}{d_{\varepsilon t}^2} \right) \dot{x} \quad (60)$$

$$\dot{p}_1 = \left(-\frac{Q^{-1} \mathbf{n}_\varepsilon \partial_x d_{\varepsilon t}}{d_{\varepsilon t}^2} + \frac{\partial_b Q^{-1} \mathbf{n}_\varepsilon \partial_x b - 2Q^{-1} H[n]_\times U}{d_{\varepsilon t}} \right) \dot{x} \quad (61)$$

subtracting $\dot{x}$ from (60) and (61), we can derive the result shown in (49) $\blacksquare$

Based on the previous proposition, the differential of (51) can be written as:

$$\partial_x D_{\mathbf{n}\varepsilon} = -2H[\Pi_{\mathbf{n}\varepsilon}^\varepsilon]_\times M + H^{-1} \partial_x \Pi_{\mathbf{n}\varepsilon}^\varepsilon \quad (62)$$

C. *Line-segment-ellipsoid constraint*

To avoid (or stay in) a region defined by several hyperplanes, the relative position between the ellipse and segment of the hyperplane needs to be studied instead of the whole plane. Assume the segment have endpoint p_1 and p_2 , the segment could be defined as:

$$\begin{bmatrix} (p_1 - p_2)^\top \\ (p_2 - p_1)^\top \end{bmatrix} p \leq \begin{bmatrix} p_2^\top \\ -p_1^\top \end{bmatrix} (p_1 - p_2) \quad (63)$$

And using the $p_\mathcal{L}$ defined in (49), we could be defined

$$D_{seg}(x) = \begin{cases} D_1(p_1) & (p_1 - p_2)^\top (p_\mathcal{L} - p_2) < 0 \\ D_1(p_2) & (p_2 - p_1)^\top (p_\mathcal{L} - p_1) < 0 \\ D_2(p_\mathcal{L}) & \text{otherwise} \end{cases} \quad (64)$$

where $D_{p\varepsilon}$ and $D_{\mathbf{n}\varepsilon}$ are corresponding constraint function introduced in section IV-A and IV-B with the constraint set and projection written as:

$$\zeta_{seg} = \{x \in \mathbb{R}^{nm} : \|D_{seg}(x)\| = 0\} \quad (65)$$

$$\Pi_{seg}(D_{seg}(x)) = 0 \quad (66)$$

D. *Convex-region-ellipsoid constraint*

To keep an ellipse away from a convex region, first, we need to keep all segments of the hyperplanes outside the ellipse, and to prevent cases where the ellipse is a subset of the region, two foci need to be kept outside of the region using the convex obstacle constraint introduced in [9].

V. RESULTS

In this section, we test our constraints using a three-agent map exploration task in simulation. Definition of the objective function is introduced in [9]. The simulation is done using the Matlab toolbox developed in [9]. The reachability constraint is set by specifying the corresponding function $D(\mathbf{q})$, its Jacobian, and the projection operator Π_{ζ} introduced in section IV.

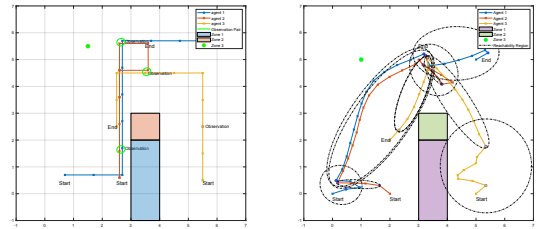
The environment is a square region with edge length of 8 meter, an obstacle *Zone 1* and two safezones *Zone 2, 3* which are shown in Fig.2a. The robots have a maximum velocity of $0.5m/s$ and a time limit of $20s$. All robots have the task of collecting sensor information on the underlying vector field. Assuming that the sensors on return data with higher accuracy for locations closer to the agent, robots should ideally perform a boustrophedon pattern. We first set up co-observation schedules considering two forbidden regions using the solver introduced in [8], and add reachability constraints in between the co-observation schedule.

The co-observation schedule generated in a grid world is shown in Fig.2a. No task has been assigned to the agent besides the start and goal location. Given the fact that agent 3 is operating alone at lower right corner, the solver will not return a secure plan unless an additional observation point is required by the solver at time 4 (either by a sensor on the field or by introducing an additional agent). Assuming agent 3 can be observed at time 4, the co-observation plan is: agent 1 and 2 meet at time 8 and 14, agent 2 and 3 meet at time 18. This result is also used as the initial trajectory input for the ADMM solver. Based on the schedule, reachability for agent 1 in-between time 0, 8, 14 and 20, agent 2 in-between time 0, 8, 18 and 20, and agent 3 in-between time 0, 8, 18 and 20 are constrained.

The result of the simulation is shown in Fig.2b. Reachability regions are shown as black ellipsoids and all of them avoid the intersection with Zone 2 and 3. Noticed that no constraint has been set between reachability regions and obstacles, assuming robots have the basic obstacle avoidance capability and can not go through any hard obstacles. Therefore, the intersections between obstacles and ellipsoids, i.e. the case here between agent 3 and zone 1, are tolerable. All constraints have been satisfied and, to the best of their capability, agents have spread out across the map to perform the best possible exploration task.

VI. CONCLUSION

In this paper, we have incorporated the reachability constraint in our ADMM based trajectory optimization algorithm. Together with the co-observation schedule generated through previous work, our planning algorithm can be used to enhance the security of a multi-agent system while still have the capability of fulfilling a relatively complex task. Simulation results are used to illustrate our method. The current method assumes a simple dynamic of the robots, future work will focus on incorporating more complex dynamics on a higher dimensional workspace.



(a) Co-observation schedule and initial trajectory generated in a 8×8 grid world. (b) Simulation results. Reachability regions are shown as black ellipsoids in the result.

Fig. 2: Simulation results.

REFERENCES

- [1] G. Pajares, "Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs)," *Photogrammetric Engineering & Remote Sensing*, vol. 81, no. 4, pp. 281–330, 2015.
- [2] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012.
- [3] M. Brunner, H. Hofinger, C. Krauß, C. Roblee, P. Schöo, and S. Todt, "Infiltrating critical infrastructures with next-generation attacks," *Fraunhofer Institute for Secure Information Technology (SIT), Munich*, 2010.
- [4] R. Stern, N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. S. Kumar, et al., "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Twelfth Annual Symposium on Combinatorial Search*, 2019.
- [5] Q. Zhu, L. Bushnell, and T. Başar, "Resilient distributed control of multi-agent cyber-physical systems," in *Control of cyber-physical systems*. Springer, 2013, pp. 301–316.
- [6] S. Gil, S. Kumar, M. Mazumder, D. Katabi, and D. Rus, "Guaranteeing spoof-resilient multi-robot networks," *Autonomous Robots*, vol. 41, no. 6, pp. 1383–1400, 2017.
- [7] Y. Shoukry, S. Mishra, Z. Luo, and S. Diggavi, "Sybil attack resilient traffic networks: A physics-based trust propagation approach," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPs)*, 2018, pp. 43–54.
- [8] 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.
- [9] Z. Yang and R. Tron, "Multi-agent path planning under observation schedule constraints," in *IEEE International Conference on Intelligent Robots and Systems*, 2020.
- [10] 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.
- [11] A. S. Householder, "Unitary triangularization of a nonsymmetric matrix," *Journal of the ACM (JACM)*, vol. 5, no. 4, pp. 339–342, 1958.
- [12] R. Tron and K. Daniilidis, "Technical report on optimization-based bearing-only visual homing with applications to a 2-d unicycle model," 2014, arXiv.
- [13] D. Eberly, "Distance from a point to an ellipse, an ellipsoid, or a hyperellipsoid." [Online]. Available: <https://www.geometrictools.com/>