Robust and Efficient Sequential Path Planning Under Adversarial Intruder

Somil Bansal*, Mo Chen*, Ken Tanabe, and Claire J. Tomlin

***Abstract*—To be written.**

I. Introduction

Reduce the number of 'we's in the paper. Recently, there has been an immense surge of interest in the use of unmanned aerial systems (UASs) for civil applications. The applications include package delivery, aerial surveillance, disaster response, among many others [1]–[5]. Unlike previous uses of UASs for military purposes, civil applications will involve unmanned aerial vehicles (UAVs) flying in urban environments, potentially in close proximity of humans, other UAVs, and other important assets. As a result, government agencies such as the Federal Aviation Administration (FAA) and National Aeronautics and Space Administration (NASA) of the United States are urgently trying to develop new scalable ways to organize an airspace in which potentially thousands of UAVs can fly together [6], [7].

One essential problem that needs to be addressed for this endeavour to be successful is that of trajectory planning: how a group of vehicles in the same vicinity can reach their destinations while avoiding situations which are considered dangerous, such as collisions. Many previous studies address this problem under different assumptions. In some studies, specific control strategies for the vehicles are assumed, and approaches such as those involving induced velocity obstacles [8]–[11] and involving virtual potential fields to maintain collision avoidance [12], [13] have been used. Methods have also been proposed for real-time trajectory generation [14], for path planning for vehicles with linear dynamics in the presence of obstacles with known motion [15], and for cooperative path planning via waypoints which do not account for vehicle dynamics [16]. Other related work include those which consider only the collision avoidance problem without path planning. These results include those that assume the system has a linear model [17]–[19], rely on a linearization of the system model [20], [21], assume a simple positional state space [22], and many others [23]–[25].

However, to make sure that a dense group of UAVs can safely fly in the close vicinity of each other, we need the capability to flexibly plan provably safe and dynamically feasible trajectories without making strong assumptions on the vehicles' dynamics and other vehicles' motion. Moreover, any trajectory planning scheme that addresses collision avoidance must also guarantee both goal satisfaction and safety of UAVs despite disturbances caused by wind and communication faults [7]. Furthermore, unexpected scenarios such as UAV malfunctions or even UAVs with malicious intent need to be accounted for. Finally, the proposed scheme should scale well with the number of vehicles.

The problem of trajectory planning and collision avoidance under disturbances in safety-critical systems has been studied using Hamilton-Jacobi (HJ) reachability analysis, which provides guarantees on goal satisfaction and safety of optimal system trajectories [26]–[31]. Reachability-based methods are particularly suitable in the context of UAVs because of the hard guarantees that are provided. In reachability analysis, one computes the reach-avoid set, defined as the set of states from which the system can be driven to a target set while satisfying (possibly time-varying) state constraints at all times. A major practical appeal of this approach stems from the availability of modern numerical tools, which can compute various definitions of reachable sets [32]–[35]. These numerical tools, for example, have been successfully used to solve a variety of differential games, path planning problems, and optimal control problems. Concrete practical applications include aircraft auto-landing [36], automated aerial refueling [37], MPC control of quadrotors [38], and multiplayer reach-avoid games [39]. Despite its power, the approach becomes numerically intractable as the state space dimension increases. In particular, reachable set computations involve solving a HJ partial differential equation (PDE) or variational inequality (VI) on a grid representing a discretization of the state space, resulting in an *exponential* scaling of computational complexity with respect to the dimensionality of the problem. Therefore, as such, dynamic programming-based approaches such as reachability analysis are not suitable for managing the next generation airspace, which is a large-scale system with a high-dimensional joint state space because of the high density of vehicles that needs to be accommodated [7].

To overcome this problem, Sequential Path Planning (SPP) method has been proposed [40], in which vehicles are assigned a strict priority ordering. Higher-priority vehicles plan their paths without taking into account the lower-priority vehicles. Lower-priority vehicles treat higher-priority vehicles as moving obstacles. Under this assumption, time-varying formulations of reachability [29], [31] can be used to obtain the optimal and provably safe paths for each vehicle, starting from the highest-priority vehicle. Thus, the curse of dimensionality is overcome for the multi-vehicle path planning problem at the cost of a mild structural assumption, under which the computation complexity scales just *linearly* with the number

This work has been supported in part by NSF under CPS:ActionWebs (CNS-931843), by ONR under the HUNT (N0014-08-0696) and SMARTS (N00014-09-1-1051) MURIs and by grant N00014-12-1-0609, by AFOSR under the CHASE MURI (FA9550-10-1-0567). The research of M. Chen has received funding from the "NSERC" program and "la Caixa" Foundation, respectively.

* Both authors contributed equally to this work. All authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. {somil, mochen72, k-tanabe, tomlin}@eecs.berkeley.edu

of vehicles. Authors in [41] and [42], respectively, extend this method to the scenarios where disturbances and adversarial intruders are present in the system, resolving some of the practical challenges associated with the basic SPP algorithm in [40]. The focus of these works, however, have mostly been on the theoretical development of SPP algorithm. Our focus in this work is instead on effectively demonstrating the linear scaling of SPP algorithm and showcasing its potential as a trajectory planning algorithm for large scale systems. In particular, our main contributions in this work are:

- we simulate a large-scale SPP system for two different space structures: when SPP algorithm is used for trajectory planning at a city level and at a regional level. For city level planning, we consider the San Francisco city in California, US and for regional level planning we consider the entire Bay area in California, US. The main differences emerge from the fact that the city level planning needs to take into account physcial obstacles (like buildings, etc.), and the origins and destinations are farther apart at a regional level.
- we demonstrate the kind and the number of different vehicle trajectories that emerge out of a large scale SPP system between a given pair of origin and destination. Furthermore, we show how these trajectories change under different wind conditions and priority orderings.
- we also show the reactivity of the control law obtained from SPP algorithm, e.g. the obtained control law is able to effectively counter the disturbances without requiring any communication with other vehicles.

On the theoretical side, one of the limitations of the SPP algorithm proposed in [42] to account for adversarial intruders is that the algorithm might need to replan the trajectories for all SPP vehicles once the intruder disappears. Since the replanning is done in real-time, the proposed algorithm may not be scalable with the number of SPP vehicles, rendering the method unsuitable for practical implementation. In this work, we also propose a novel intruder avoidance algorithm, which will need to replan trajectories only for a fixed number of vehicles, irrespective of the total number of SPP vehicles, thus overcoming the limitations of the algorithm in [42]. Moreover, this number is a design parameter, which can be chosen based on the computational resources available during the run time. Finally, we illustrate this algorithm through a simulation at the city level.

Rest of the paper is organized as follows: in Section II, we formally present the SPP problem in the presence of disturbances. In Section III, we present a brief review of time-varying reachability and the Robust Trajectory Tracking (RTT) method proposed in [41] to account for disturbances. We also use this algorithm for all our simulations in this paper. In Section IV, a novel algorithm to account for intruders has been proposed. All simulation results are in Section V.

II. Sequential Path Planning Problem

Consider N vehicles (also denoted as *SPP vehicles*) which participate in the SPP process $Q_i, i = 1, \dots, N$. We assume their dynamics are given by

$$\begin{aligned} &\dot{x}_i = f_i(x_i, u_i, d_i), t \leq t_i^{\text{STA}} \\ &u_i \in \mathcal{U}_i, d_i \in \mathcal{D}_i, i = 1 \dots, N \end{aligned} \tag{1}$$

where $x_i \in \mathbb{R}^{n_i}$, $u_i \in \mathcal{U}_i$ and $d_i \in \mathcal{D}_i$, respectively, represent the state, control and disturbance experienced by vehicle Q_i. We partition the state x_i into the position component $p_i \in \mathbb{R}^{n_p}$ and the non-position component $h_i \in \mathbb{R}^{n_i - n_p}$: $x_i = (p_i, h_i)$. We will use the sets $\mathbb{U}_i, \mathbb{D}_i$ to respectively denote the set of functions from which the control and disturbance functions $u_i(\cdot), d_i(\cdot)$ are drawn.

Each vehicle Q_i has initial state x_i^0, and aims to reach its target $\mathcal{L}_i$ by some scheduled time of arrival t_i^{STA}. The target in general represents some set of desirable states, for example the destination of Q_i. On its way to $\mathcal{L}_i$, Q_i must avoid a set of static obstacles $\mathcal{O}_i^{\text{static}} \subset \mathbb{R}^{n_i}$. The interpretation of $\mathcal{O}_i^{\text{static}}$ could be a tall building or any set of states that are forbidden for each SPP vehicle.

In addition to the static obstacles, each vehicle Q_i must also avoid the danger zones with respect to every other vehicle $Q_j, j \neq i$. The danger zones in general can represent any joint configurations between Q_i and Q_j that are considered to be unsafe. We define the danger zone of Q_i with respect to Q_j to be

$$\mathcal{Z}_{ij} = \{(x_i, x_j) : \|p_i - p_j\|_2 \leq R_c\} \tag{2}$$

whose interpretation is that Q_i and Q_j are considered to be in an unsafe configuration when they are within a distance of R_c of each other. In particular, Q_i and Q_j are said to have collided, ifif $(x_i, x_j) \in \mathcal{Z}_{ij}$.

Given the set of SPP vehicles, their targets $\mathcal{L}_i$, the static obstacles $\mathcal{O}_i^{\text{static}}$, and the vehicles' danger zones with respect to each other $\mathcal{Z}_{ij}$, our goal is, for each vehicle Q_i, to synthesize a controller which guarantees that Q_i reaches its target $\mathcal{L}_i$ at or before the scheduled time of arrival t_i^{STA}, while avoiding the static obstacles $\mathcal{O}_i^{\text{static}}$ as well as the danger zones with respect to all other vehicles $\mathcal{Z}_{ij}, j \neq i$. In addition, we would like to obtain the latest departure time t_i^{LDT} such that Q_i can still arrive at $\mathcal{L}_i$ on time.

In general, the above optimal path planning problem must be solved in the joint space of all N SPP vehicles. However, due to the high joint dimensionality, a direct dynamic programming-based solution is intractable. Therefore, authors in [40] proposed to assign a priority to each vehicle, and perform SPP given the assigned priorities. Without loss of generality, let Q_j have a higher priority than Q_i if $j < i$. Under the SPP scheme, higher-priority vehicles can ignore the presence of lower-priority vehicles, and perform path planning without taking into account the lower-priority vehicles' danger zones. A lower-priority vehicle Q_i, on the other hand, must ensure that it does not enter the danger zones of the higher-priority vehicles $Q_j, j < i$; each higher-priority vehicle Q_j induces a set of time-varying obstacles $\mathcal{O}_i^j(t)$, which represents the possible states of Q_i such that a collision between Q_i and Q_j could occur.

It is straight-forward to see that if each vehicle Q_i is able to plan a trajectory that takes it to $\mathcal{L}_i$ while avoiding the static obstacles $\mathcal{O}_i^{\text{static}}$ and the danger zones of *higher-priority vehicles* $Q_j, j < i$, then the set of SPP vehicles $Q_i, i = 1, \ldots, N$ would all be able to reach their targets safely. Under the SPP scheme, path planning can be done sequentially in descending order of vehicle priority in the state space of only a single vehicle. Thus, SPP provides a solution whose complexity scales linearly with the number of vehicles, as opposed to exponentially with a direct application of dynamic programming approaches.

III. Background

The basic SPP algorithm presented in [40] ignores disturbances in vehicles' dynamics and assumes perfect information of vehicles' positions. However, in presence of disturbances, it is no longer possible to commit to exact trajectories (and hence positions), since the disturbance $d_i(\cdot)$ is *a priori* unknown. Thus, disturbances and incomplete information significantly complicate the SPP scheme. In this section, we present the robust trajectory tracking algorithm (proposed in [41]) that can be used to make basic SPP approach robust to disturbances as well as to an imperfect knowledge of other vehicles' positions. In the next section, we will present an algorithm to further robustify the SPP approach by considering how the set of SPP vehicles may respond to the presence of an intruder vehicle which may be adversarial. All of these algorithms use time-varying reachability analysis to provide goal satisfaction and safety guarantees; therefore, we start with an overview of time-varying reachability.

A. Time-Varying Reachability Background

We will be using reachability analysis to compute either a backward reachable set (BRS) $\mathcal{V}$, a forward reachable set (FRS) $\mathcal{W}$, or a sequence of BRSs and FRSs, given some target set $\mathcal{L}$, time-varying obstacle $\mathcal{G}(t)$, and the Hamiltonian function H which captures the system dynamics as well as the roles of the control and disturbance. The BRS $\mathcal{V}$ in a time interval $[t, t_f]$ or FRS $\mathcal{W}$ in a time interval $[t_0, t]$ will be denoted by

$$\begin{aligned} &\mathcal{V}(t, t_f) \quad \text{(backward reachable set)} \\ &\mathcal{W}(t_0, t) \quad \text{(forward reachable set)} \end{aligned} \tag{3}$$

Several formulations of reachability are able to account for time-varying obstacles [29], [31] (or state constraints in general). For our application in SPP, we utilize the time-varying formulation in [31], which accounts for the time-varying nature of systems without requiring augmentation of the state space with the time variable. In the formulation in [31], a BRS is computed by solving the following *final value* double-obstacle HJ VI:

$$\begin{aligned} \max\Big\{ &\min\{D_t V(t,x) + H(t, x, \nabla V(t,x)), l(x) - V(t,x)\}, \\ &- g(t,x) - V(t,x)\Big\} = 0, \quad t \le t_f \\ &V(t_f, x) = \max\{l(x), -g(t_f, x)\} \end{aligned} \tag{4}$$

In a similar fashion, the FRS is computed by solving the following *initial value* HJ PDE:

$$\begin{aligned} D_t W(t,x) + H(t,x,\nabla W(t,x)) = 0, \quad t \ge t_0 \\ W(t_0, x) = \max\{l(x), -g(t_0, x)\} \end{aligned} \tag{5}$$

In both (4) and (5), the function $l(x)$ is the implicit surface function representing the target set $\mathcal{L} = \{x : l(x) \le 0\}$. Similarly, the function $g(t,x)$ is the implicit surface function representing the time-varying obstacles $\mathcal{G}(t) = \{x : g(t,x) \le 0\}$. The BRS $\mathcal{V}(t, t_f)$ and FRS $\mathcal{W}(t_0, t)$ are given by

$$\begin{aligned} \mathcal{V}(t, t_f) = \{x : V(t,x) \le 0\} \\ \mathcal{W}(t_0, t) = \{x : W(t,x) \le 0\} \end{aligned} \tag{6}$$

Some of the reachability computations will not involve an obstacle set $\mathcal{G}(t)$, in which case we can simply set $g(t,x) \equiv \infty$ which effectively means that the outside maximum is ignored in (4). Also, note that unlike in (4), there is no inner minimization in (5). As we will see later, we will be using the BRS to determine all states that can reach some target set *within the time horizon* $[t, t_f]$, whereas we will be using the FRS to determine where a vehicle could be *at some particular time* t. In addition, (5) has no outer maximum, since the FRSs that we will compute will not involve any obstacles.

The Hamiltonian, $H(t, x, \nabla V(t,x))$, depends on the system dynamics, and the role of control and disturbance. Whenever H does not depend explicit on t, we will drop the argument. In addition, the Hamiltonian is an optimization that produces the optimal control $u^*(t,x)$ and optimal disturbance $d^*(t,x)$, once V is determined. For BRSs, whenever the existence of a control ("$\exists u$") or disturbance is sought, the optimization is a minimum over the set of controls or disturbance. Whenever a BRS characterizes the behavior of the system for all controls ("$\forall u$") or disturbances, the optimization is a maximum. We will introduce precise definitions of reachable sets, expressions for the Hamiltonian, expressions for the optimal controls as needed for the many different reachability calculations we use.

B. Robust Trajectory Tracking (RTT)

In the basic SPP algorithm, lower priority vehicles know the trajectories of all higher priority vehicles. The region that a lower priority vehicle needs to avoid is thus simply given by the danger zones around these trajectories; however, disturbances and incomplete information significantly complicate the SPP scheme. Committing to exact trajectories is no longer possible, since the disturbance $d_i(\cdot)$ is *a priori* unknown. Thus, the induced obstacles $\mathcal{O}_i^j(t)$ are no longer just the danger zones centered around positions. In this section, we provide an overview of the RTT algorithm that can overcome these issues. For simplicity of explanation, we will assume that no static obstacles exist, but method can be generalized even when static obstacles do exist [41]. The material in this section is taken partially from [41]. Note that even though the theoretical development in Section IV is valid for all three algorithms developed in [41] to account for the disturbances, we use RTT algorithm for the simulations in this paper and only present

RTT algorithm here. Interested readers are referred to [41] for the other two algorithms.

Even though it is impossible to commit to and track an exact trajectory in the presence of disturbances, it may still be possible to instead *robustly* track a feasible *nominal* trajectory with a bounded error at all times. If this can be done, then the tracking error bound can be used to determine the induced obstacles. Here, computation is done in two phases: the *planning phase* and the *disturbance rejection phase*.

In the planning phase, a nominal trajectory $x_{r,j}(\cdot)$ is computed that is feasible in the absence of disturbances. This planning is done for a reduced control set $\mathcal{U}^p \subset \mathcal{U}$, as some margin is needed to reject unexpected disturbances while tracking the nominal trajectory.

In the disturbance rejection phase, we compute a bound on the tracking error, independently of the nominal trajectory. To compute this error bound, we find a robust controlled-invariant set in the joint state space of the vehicle and a tracking reference that may "maneuver" arbitrarily in the presence of an unknown bounded disturbance. Taking a worst-case approach, the tracking reference can be viewed as a virtual evader vehicle that is optimally avoiding the actual vehicle to enlarge the tracking error. We therefore can model trajectory tracking as a pursuit-evasion game in which the actual vehicle is playing against the coordinated worst-case action of the virtual vehicle and the disturbance.

Let x_j and $x_{r,j}$ denote the states of the actual vehicle Q_j and the virtual evader, respectively, and define the tracking error $e_j = x_j - x_{r,j}$. When the error dynamics are independent of the absolute state as in (7) (and also (7) in [27]), we can obtain error dynamics of the form

$$\begin{aligned} \dot{e_j} &= f_{e_j}(e_j, u_j, u_{r,j}, d_j), \\ u_j &\in \mathcal{U}_j, u_{r,j} \in \mathcal{U}_j^p, d_j \in \mathcal{D}_j, \quad t \leq 0 \end{aligned} \tag{7}$$

To obtain bounds on the tracking error, we first conservatively estimate the error bound around any reference state $x_{r,j}$, denoted $\mathcal{E}_j$:

$$\mathcal{E}_j = \{e_j : \|p_{e_j}\|_2 \leq R_{\text{EB}}\}, \tag{8}$$

where p_{e_j} denotes the position coordinates of e_j and R_{EB} is a design parameter. We next solve a reachability problem with its complement $\mathcal{E}_j^c$, the set of tracking errors violating the error bound, as the target in the space of the error dynamics. From $\mathcal{E}_j^c$, we compute the following BRS:

$$\begin{aligned} \mathcal{V}_j^{\text{EB}}(t, 0) = &\{y : \forall u_j(\cdot) \in \mathbb{U}_j, \exists u_{r,j}(\cdot) \in \mathbb{U}_j^p, \exists d_j(\cdot) \in \mathbb{D}_i, \\ &e_j(\cdot) \text{ satisfies (7)}, e_j(t) = y, \\ &\exists s \in [t, 0], e_j(s) \in \mathcal{E}_j^c\}, \end{aligned} \tag{9}$$

where the Hamiltonian to compute the BRS is given by:

$$H_j^{\text{EB}}(e_j, \lambda) = \max_{u_j \in \mathcal{U}_j} \min_{u_r \in \mathcal{U}_j^p, d_j \in \mathcal{D}_j} \lambda \cdot f_{e_j}(e_j, u_j, u_{r,j}, d_j). \tag{10}$$

Letting $t \to -\infty$, we obtain the infinite-horizon control-invariant set $\Omega_j := \lim_{t \to -\infty} \left(\mathcal{V}_j^{\text{EB}}(t, 0)\right)^c$. If Ω_j is nonempty, then the tracking error e_j at flight time is guaranteed to remain within $\Omega_j \subseteq \mathcal{E}_j$ provided that the vehicle starts inside Ω_j and subsequently applies the feedback control law

$$\kappa_j(e_j) = \arg\max_{u_j \in \mathcal{U}_j} \min_{u_r \in \mathcal{U}_j^p, d_j \in \mathcal{D}_j} \lambda \cdot f_{e_j}(e_j, u_j, u_{r,j}, d_j). \tag{11}$$

The induced obstacles by each higher-priority vehicle Q_j can thus be obtained by:

$$\begin{aligned} \mathcal{O}_i^j(t) &= \{x_i : \exists y \in \mathcal{P}_j(t), \|p_i - y\|_2 \leq R_c\} \\ \mathcal{P}_j(t) &= \{p_j : \exists h_j, (p_j, h_j) \in \mathcal{M}_j(t)\} \\ \mathcal{M}_j(t) &= \Omega_j + x_{r,j}(t), \end{aligned} \tag{12}$$

where the "+" in (12) denotes the Minkowski sum[1]. Intuitively, if Q_j is tracking $x_{r,j}(t)$, then it will remain within the error bound Ω_j around $x_{r,j}(t)$ $\forall t$. This is precisely the set $\mathcal{P}_j(t)$. The induced obstacles can then be obtained by augmenting a danger zone around this set. Finally, we can obtain the total obstacle set $\mathcal{G}_i(t)$ using:

$$\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}} \cup \bigcup_{j=1}^{i-1} \mathcal{O}_i^j(t) \tag{13}$$

Since each vehicle Q_j, $j < i$, can only be guaranteed to stay within Ω_j, we must make sure during the path planning of Q_i that at any given time, the error bounds of Q_i and Q_j, Ω_i and Ω_j, do not intersect. This can be done by augmenting the total obstacle set by Ω_i:

$$\tilde{\mathcal{G}}_i(t) = \mathcal{G}_i(t) + \Omega_i. \tag{14}$$

Finally, given Ω_i, we can guarantee that Q_i will reach its target $\mathcal{L}_i$ if $\Omega_i \subseteq \mathcal{L}_i$; thus, in the path planning phase, we modify $\mathcal{L}_i$ to be $\tilde{\mathcal{L}}_i := \{x_i : \Omega_i + x_i \subseteq \mathcal{L}_i\}$, and compute a BRS, with the control authority $\mathcal{U}_i^p$, that contains the initial state of the vehicle. Mathematically,

$$\begin{aligned} \mathcal{V}_i^{\text{rtt}}(t, t_i^{\text{STA}}) = &\{y : \exists u_i(\cdot) \in \mathbb{U}_i^p, x_i(\cdot) \text{ satisfies (18)}, \\ &\forall s \in [t, t_i^{\text{STA}}], x_i(s) \notin \tilde{\mathcal{G}}_i(t), \\ &\exists s \in [t, t_i^{\text{STA}}], x_i(s) \in \tilde{\mathcal{L}}_i, x_i(t) = y\} \end{aligned} \tag{15}$$

The BRS $\mathcal{V}_i^{\text{rtt}}(t, t_i^{\text{STA}})$ can be obtained by solving (4) using the Hamiltonian:

$$H_i^{\text{rtt}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i^p} \lambda \cdot f_i(x_i, u_i) \tag{16}$$

The corresponding optimal control for reaching $\tilde{\mathcal{L}}_i$ is given by:

$$u_i^{\text{rtt}}(t) = \arg\min_{u_i \in \mathcal{U}_i^p} \lambda \cdot f_i(x_i, u_i). \tag{17}$$

The nominal trajectory $x_{r,i}(\cdot)$ can thus be obtained by using vehicle dynamics in the absence of disturbances

$$\begin{aligned} \dot{x}_i &= f_i(x_i, u_i), t \leq t_i^{\text{STA}} \\ u_i &\in \mathcal{U}_i, \qquad i = 1 \ldots, N, \end{aligned} \tag{18}$$

with the optimal control $u_i^{\text{rtt}}(\cdot)$ given by (17). From the resulting nominal trajectory $x_{r,i}(\cdot)$, the overall control policy to reach $\mathcal{L}_i$ can be obtained via (11).

[1]The Minkowski sum of sets A and B is the set of all points that are the sum of any point in A and B.

IV. Response to Intruders

In Section III-B, we presented RTT algorithm that can take into account disturbances in vehicles' dynamics. However, if a vehicle not in the set of SPP vehicles enters the system, or even worse, if this vehicle is an adversarial intruder, the original plan can lead to vehicles entering into each other's danger zones. If vehicles do not plan with an additional safety margin that takes a potential intruder into account, a vehicle trying to avoid the intruder may effectively become an intruder itself, leading to a domino effect.

In general, the effect of an intruder on the vehicles in structured flight can be entirely unpredictable, since the intruder in principle could be adversarial in nature, and the number of intruders could be arbitrary. Therefore, we make the following two assumptions:

Assumption 1: At most one intruder (denoted as Q_I here on) affects the SPP vehicles at any given time. The intruder exits the altitude level affecting the SPP vehicles after a duration of t^{IAT}.

Let the time at which intruder appears in the system be $\underline{t}$ and the time at which it disappears be $\bar{t}$. Assumption 1 implies that $\bar{t} \leq \underline{t} + t^{\text{IAT}}$. Thus, any vehicle Q_i would need to avoid the intruder Q_I for a maximum duration of t^{IAT}. Note that we do not pose any restriction on $\underline{t}$; however, we assume that once the intruder appears, it stays for a maximum duration of t^{IAT}.

Assumption 2: The dynamics of the intruder are known and given by $\dot{x}_I = f_I(x_I, u_I, d_I)$.

Assumption 2 is required for HJ reachability analysis. In situations where the dynamics of the intruder are not known exactly, a conservative model of the intruder may be used instead. We also denote the initial state of the intruder as x_I^0. Note that x_I^0 is unknown.

Our goal is to design a control policy that ensures separation with the intruder and other SPP vehicles, and ensures a successful transit to the destination. However, depending on the initial state of the intruder and its control policy, a vehicle may arrive at different states after avoiding the intruder. Therefore, a control policy that ensures a successful transit to the destination needs to account for all such possible states, which is a path planning problem with multiple initial states and a single destination, and is hard to solve in general. Thus, we divide the intruder avoidance problem into two sub-problems: (i) we first design a control policy that ensures a successful transit to the destination if no intruder appears and that successfully avoid the intruder, if it does. (ii) after the intruder disappears at $\bar{t}$, we replan the trajectories of the affected vehicles. Following the same theme and assumptions, authors in [42] present an algorithm to avoid an intruder in SPP formulation; however, once the intruder disappears, the algorithm might need to replan the trajectories for all SPP vehicles. Since the replanning is done in real-time, it should be fast and scalable with the number of SPP vehicles, rendering the method in [42] unsuitable for practical implementation.

In this work, we propose a novel intruder avoidance algorithm, which will need to replan trajectories only for a fixed number of vehicles, irrespective of the total number of SPP vehicles. Moreover, this number is a design parameter, which can be chosen based on the resources available during the run time. Intuitively, one can think about dividing the flight space of vehicles such that at any given time, any two vehicles are far enough from each other so that an intruder can only affect atmost $\bar{k}$ vehicles in a duration of t^{IAT} despite its best efforts. In this method, we build upon this intuition and show that such a division of space is indeed possible. The advantage of such an approach is that after the intruder disappears, we only have to replan the trajectory of *atmost* $\bar{k}$ vehicles, which can be efficiently done in real-time if $\bar{k}$ is low enough, thus making this approach particularly suitable for practical systems.

In Sections IV-A, we discuss the intruder avoidance control and explain the sensing range required to avoid the intruder. In Sections IV-B and IV-C, we compute a space division of state-space such that atmost $\bar{k}$ vehicles need to apply the avoidance maneuver computed in Section IV-A, regardless of the initial state of the intruder. Trajectopry planning and replanning are discussed in Sections IV-D and IV-E respectively.

A. Optimal Avoidance Controller

To compute the optimal avoidance control for Q_i in presence of Q_I, we compute the set of states from which the joint states of Q_I and Q_i can enter danger zone $\mathcal{Z}_{iI}$ despite the best efforts of Q_i to avoid Q_I.

We define relative dynamics of the intruder Q_I with state x_I with respect to Q_i with state x_i:

$$\begin{aligned} x_{I,i} &= x_I - x_i \\ \dot{x}_{I,i} &= f_r(x_{I,i}, u_i, u_I, d_i, d_I) \end{aligned} \tag{19}$$

Given the relative dynamics, we compute the set of states from which the joint states of Q_I and Q_i can enter danger zone $\mathcal{Z}_{iI}$ in a duration of t^{IAT} despite the best efforts of Q_i to avoid Q_I. Under the relative dynamics (19), this set of states is given by the backwards reachable set $\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}}),\ \tau \in [0, t^{\text{IAT}}]$:

$$\begin{aligned} \mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}}) = \{ y : & \forall u_i(\cdot) \in \mathbb{U}_i, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & \exists d_I(\cdot) \in \mathbb{D}_I, x_{I,i}(\cdot) \text{ satisfies (19)}, \\ & \exists s \in [\tau, t^{\text{IAT}}], x_{I,i}(s) \in \mathcal{L}_i^{\text{A}}, x_{I,i}(\tau) = y \}, \end{aligned} \tag{20}$$

where

$$\begin{aligned} \mathcal{L}_i^{\text{A}} &= \{ x_{I,i} : \|p_{I,i}\|_2 \leq R_c \} \\ H_i^{\text{A}}(x_{I,i}, \lambda) &= \max_{u_i \in \mathcal{U}_i} \Big(\min_{u_I \in \mathcal{U}_I, d_i \in \mathcal{D}_i, d_I \in \mathcal{D}_I} \lambda \cdot f_r(x_{I,i}, u_i, u_I, d_i, d_I) \Big) \end{aligned} \tag{21}$$

We refer to $\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$ as *avoid region* hereon. Once the value function $V_i^{\text{A}}(\cdot)$ is computed, the optimal avoidance control u_i^{A} can be obtained as:

$$u_i^{\text{A}} = \arg\max_{u_i \in \mathcal{U}_i} \left(\min_{u_I \in \mathcal{U}_I, d_i \in \mathcal{D}_i, d_I \in \mathcal{D}_I} \lambda \cdot f_r(x_{I,i}, u_i, u_I, d_i, d_I) \right) \tag{22}$$

Let $\partial \mathcal{V}_i^{\text{A}}(\cdot, t^{\text{IAT}})$ denotes the boundary of the set $\mathcal{V}_i^{\text{A}}(\cdot, t^{\text{IAT}})$. The interpretation of $\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$ is that if $x_{I,i}(t) \in \partial \mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$, then Q_i can successfully avoid the intruder for a duration of $(t^{\text{IAT}} - \tau)$ using the optimal avoidance control in (22). In the worst case, Q_i might need to avoid the intruder for a duration of t^{IAT}; thus, we must have that $x_{I,i}(\underline{t}) \in \left(\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})\right)^C$. Equivalently, every SPP vehicle should be able to detect the intruder at a distance of d^{A}, where

$$d^{\text{A}} = \max\{\|p_i\| : \exists h_i, (p_i, h_i) \in \mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})\}. \tag{23}$$

Hereon, we assume that every SPP vehicle can detect the intruder at a distance of d^{A}.

B. Separation and Buffer Regions- Case1

In the next two sections, our goal is to ensure that any two vehicles are separated enough from each other so that atmost $\bar{k}$ vehicles should be *forced* to apply avoidnace maneuver during the duration $[\underline{t}, \bar{t}]$. To capture this mathematically, we define

$$\mathcal{A}_m := \{t : x_{I,m}(t) \in \mathcal{V}_m^{\text{A}}(t - \underline{t}, t^{\text{IAT}}), t \in [\underline{t}, \bar{t}]\}$$

$\mathcal{A}_m$ is the set of all times at which Q_m is forced to apply an intruder avoidance maneuver. We also define *avoid start time*, $\underline{t}_m$, for Q_m as:

$$\underline{t}_m = \begin{cases} \min_{t \in \mathcal{A}_m} t & \text{if } \mathcal{A}_m \neq \emptyset \\ \infty & \text{otherwise} \end{cases} \tag{24}$$

Therefore, $\underline{t}_m \in [\underline{t}, \bar{t}]$ denotes the first time at which Q_m applies an avoidance maneuver and defined to be ∞ if Q_m never applies an avoidance maneuver. Therefore, if we ensure that

$$\forall i \neq j, min(\underline{t}_i, \underline{t}_j) < \infty \implies |\underline{t}_i - \underline{t}_j| \geq \frac{t^{\text{IAT}}}{\bar{k}} := t^{\text{BRD}}, \tag{25}$$

then atmost $\bar{k}$ vehicles are forced to apply the avoidance maneuver during the time interval $[\underline{t}, \bar{t}]$. We refer to the condition in (25) as *separation requirement* hereon.

For any given time t, if we could find the set of all states of Q_i such that the separation requirement holds for Q_i and Q_m for all $m < i$ and for all intruder strategies, then during the path planning of Q_i, we can ensure that Q_i is in one of these states at time t. The sequential path planning will therefore guarantee that the separation requirement holds for every SPP vehicle pair. Thus, hereon we focus on finding all states $x_i(t)$ such that the separation requirements are ensured between vehicles Q_j and Q_i, $j < i$ at time t for all possible intruder scenarios (meaning all possible $\underline{t}, \bar{t}, x_I^0$ and $u_I(\cdot)$). For our analysis, we consider the following two mutually exclusive and exhaustive cases: $\underline{t}_j \leq \underline{t}_i, \underline{t}_j < \infty$ and $\underline{t}_i < \underline{t}_j, \underline{t}_i < \infty$. In this section, we consider Case1: $\underline{t}_j \leq \underline{t}_i, \underline{t}_j < \infty$. Case2 is discussed in the next section.

In Case1, the intruder forces Q_j to apply avoidance control before or at the same time as Q_i. To ensure the separation requirement in this case, we begin with the following observation which narrows down the intruder scenarios that we need to consider:

Observation 1: Without loss of generality, we can assume that the intruder appears at the boundary of the avoid region of Q_j, e.g. $x_{I,j}(\underline{t}) \in \partial\mathcal{V}_j^{\text{A}}(0, t^{\text{IAT}})$. Otherwise, vehicles Q_j and Q_i need not account for the intruder until it reaches the boundary of the avoid region of Q_j. Also note that since $\infty < \underline{t}_j \leq \underline{t}_i$, Q_I reaches the boundary of the avoid region of Q_j first. Equivalently, we can assume that $\underline{t}_j = \underline{t}$.

1) Separation region: Consider any $\underline{t} \in \mathbb{R}$. In this section, we find the set of all states $x_I^0 := x_I(\underline{t})$ for which Q_j is forced to apply an avoidnace maneuver. We refer to this set as *separation region*, and denote it as $\mathcal{S}_j(\underline{t})$. As discussed in Section IV-A, Q_j needs to apply avoidnace maneuver at time $\underline{t}$ only if $x_{I,j}(\underline{t}) \in \partial\mathcal{V}_j^{\text{A}}(0, t^{\text{IAT}})$. To compute set $\mathcal{S}_j(\underline{t})$, we thus need to translate this set in relative coordinates to a set in absolute coordinates, for which we need to know all possible states of Q_j at time $\underline{t}$.

Depending on the information known to a lower-priority vehicle Q_i about Q_j's control strategy, we can use one of the three methods described in Section 5 in [42] to compute the "base" obstacles $\mathcal{M}_j(\underline{t})$, the obstacles that would have been induced by Q_j in the presence of disturbances, but in the absence of an intruder. The base obstacles are respectively given by equations (25), (31) and (37) in [42] for centralized control, least restrictive control and robust trajectory tracking algorithms.

Given $\mathcal{M}_j(\underline{t})$, $\mathcal{S}_j(\underline{t})$ can be obtained as:

$$\mathcal{S}_j(\underline{t}) = \mathcal{M}_j(\underline{t}) + \partial\mathcal{V}_j^{\text{A}}(0, t^{\text{IAT}}), \ \underline{t} \in \mathbb{R}, \tag{26}$$

where the "+" in (26) denotes the Minkowski sum. Since $\mathcal{S}_j(\underline{t})$ represents the set of all states of Q_I for which Q_j is forced to apply an avoidnace maneuver, we have $x_I(\underline{t}) \in \mathcal{S}_j(\underline{t})$ in Case1.

2) Buffer Region: From Section IV-B1, we know that $x_I^0 := x_I(\underline{t}) \in \mathcal{S}_j(\underline{t}) := \mathcal{S}_j(\underline{t}_j)$, where last equivalence holds because $\underline{t}_j = \underline{t}$ (by Observation 1). Moroever, since avoid start time of Q_i is $\underline{t}_i$, we must have $x_{I,i}(\underline{t}_i) \in \partial\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$.

In this section, we first compute the set of all relative states $\mathcal{V}_i^{\text{B}}$ from which it is possible to reach $\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$ for some control applied by Q_I and Q_i. If we augment this relative coordinates set $\mathcal{V}_i^{\text{B}}$ on $\mathcal{S}_j(\underline{t})$ to get $\mathcal{B}_{ij}(\underline{t})$, then we can guarantee that Q_I cannot reach the boundary (or inside) of $\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$ before $t = \underline{t} + t^{\text{BRD}}$ irrespective of the control applied by Q_I and Q_i during $[\underline{t}, t]$, if $x_I(\underline{t}) \in \mathcal{S}_j(\underline{t})$ and $x_i(\underline{t}) \in \left(\mathcal{B}_{ij}(\underline{t})\right)^C$. Equivalently, $(\underline{t}_i - \underline{t}_j) \geq t^{\text{BRD}}$, if $x_i(\underline{t}) \in \left(\mathcal{B}_{ij}(\underline{t})\right)^C$ ($x_I(\underline{t}) \in \mathcal{S}_j(\underline{t})$ from Section IV-B1 and $\underline{t} = \underline{t}_j$ by Observation 1).

We refer to the set $\mathcal{B}_{ij}(\underline{t})$ as *buffer region* hereon. To compute $\mathcal{B}_{ij}(\underline{t})$, we first compute $\mathcal{V}_i^{\text{B}}$ which is given by the following BRS:

$$\begin{aligned} \mathcal{V}_i^{\text{B}}(0, t^{\text{BRD}}) = \{ & y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & \exists d_I(\cdot) \in \mathbb{D}_I, x_{i,I}(\cdot) \text{ satisfies (19)}, \\ & \exists s \in [0, t^{\text{BRD}}], x_{i,I}(s) \in -\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}}), \\ & x_{i,I}(t) = y\}, \end{aligned} \tag{27}$$

where

$$H_i^{\text{B}}(x_{i,I}, \lambda) = \min_{u_i \in \mathcal{U}_i, u_I \in \mathcal{U}_I, d_i \in \mathcal{D}_i, d_I \in \mathcal{D}_I} \lambda \cdot f_r(x_{i,I}, u_i, u_I, d_i, d_I) \tag{28}$$

Note that we use $-\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$ as the target set for our computation above because we are computing BRS $\mathcal{V}_i^{\text{B}}(0, t^{\text{BRD}})$ using the relative state $x_{i,I}$ and not $x_{I,i}$. Also, since Q_I appears for a maximum duration of t^{IAT}, it is sufficient to ensure that Q_I and Q_i do not reach the set $\mathcal{V}_i^{\text{A}}(t^{\text{BRD}}, t^{\text{IAT}})$

instead of $\mathcal{V}_i^{\mathrm{A}}(0, t^{\mathrm{IAT}})$, as a duration of t^{BRD} is already spent by Q_I to reach the avoid region of Q_i.

The buffer region can not be computed by augmenting $\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$ on the separation region of the Q_j:

$$\mathcal{B}_{ij}(\underline{t}) = \mathcal{S}_j(\underline{t}) + \mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}}). \tag{29}$$

We can thus ensure that $(\underline{t}_i - \underline{t}_j) \geq t^{\mathrm{BRD}}$ as long as $x_i(\underline{t}) \in (\mathcal{B}_{ij}(\underline{t}))^C$.

3) Obstacle Computation: In sections IV-B1 and IV-B2, we computed a separation between Q_i and Q_j such that if $\underline{t}_i - \underline{t}_j \geq t^{\mathrm{BRD}}$. However, we need to make sure that while applying avoidance control these vehicles do not enter in each other's danger zones or collide with the static obstacles. In this section, we compute the obstacles that reflect this possibility. In particular, we want to find the set of states that Q_i needs to avoid to avoid entering in the danger zone of Q_j. To find such states, we consider the following two cases:

1) CaseA: intruder affects Q_j, but not Q_i.
2) CaseB: intruder first affects Q_j and then Q_i.

For each case, we compute the set of states that Q_i needs to avoid at time t to avoid entering in $\mathcal{Z}_{ji}$. Let ${}^A_1\mathcal{O}_i^j(\cdot)$ and ${}^B_1\mathcal{O}_i^j(\cdot)$ denote the set of obstacles corresponding to CaseA and CaseB respectively. We begin with the following observation:

Observation 2: To compute obstacles at time t, it is sufficient to consider the scenarios where $\underline{t} \in [t - t^{\mathrm{IAT}}, t]$. This is because if $\underline{t} < t - t^{\mathrm{IAT}}$, then Q_j and/or Q_i will already be in the replanning phase at time t (see assumption 1) and hence the two vehicles cannot be in conflict at time t. On the other hand, if $\underline{t} > t$, then Q_j wouldn't apply any avoidance maneuver at time t.

- CaseA: Note that since $\underline{t}_j = \underline{t}$ (by Observation 1), ${}^A_1\mathcal{O}_i^j(\cdot)$ is given by the states that Q_j can reach while avoiding the intruder, starting from some state in $\mathcal{M}_j(\underline{t}), \underline{t} \in [t - t^{\mathrm{IAT}}, t]$. These states can be obtained by computing a FRS from the base obstacles.

$$\begin{aligned} \mathcal{W}_j^{\mathcal{O}}(\underline{t}, t) = \{ & y : \exists u_j(\cdot) \in \mathbb{U}_j, \exists d_j(\cdot) \in \mathbb{D}_j, \\ & x_j(\cdot) \text{ satisfies (1)}, x_j(\underline{t}) \in \mathcal{M}_j(\underline{t}), \\ & x_j(t) = y\}. \end{aligned} \tag{30}$$

$\mathcal{W}_j^{\mathcal{O}}(\underline{t}, t)$ represents the set of all possible states that Q_j can reach after a duration of $(t - \underline{t})$ starting from inside $\mathcal{M}_j(\underline{t})$. This FRS can be obtained by solving the HJ VI in (5) with the following Hamiltonian:

$$H_j^{\mathcal{O}}(x_j, \lambda) = \max_{u_j \in \mathcal{U}_j} \max_{d_j \in \mathcal{D}_j} \lambda \cdot f_j(x_j, u_j, d_j). \tag{31}$$

Since $\underline{t} \in [t - t^{\mathrm{IAT}}, t]$, the induced obstacles in this case can be obtained as:

$$\begin{aligned} &{}^A_1\mathcal{O}_i^j(t) = \{x_i : \exists y \in \mathcal{P}_j(t), \|p_i - y\|_2 \leq R_c\} \\ &\mathcal{P}_j(t) = \{p_j : \exists h_j, (p_j, h_j) \in \bigcup_{\underline{t} \in [t - t^{\mathrm{IAT}}, t]} \mathcal{W}_j^{\mathcal{O}}(\underline{t}, t)\} \end{aligned} \tag{32}$$

Observation 3: Since the base obstacles represent possible states that a vehicle can be in in the absence of an intruder, the base obstacle at any time τ_2 is contained within the FRS of the base obstacle at any time $\tau_1 (< \tau_2)$, computed forward for a duration of $(\tau_2 - \tau_1)$. That is, $\mathcal{M}_j(\tau_2) \subseteq \mathcal{W}_j^{\mathcal{O}}(\tau_1, \tau_2)$, where $\mathcal{W}_j^{\mathcal{O}}(\tau_1, \tau_2)$, as before, denotes the FRS of $\mathcal{M}_j(\tau_1)$ computed forward for a duration of $(\tau_2 - \tau_1)$. The same argument can be applied for the FRS computed from $\mathcal{M}_j(\tau_2)$ and $\mathcal{M}_j(\tau_1)$, i.e. $\mathcal{W}_j^{\mathcal{O}}(\tau_2, \tau_3) \subseteq \mathcal{W}_j^{\mathcal{O}}(\tau_1, \tau_3)$, where $\tau_1 < \tau_2 < \tau_3$.

Using observation 3, $\mathcal{P}_j(t)$ in (32) can be equivalently written as:

$$\mathcal{P}_j(t) = \{p_j : \exists h_j, (p_j, h_j) \in \mathcal{W}_j^{\mathcal{O}}(t - t^{\mathrm{IAT}}, t)\}. \tag{33}$$

- CaseB: In this case, the intruder first affects Q_j and then Q_i. Therefore, we have $\underline{t}_j, \underline{t}_i \in [\underline{t}, \bar{t}]$. Once Q_j starts applying avoidance control at time $\underline{t} = \underline{t}_j$, it might deviate from its planned trajectory. Thus from the perspective of Q_i, Q_j can apply any control during $[\underline{t}, \bar{t}]$. Furthermore, Q_i itself might need to apply avoidnace maneuver during $[\underline{t}_i, \bar{t}]$. Thus, the main challenge in this case is to ensure that Q_i and Q_j do not enter each other's danger zones even when both have deviated from their planned trajectory and hence can apply *any* control from each other's perspective. Thus at time t, Q_i not only need no avoid the states that Q_j could be in at time t, but also all the states that could lead it to $\mathcal{Z}_{ji}$ *in future* under some control actions of Q_i and Q_j. To compute this set of states, we make the following key observation:

Observation 4: For computing ${}^B_1\mathcal{O}_i^j(t)$, it is sufficient to consider $\underline{t}_i = t$. If $\underline{t}_i > t$, then Q_i is not applying any avoidance maneuver at time t and hence should only avoid the states that Q_j could be in at time t. However, this is already ensured during computation of ${}^A_1\mathcal{O}_i^j(t)$. If $\underline{t}_i < t$, then for a given $\underline{t}$, Q_i still needs to avoid the same set of states at time t, as it would have if $\underline{t}_i = t$.

Due to the separation and buffer regions, we have $\underline{t}_i - \underline{t}_j \geq t^{\mathrm{BRD}}$. Combining this with Observations 2 and 4 implies that $\underline{t}_j \in [t - t^{\mathrm{IAT}}, t - t^{\mathrm{BRD}}]$. From the perspective of Q_i, Q_j can reach any state in $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, t')$ at time $t' \in [\underline{t}_j, \underline{t}_j + t^{\mathrm{IAT}}]$ starting from some state in $\mathcal{M}_j(\underline{t}_j)$ at time $\underline{t}_j$. Here, $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, t')$ represents the FRS of $\mathcal{M}_j(\underline{t}_j)$ computed for a duration of $(t' - \underline{t}_j)$ and is given by (30). Taking into account all possible $\underline{t}_j \in [t - t^{\mathrm{IAT}}, t - t^{\mathrm{BRD}}]$, $x_j(\tau)$ is contained in the set:

$$\mathcal{K}^{\mathrm{B1}}(\tau) = \bigcup_{\underline{t}_j \in [\tau - t^{\mathrm{IAT}}, t - t^{\mathrm{BRD}}]} \mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau) \tag{34}$$

at time $\tau \in [t, t - t^{\mathrm{BRD}} + t^{\mathrm{IAT}}]$. From Observation 3, we have $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau) \subseteq \mathcal{W}_j^{\mathcal{O}}(\tau - t^{\mathrm{IAT}}, \tau)$ for all $\underline{t}_j \in [\tau - t^{\mathrm{IAT}}, t - t^{\mathrm{BRD}}]$. Therefore, $\mathcal{K}^{\mathrm{B1}}(\tau) = \mathcal{W}_j^{\mathcal{O}}(\tau - t^{\mathrm{IAT}}, \tau)$.

From the perspective of Q_i, it needs to avoid all states at time t that can reach $\mathcal{K}^{\mathrm{B1}}(\tau)$ for some control action of Q_i during time duration $[t, \tau]$. This will ensure that Q_i and Q_j will not enter into each other's danger zones regardless of the avoidance maneuver applied by them. This set of states is given by the following BRS:

$$\begin{aligned} \mathcal{V}_i^{\mathrm{B1}}(t, t - t^{\mathrm{BRD}} + t^{\mathrm{IAT}}) = \{ & y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y, \\ & \exists s \in [t, t - t^{\mathrm{BRD}} + t^{\mathrm{IAT}}], \\ & x_i(s) \in \tilde{\mathcal{K}}^{\mathrm{B1}}(s)\}, \end{aligned} \tag{35}$$

where

$$\tilde{\mathcal{K}}^{\text{B1}}(s) = \{x_j : \exists (y,h) \in \mathcal{K}^{\text{B1}}(s), \|p_j - y\|_2 \leq R_c\}.$$

The Hamiltonian H_i^{B1} to compute $\mathcal{V}_i^{\text{B1}}(\cdot)$ is given by:

$$H_i^{\text{B1}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i} \min_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \tag{36}$$

Finally, the induced obstacle in this case can be obtained as:

$${}_1^B\mathcal{O}_i^j(t) = \mathcal{V}_i^{\text{B1}}(t, t - t^{\text{BRD}} + t^{\text{IAT}}). \tag{37}$$

C. Separation and Buffer Regions- Case2

In this section, we consider Case2: $\underline{t}_i < \underline{t}_j, \underline{t}_i < \infty$. In this case, the intruder forces Q_i to apply avoidance control before Q_j. We can make a similar observation to that of 1 in this case.

Observation 5: Without loss of generality, we can assume that $x_{I,i}(\underline{t}) \in \partial\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$. Equivalently, we can assume that $\underline{t}_i = \underline{t}$.

1) Separation region: Similar to Section IV-C1, we want to compute the set of all states $x_I(\underline{t}_j)$ for which Q_j is forced to apply an avoidnace maneuver. Since, Q_j applies teh avoidance maneuver after Q_i in this case, Q_j will need to avoid the intruder for a maximum duration of $t^{\text{RD}} := t^{\text{IAT}} - t^{\text{BRD}}$. $\mathcal{S}_j(\underline{t}_j)$ can thus be obtained as:

$$\mathcal{S}_j(\underline{t}_j) = \mathcal{M}_j(\underline{t}_j) + \partial\mathcal{V}_j^{\text{A}}(0, t^{\text{RD}}). \tag{38}$$

2) Buffer Region: We now want to compute the set of all states x_I such that if Q_I starts in this set at time t, it cannot reach $\mathcal{S}_j(\cdot)$ before $t_1 = t + t^{\text{BRD}}$, regardless of the control applied by Q_j and Q_I during interval $[t, t_1]$. Similar to Section IV-B2, this set is given by $\mathcal{V}_j^{\text{B}}(0, t^{\text{BRD}})$:

$$\begin{aligned}\mathcal{V}_j^{\text{B}}(0, t^{\text{BRD}}) = \{y : \; & \exists u_j(\cdot) \in \mathbb{U}_j, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_j(\cdot) \in \mathbb{D}_j, \\ & \exists d_I(\cdot) \in \mathbb{D}_I, x_{I,j}(\cdot) \text{ satisfies (19)}, \\ & \exists s \in [0, t^{\text{BRD}}], x_{I,j}(s) \in \mathcal{V}_j^{\text{A}}(t^{\text{BRD}}, t^{\text{IAT}}), \\ & x_{I,j}(t) = y\},\end{aligned} \tag{39}$$

where

$$H_j^{\text{B}}(x_{I,j}, \lambda) = \min_{u_j \in \mathcal{U}_j, u_I \in \mathcal{U}_I, d_j \in \mathcal{D}_j, d_I \in \mathcal{D}_I} \lambda \cdot f_r(x_{I,j}, u_j, u_I, d_j, d_I) \tag{40}$$

In absolute coordinates, we thus have that if the intruder starts outside $\tilde{\mathcal{B}}_{ji}(t) = \mathcal{M}_j(t) + \mathcal{V}_j^{\text{B}}(0, t^{\text{BRD}})$ at time t, then it cacnot reach $\mathcal{S}_j(\cdot)$ before time $t + t^{\text{BRD}}$. Finally, if we can ensure that the avoid region of Q_i at time t is outside $\tilde{\mathcal{B}}_{ji}(t)$, then $x_{I,i}(\underline{t}) := x_{I,i}(\underline{t}_i) \in \partial\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$ implies that $\underline{t}_j - \underline{t}_i \geq t^{\text{BRD}}$. Mathematically, if we define the set,

$$\mathcal{B}_{ji}(\underline{t}) = \mathcal{M}_j(\underline{t}) + \mathcal{V}_j^{\text{B}}(0, t^{\text{BRD}}) + \left(-\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})\right), \tag{41}$$

then $(\underline{t}_j - \underline{t}_i) \geq t^{\text{BRD}}$ as long as $x_i(\underline{t}) \in (\mathcal{B}_{ji}(\underline{t}))^C$.

3) Obstacle Computation: In this section, we want to find the set of states that Q_i needs to avoid to avoid entering in the danger zone of Q_j. We consider the following two mutually exclusive and exhaustive cases:

1) CaseA: intruder affects Q_i, but not Q_j.
2) CaseB: intruder first affects Q_i and then Q_j.

For each case, we compute the set of states that Q_i needs to avoid at time t to avoid entering in $\mathcal{Z}_{ji}$. We also let ${}_2^A\mathcal{O}_i^j(\cdot)$ and ${}_2^B\mathcal{O}_i^j(\cdot)$ denote the set of obstacles corresponding to CaseA and CaseB respectively.

- CaseA: In this case, we need to ensure that Q_i does not collide with the obstacle set $\mathcal{M}_j(\cdot)$ while it is avoiding the intruder. We begin with the following observation:

  Observation 6: By Observation 2, it is sufficient to consider the scenarios where $\underline{t} = \underline{t}_i \in [t - t^{\text{IAT}}, t]$. Since Q_i can be forced to apply an avoidance maneuver for a duration of $[\underline{t}_i, \underline{t}_i + t^{\text{IAT}}]$, to compute obstacles at time t for a given $\underline{t}_i$, we need to make sure that Q_i avoid all states at time t that can lead to a collision with Q_j during the interval $[t, \underline{t}_i + t^{\text{IAT}}]$ for some avoidnace control. Therefore, it is sufficient to consider the scenario $\underline{t}_i = t$ as that will maximize the avoidnace duration $[t, \underline{t}_i + t^{\text{IAT}}]$ for the obstacle computation at time t.

  Mathematically, Q_i needs to avoid all states at time t that can reach $\mathcal{K}^{\text{A2}}(\tau)$ for some control action of Q_i during time duration $[t, \tau]$. $\mathcal{K}^{\text{A2}}(\tau)$ here is given by:

$$\begin{aligned}\mathcal{K}^{\text{A2}}(\tau) =& \tilde{\mathcal{M}}_j(s), \\ \tilde{\mathcal{M}}_j(s) =& \{x_j : \exists (y,h) \in \mathcal{M}_j(s), \|p_j - y\|_2 \leq R_c\}.\end{aligned} \tag{42}$$

  Note that since the intruder is present in the system for a maximum duration of t^{IAT} and since $\underline{t}_i = t$ (by Observation 6), we have that $\tau \in [t, t + t^{\text{IAT}}]$.
  Avoiding $\mathcal{K}^{\text{A2}}(\cdot)$ will ensure that Q_i and Q_j will not enter into each other's danger zones regardless of the avoidance maneuver applied by them. The set of states that Q_i needs to avoid at time t is given by the following BRS:

$$\begin{aligned}\mathcal{V}_i^{\text{A2}}(t, t + t^{\text{IAT}}) = \{y : \; & \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y, \\ & \exists s \in [t, t + t^{\text{IAT}}], x_i(s) \in \mathcal{K}^{\text{A2}}(s)\}.\end{aligned} \tag{43}$$

  The Hamiltonian H_i^{A2} to compute $\mathcal{V}_i^{\text{A2}}(t, t + t^{\text{IAT}})$ is given by:

$$H_i^{\text{A2}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i} \min_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \tag{44}$$

  Finally, the induced obstacle in this case is given as

$${}_2^A\mathcal{O}_i^j(t) = \mathcal{V}_i^{\text{A2}}(t, t + t^{\text{IAT}}). \tag{45}$$

- CaseB: In this case, the intruder first affects Q_i and then Q_j. Q_i and Q_j apply their first avoidnace maneuver at $\underline{t}_i$ and $\underline{t}_j$ respectively. From the perspective of Q_i, Q_j can apply any control during the duration $[\underline{t}_j, \underline{t}_i + t^{\text{IAT}}]$ and hence can be anywhere in the set $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau)$ at $\tau \in [\underline{t}_j, \underline{t}_i + t^{\text{IAT}}]$, where $\mathcal{W}_j^{\mathcal{O}}$ denotes the FRS of base obstacle $\mathcal{M}_j(\underline{t}_j)$ computed forward for a duration of $(\underline{t}_i + t^{\text{IAT}} - \underline{t}_j)$. Q_i thus needs to make sure that it avoids all states at time t that can reach

$\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau)$, regardless of the control applied by Q_i during $[t, \tau]$. We now make the following key observation:

Observation 7: Observation 3 implies that $\mathcal{W}_j^{\mathcal{O}}(\tau_2, \tau) \subseteq \mathcal{W}_j^{\mathcal{O}}(\tau_1, \tau)$ if $\tau > \tau_2 > \tau_1$. Therefore, the biggest obstacle is induced by Q_j at τ if $\underline{t}_j$ is as early as possible and it is sufficient to avoid this obstacle to ensure collision avoidnace with Q_j at time τ. Given the separation and buffer regions between Q_i and Q_j, we must have $\underline{t}_j - \underline{t}_i \geq t^{\text{BRD}}$. Hence, the biggest obstacle is induced by Q_j when $\underline{t}_j = \underline{t}_i + t^{\text{BRD}}$. Intuitively, Observation 7 implies that the biggest obstacle is induced by Q_j when intruder forces Q_i to apply the avoidance maneuver and *immediately* "travels" through the buffer region between two vehicles to affect Q_j after a duration of t^{BRD}. Therefore, Q_i needs to avoid $\mathcal{K}^{\text{B2}}(\tau)$ at time $\tau > t$, where

$$\mathcal{K}^{\text{B2}}(\tau) = \bigcup_{\underline{t}_i \in [t - t^{\text{IAT}}, t], \tau - \underline{t}_i \leq t^{\text{IAT}}} \mathcal{W}_j^{\mathcal{O}}(\underline{t}_i + t^{\text{BRD}}, \tau), \tau > t, \tag{46}$$

where we have substituted $\underline{t}_j = \underline{t}_i + t^{\text{BRD}}$. In (47) $\underline{t} = \underline{t}_i \in [t - t^{\text{IAT}}, t]$ (By Observation 2) and $\tau - \underline{t}_i \leq t^{\text{IAT}}$ because the intruder can appear for a maximum duration of t^{IAT}. (47) can be equivalently written as:

$$\begin{aligned}
\mathcal{K}^{\text{B2}}(\tau) &= \bigcup_{\underline{t}_i \in [\tau - t^{\text{IAT}}, t]} \mathcal{W}_j^{\mathcal{O}}(\underline{t}_i + t^{\text{BRD}}, \tau), t < \tau \leq t + t^{\text{IAT}} \\
\mathcal{K}^{\text{B2}}(\tau) &= \mathcal{W}_j^{\mathcal{O}}(\tau - t^{\text{IAT}} + t^{\text{BRD}}, \tau), t < \tau \leq t + t^{\text{IAT}},
\end{aligned} \tag{47}$$

where the second equality holds because of Observation 3. The set of states that that Q_i needs to avoid at time t is thus given by the following BRS:

$$\begin{aligned}
\mathcal{V}_i^{\text{B2}}(t, t + t^{\text{IAT}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i, \\
& x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y, \\
& \exists s \in [t + t^{\text{BRD}}, t + t^{\text{IAT}}], x_i(s) \in \tilde{\mathcal{K}}^{\text{B2}}(s)\}, \\
\tilde{\mathcal{K}}^{\text{B2}}(s) =& \{x_i : \exists (y, h) \in \mathcal{K}^{\text{B2}}(s), \|p_i - y\|_2 \leq R_c\}.
\end{aligned} \tag{48}$$

The Hamiltonian H_i^{B2} to compute $\mathcal{V}_i^{\text{B2}}(t, t + t^{\text{IAT}})$ is given by:

$$H_i^{\text{B2}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i} \min_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \tag{49}$$

Finally, the induced obstacle in this case is given as

$${}_2^B\mathcal{O}_i^j(t) = \mathcal{V}_i^{\text{B2}}(t, t + t^{\text{IAT}}). \tag{50}$$

D. Path Planning

In sections IV-B and IV-C, we computed obstacles for Q_i such that Q_i and Q_j do not collide with each other while avodiing intruder. We next compute the states that Q_i needs to avoid to avoid a collision with static obstacles, regardless of the avoidance maneuver applied by it. Since Q_i applies avoidnace maneuver for a maximum duration of t^{IAT} this set is given by the following BRS:

$$\begin{aligned}
\mathcal{V}_i^{\text{S}}(t, t + t^{\text{IAT}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i, \\
& x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y, \\
& \exists s \in [t, t + t^{\text{IAT}}], x_i(s) \in \mathcal{K}^{\text{S}}(s)\}, \\
\mathcal{K}^{\text{S}}(s) =& \{x_i : \exists (y, h) \in \mathcal{O}_i^{\text{static}}, \|p_i - y\|_2 \leq R_c\}.
\end{aligned} \tag{51}$$

The Hamiltonian H_i^{S} to compute $\mathcal{V}_i^{\text{S}}(t, t + t^{\text{IAT}})$ is given by:

$$H_i^{\text{S}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i} \min_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \tag{52}$$

The overall obstacle for Q_i is thus given by:

$$\begin{aligned}
\mathcal{G}_i(t) = & \bigcup_{j=1}^{i-1} \left(\mathcal{B}_{ij}(t) \cup \mathcal{B}_{ji}(t) \bigcup_{k \in \{1,2\}} {}_k^A\mathcal{O}_i^j(t) \bigcup_{k \in \{1,2\}} {}_k^B\mathcal{O}_i^j(t) \right) \\
& \bigcup \mathcal{V}_i^{\text{S}}(t, t + t^{\text{IAT}}).
\end{aligned} \tag{53}$$

Given $\mathcal{G}_i(t)$, we compute a BRS $\mathcal{V}_i^{\text{AO}}(t, t_i^{\text{STA}})$ for path planning that contains the initial state of Q_i while avoiding these obstacles:

$$\begin{aligned}
\mathcal{V}_i^{\text{PP}}(t, t_i^{\text{STA}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \forall d_i(\cdot) \in \mathbb{D}_i, \\
& x_i(\cdot) \text{ satisfies (1)}, \forall s \in [t, t_i^{\text{STA}}], x_i(s) \notin \mathcal{G}_i(s), \\
& \exists s \in [t, t_i^{\text{STA}}], x_i(s) \in \mathcal{L}_i, x_i(t) = y\}.
\end{aligned} \tag{54}$$

The Hamiltonian H_i^{PP} to compute BRS in (54) is given by:

$$H_i^{\text{PP}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i} \max_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \tag{55}$$

Note that $\mathcal{V}_i^{\text{PP}}(\cdot)$ ensures goal satisfaction for Q_i in the absence of intruder. The goal satisfaction controller is given by:

$$u_i^{\text{PP}}(t, x_i) = \arg \min_{u_i \in \mathcal{U}_i} \max_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \tag{56}$$

Moreover, if Q_i starts within $\mathcal{V}_i^{\text{PP}}$ and use avoidance control strategy in (22), it is guaranteed to avoid collision with the intruder and other SPP vehicles, regardless of the control strategy of Q_I. Finally, since we use separation and buffer regions as obstacles during the path planning of Q_i, it is guaranteed that $|\underline{t}_i - \underline{t}_j| \geq t^{\text{BRD}}$ for all $j < i$. The overall control policy for avoiding the intruder and collision with other vehicles is thus given by:

$$u_i^*(t) = \begin{cases} u_i^{\text{PP}}(t) & t \leq \underline{t} \\ u_i^{\text{A}}(t) & \underline{t} \leq t \leq \bar{t} \end{cases}$$

Remark 1: Note that if we use the robust trajectory tracking method to compute the base obstacles, we would need to augment the obstacles in (53) by the error bound of Q_i, Ω_i, as discussed in section III-B.

E. Replanning after intruder avoidance

Intruder can force some SPP vehicles to deviate from their planned trajectory; therefore, we have to replan the trajectory of these vehicles once Q_I disappears. The set of all vehicles Q_i for whom we need to replan the trajectories, $\mathcal{N}^{\text{RP}}$, can be obtained by checking if a vehicle Q_i applied any avoidnace control during $[\underline{t}, \bar{t}]$, e.g.,

$$\mathcal{N}^{\text{RP}} = \{Q_i : \underline{t}_i < \infty, i \in \{1, \ldots, N\}\}. \tag{57}$$

Note that due to the presence of separation and buffer regions, atmost $\bar{k}$ vehicles can be affected by Q_I, e.g. $|\mathcal{N}^{\text{RP}}| \leq \bar{k}$. Goal satisfaction controllers which ensure that these vehicles reach their destinations can be obtained by solving a new SPP problem, where the starting states of the vehicles are now given by the states they end up in, denoted

$\tilde{x}_i^0$, after avoiding the intruder. Let the optimal control policy corresponding to this liveness controller be denoted $u_i^{\text{L}}(t, x_i)$. The overall control policy that ensures intruder avoidance, collision avoidance with other vehicles, and successful transition to the destination for vehicles in $\mathcal{N}^{\text{RP}}$ is given by:

$$u_i^{\text{RP}}(t) = \begin{cases} u_i^*(t, x_i) & t \leq \bar{t} \\ u_i^{\text{L}}(t, x_i) & t > \bar{t} \end{cases}$$

Note that in order to re-plan using a SPP method, we need to determine feasible t_i^{STA} for all vehicles. This can be done by computing an FRS:

$$\begin{aligned} \mathcal{W}_i^{\text{RP}}(\bar{t}, t) =&\{y \in \mathbb{R}^{n_i} : \exists u_i(\cdot) \in \mathbb{U}_i, \forall d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, x_i(\bar{t}) = \tilde{x}_i^0, \\ & x_i(t) = y, \forall s \in [\bar{t}, t], x_i(s) \notin \mathcal{G}_i^{\text{RP}}(s)\}, \end{aligned} \tag{58}$$

where $\tilde{x}_i^0$ represents the state of Q_i at $t = \bar{t}$; $\mathcal{G}_i^{\text{RP}}(\cdot)$ takes into account the fact that Q_i now needs to avoid all other vehicles $Q_j \notin \mathcal{N}^{\text{RP}}$ and is defined in a way analogous to (13).

The FRS in (58) can be obtained by solving

$$\begin{aligned} \max \Big\{ & D_t W_i^{\text{RP}}(t, x_i) + H_i^{\text{RP}}(t, x_i, \nabla W_i^{\text{RP}}(t, x_i)), \\ & - g_i^{\text{RP}}(t, x_i) - W_i^{\text{RP}}(t, x_i) \Big\} = 0 \\ W_i^{\text{RP}}(\underline{t}, x_i) &= \max\{l_i^{\text{RP}}(x_i), -g_i^{\text{RP}}(\underline{t}, x_i)\} \\ H_i^{\text{RP}}(x_i, \lambda) &= \max_{u_i \in \mathcal{U}_i} \min_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \end{aligned} \tag{59}$$

where $W_i^{\text{RP}}, g_i^{\text{RP}}, l_i^{\text{RP}}$ represent the FRS, obstacles during re-planning, and the initial state of Q_i, respectively. The new t_i^{STA} of Q_i is now given by the earliest time at which $\mathcal{W}_i^{\text{RP}}(\bar{t}, t)$ intersects the target set $\mathcal{L}_i$, $t_i^{\text{STA}} := \arg\inf_t\{\mathcal{W}_i^{\text{RP}}(\bar{t}, t) \cap \mathcal{L}_i \neq \emptyset\}$. Intuitively, this means that there exists a control policy which will steer the vehicle Q_i to its destination by that time, despite the worst case disturbance it might experience.

Remark 2: Note that even though we have presented the analysis for one intruder, the proposed method can handle multiple intruders as long as only one intruder is present *at any given time*.

We conclude this section by summarizing the overall SPP algorithm:

Algorithm 1: **Intruder Avoidance algorithm (offline planning)**: Given initial conditions x_i^0, vehicle dynamics (1), intruder dynamics in Assumption 2, target sets $\mathcal{L}_i$, and static obstacles $\mathcal{O}_i^{\text{static}}, i = 1 \ldots, N$, for each i,

1) determine the separation region and buffer regions given by (26), (29) and (41).
2) compute the induced obstacles for Q_i by Q_j, given by (32), (37), (45), (50) and (51).
3) compute the total obstacle set $\mathcal{G}_i(t)$, given in (53). In the case $i = 1$, $\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}}\ \forall t$;
4) given $\mathcal{G}_i(t)$, compute the BRS $\mathcal{V}_i^{\text{PP}}(t, t_i^{\text{STA}})$ defined in (54);
5) the optimal control to avoid the intruder can be obtained by using (22);

Intruder Avoidance algorithm (online re-planning): For each vehicle $Q_i \in \mathcal{N}^{\text{RP}}$ which performed avoidance in response to the intruder,

1) compute $\mathcal{W}_i^{\text{RP}}(\bar{t}, t)$ using (58). The new t_i^{STA} for Q_i is given by $\arg\inf_t\{\mathcal{W}_j^{\text{RP}}(\bar{t}, t) \cap \mathcal{L}_j \neq \emptyset\}$;
2) given t_i^{STA}, $\tilde{x}_i^0$, vehicle dynamics (1), target set $\mathcal{L}_i$, and obstacles $\mathcal{G}_i^{\text{RP}}$, use any of the three SPP methods discussed in [42] for re-planning.

V. Simulations

We now illustrate the proposed algorithm using a fifty-vehicle example.

A. Setup

For this example, we will use the following dynamics for each vehicle:

$$\begin{aligned} \dot{p}_{x,i} &= v_i \cos\theta_i + d_{x,i} \\ \dot{p}_{y,i} &= v_i \sin\theta_i + d_{y,i} \\ \dot{\theta}_i &= \omega_i, \\ \underline{v} \leq v_i \leq \bar{v}, &|\omega_i| \leq \bar{\omega}, \|(d_{x,i}, \quad d_{y,i})\|_2 \leq d_r, \end{aligned} \tag{60}$$

where $x_i = (p_{x,i}, p_{y,i}, \theta_i)$ is the state of vehicle Q_i, $p_i = (p_{x,i}, p_{y,i})$ is the position, θ_i is the heading, and $d = (d_{x,i}, d_{y,i})$ represents Q_i's disturbances, for example wind, that affect its position evolution. The control of Q_i is $u_i = (v_i, \omega_i)$, where v_i is the speed of Q_i and ω_i is the turn rate; both controls have a lower and upper bound.

Our goal is to simulate a scenario where UAVs are flying through an urban environment. This setup can be representative of many UAV applications, such as package delivery, aerial surveillance, etc. For this purpose, we grid San Francisco (SF) city in California, US and use it as our state space, as shown in Figure **??**. Each box in Figure **??** represents a $500m \times 500m$ area of SF. The origin point for the vehicles is denoted by the Blue star. Four different areas in the city are chosen as the destinations for the vehicles. Mathematically, the target sets $\mathcal{L}_i$ of the vehicles are circles of radius r in the position space, i.e. each vehicle is trying to reach some desired set of positions. In terms of the state space x_i, the target sets are defined as

$$\mathcal{L}_i = \{x_i : \|p_i - c_i\|_2 \leq r\} \tag{61}$$

where c_i are centers of the target circles. In this simulation, we use $r = 100m$. The four targets are represented by four circles in Figure **??**. The destination of each vehicle is chosen randomly from these four destinations. Finally, tall buildings in downtown San Frnacisco are used as static obstacles for SPP vehicles, denoted by black contours in Figure **??**.

To make our simulations as close as possible to real scenarios, we choose velocity and turn-rate bounds as $\underline{v} = 0m/s, \bar{v} = 25m/s, \bar{\omega} = 2rad/s$, aligned with the modern UAV specs [43], [44]. The disturbance bound is chosen as $d_r = 11m/s$, which corresponds to *strong winds* on Beaufort wind force scale [45]. Note that we have used same dynamics and input bounds across all vehicles for clarity of illustration; however, our method can easily handle more general systems of the form in which the vehicles have different control bounds and dynamics.

The goal of the vehicles is to reach their destinations while avoiding a collision with the other vehicles or the static obstacles. The vehicles also need to account for the possibility of the presence of an intruder for a maximum duration of $t^{\text{IAT}} = 10s$, whose dynamics are given by (60). The joint state space of this fifty-vehicle system is 150-dimensional (150D), making the joint path planning and collision avoidance problem intractable for direct analysis using HJ reachability. Therefore, we assign a priority order to vehicles and solve the path planning problem sequentially. For this simulation, we assign a random priority order to fifty vehicles and use the algorithm proposed in Section IV to compute a separation between SPP vehicles so that they do not collide with each other or the intruder.

B. Results

In this section, we present the simulation results for different scenarios: $\bar{k} = 2$ and $\bar{k} = 3$. For the computation of base obsatcles, we use RTT method [41]. In RTT method, a nominal trajectory is declared by the higher priority vehicles, which is then guaranteed to be tracked with some known error bound in the presence of disturbances. The base obstacles are thus given by a "bubble" around the nominal trajectory. For further details of RTT method, we refer the interested readers to Section 4C in [41]. The tracking error bound obtained for the strong wind conditions is $35m$. Thus, the base obstacles induced by a higher priority vehicle are given by a circle of radius of $35m$ around the nominal trajectory, the trajectory that a vehicle will follow if the intruder never appears in the system. The base obstacle $\mathcal{M}$ around the nominal tarjectory point $(0, 0, 0)$ is shown in Figure **??**.

We next compute set $\mathcal{V}^{\text{A}}(0, t^{\text{IAT}})$ as defined in (20), which is shown in Figure **??**. Given $\mathcal{M}$ and $\mathcal{V}^{\text{A}}(0, t^{\text{IAT}})$, we compute the separation region $\mathcal{S}$ as defined in (26), shown in Figure **??**. Note that we dropped the vehicle subscript from the sets as vehicle dynamics and hence sets are same for all vehicles. $\mathcal{V}^{\text{B}}(0, t^{\text{BRD}})$, defined in (39), is computed next and shown in Figure **??** for $\bar{k} = 3$, e.g. $t^{\text{BRD}} = 10/3$. Finally, we compute the buffer region as defined in (29). Same process was repeated for $\bar{k} = 2$ and $\bar{k} = 4$. The results are shown in Figure **??**. As evident from Figure **??**, a bigger buffer region is required between vehicles when $\bar{k}$ is smaller. Intuitively, when $\bar{k}$ is smaller, a larger buffer is required to ensure that the intruder spends more time "traveling" through this buffer region so that it can affect fewer vehicles.

We similarly sequentially computed the obstacles induced by the higher priority vehicles for each vehicle, i.e. $\mathcal{G}(\cdot)$, and the corresponding nominal trajectories, obtained by executing the control policy $u^{\text{PP}}(\cdot)$ defined in (56). The nominal trajectory thus corresponds to the trajectory that a vehicle will follow if the intruder does not appear in the system. The nominal trajectories and the overall obstacles for different vehicles at time $t = XYZ$ are shown in Figure **??** for the case $\bar{k} = 3$. The nominal trajectories are well separated from each other to ensure safe transition even during a worst-case intruder "attack". Accounting for this worst-case behavior makes our reachability analysis conservative and this limitation is discussed more Section V-C. Note that in the absence of an intruder the vehicles transit successfully to their destinations with control policy $u^{\text{PP}}(\cdot)$.

Under the proposed algorithm, the intruder will affect maximum number of vehicles ($\bar{k}$ vehicles), when it appears at the boundary of the avoid region of a vehicle, immediately "travels" through the buffer region between vehicles and reach the boundary of the avoid region of another vehicle at $\underline{t} + t^{\text{BRD}}$ and then the boundary of the avoid region of another vehicle at $\underline{t} + 2t^{\text{BRD}}$ and so on. This strategy will make sure that the intruder affects $\bar{k}$ vehicles during a duration of t^{IAT}. However, the buffer region between vehicles is computed under the assumption that both the SPP vehicle and the intruder are trying to collide with each other, but that is not necessarily true so it is very likely that the intruder will affect less than $\bar{k}$ vehicles even with this best strategy to affect maximum vehicles. This is also evident from Figures **??** and **??**. Add the explanation of what is happening in figures once we have the figures. Make time points and vehicle numbers precise.

In Case-1, the intruder forces all 3 vehicles to apply an avoidance maneuver so we need to replan the trajectories of $3 (= \bar{k})$ vehicles once the intruder disappears. However, a vehicle applies control policy $u^{\text{PP}}(\cdot)$ while the intruder is traveling through the buffer region, which may not necessarily correspond to the policy that the vehicle will use to *deliberately* collide with the intruder, unlike assumed during the computation of the set $\mathcal{V}^{\text{B}}(0, t^{\text{BRD}})$. Thus, at $\underline{t} + t^{\text{BRD}}$, the intruder may not reach the boundary of the avoid region of Q_{XY}.

Finally, in Figure **??**, we show that how the avoidance control can cause a deviation from the nominal trajectory. As evident from the figure, if vehicle doesn't apply the avoidnace maneuver and continue to track the nominal trajectory in the presence of an intruder, this might lead to a collision with the intruder. On the other hand, if the vehicle applies the avoidance policy $u^{\text{A}}(\cdot)$, it can successfully avoid a collision with the intruder.

C. Discussion

Need to add following things:

- Conservatism due to worst-case intruder and disturbance attack
- Conservatism due to the min-min set

References

[1] B. P. Tice, "Unmanned Aerial Vehicles – The Force Multiplier of the 1990s," *Airpower Journal*, 1991.

[2] W. DeBusk, "Unmanned Aerial Vehicle Systems for Disaster Relief: Tornado Alley," in *AIAA Infotech@Aerospace 2010*. American Institute of Aeronautics and Astronautics, Apr. 2010, pp. 2010–3506.

[3] Amazon.com, Inc., "Amazon Prime Air," 2016. [Online]. Available: http://www.amazon.com/b?node=8037720011

[4] AUVSI News, "UAS Aid in South Carolina Tornado Investigation," 2016. [Online]. Available: http://www.auvsi.org/blogs/auvsi-news/2016/01/29/tornado

[5] BBC Technology, "Google plans drone delivery service for 2017," 2016. [Online]. Available: http://www.bbc.com/news/technology-34704868

[6] Joint Planning and Development Office, "Unmanned Aircraft Systems (UAS) Comprehensive Plan," Federal Aviation Administration, Tech. Rep., 2014.

[7] T. Prevot, J. Rios, P. Kopardekar, J. E. Robinson III, M. Johnson, and J. Jung, “UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations,” in *16th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics, Jun. 2016, pp. 1–16.
[8] P. Fiorini and Z. Shiller, “Motion Planning in Dynamic Environments Using Velocity Obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, Jul. 1998.
[9] G. Chasparis and J. Shamma, “Linear-programming-based multi-vehicle path planning with adversaries,” in *American Control Conference*, Jun. 2005, pp. 1072–1077.
[10] J. van den Berg, Ming Lin, and D. Manocha, “Reciprocal Velocity Obstacles for real-time multi-agent navigation,” in *International Conference on Robotics and Automation*, May 2008, pp. 1928–1935.
[11] A. Wu and J. P. How, “Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles,” *Autonomous Robots*, vol. 32, no. 3, pp. 227–242, Apr. 2012.
[12] R. Olfati-Saber and R. M. Murray, “DISTRIBUTED COOPERATIVE CONTROL OF MULTIPLE VEHICLE FORMATIONS USING STRUCTURAL POTENTIAL FUNCTIONS,” *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 495–500, 2002.
[13] Y. L. Chuang, Y. R. Huang, M. R. D’Orsogna, and A. L. Bertozzi, “Multi-vehicle flocking: Scalability of cooperative control algorithms using pairwise potentials,” in *International Conference on Robotics and Automation*, Apr. 2007, pp. 2292–2299.
[14] Feng-Li Lian and R. Murray, “Real-time trajectory generation for the cooperative path planning of multi-vehicle systems,” in *Conference on Decision and Control*, vol. 4, Dec. 2002, pp. 3766–3769.
[15] A. Ahmadzadeh, N. Motee, A. Jadbabaie, and G. Pappas, “Multi-vehicle path planning in dynamically changing environments,” in *International Conference on Robotics and Automation*, May 2009, pp. 2449–2454.
[16] J. Bellingham, M. Tillerson, M. Alighanbari, and J. How, “Cooperative path planning for multiple UAVs in dynamic and uncertain environments,” in *Conference on Decision and Control*, vol. 3, Dec. 2002, pp. 2816–2822.
[17] R. Beard and T. McLain, “Multiple UAV cooperative search under collision avoidance and limited range communication constraints,” in *Conference on Decision and Control*, vol. 1, 2003, pp. 25–30.
[18] T. Schouwenaars and E. Feron, “Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees,” in *Guidance, Navigation, and Control Conference and Exhibit*, Aug. 2004, pp. 2004–5141.
[19] D. M. Stipanovic, P. F. Hokayem, M. W. Spong, and D. D. Siljak, “Cooperative Avoidance Control for Multiagent Systems,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, p. 699, 2007.
[20] M. Massink and N. De Francesco, “Modelling free flight with collision avoidance,” in *International Conference on Engineering of Complex Computer Systems*, 2001, pp. 270–279.
[21] M. Althoff and J. M. Dolan, “Set-based computation of vehicle behaviors for the online verification of autonomous vehicles,” in *International Conference on Intelligent Transportation Systems*, Oct. 2011, pp. 1162–1167.
[22] Y. Lin and S. Saripalli, “Collision avoidance for UAVs using reachable sets,” in *International Conference on Unmanned Aircraft Systems*, Jun. 2015, pp. 226–235.
[23] E. Lalish, K. A. Morgansen, and T. Tsukamaki, “Decentralized reactive collision avoidance for multiple unicycle-type vehicles,” in *American Control Conference*, Jun. 2008, pp. 5055–5061.
[24] G. M. Hoffmann and C. J. Tomlin, “Decentralized cooperative collision avoidance for acceleration constrained vehicles,” in *Conference on Decision and Control*, 2008, pp. 4357–4363.
[25] M. Chen, C.-Y. Shih, and C. J. Tomlin, “Multi-Vehicle Collision Avoidance via Hamilton-Jacobi Reachability and Mixed Integer Programming,” in *Conference on Decision and Control (to appear)*, 2016.
[26] E. N. Barron, “Differential games with maximum cost,” *Nonlinear Analysis*, vol. 14, no. 11, pp. 971–989, Jun. 1990.
[27] I. Mitchell, A. Bayen, and C. Tomlin, “A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games,” *Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.
[28] O. Bokanowski, N. Forcadel, and H. Zidani, “Reachability and Minimal Times for State Constrained Nonlinear Problems without Any Controllability Assumption,” *Journal on Control and Optimization*, vol. 48, no. 7, pp. 4292–4316, Jan. 2010.
[29] O. Bokanowski and H. Zidani, “MINIMAL TIME PROBLEMS WITH MOVING TARGETS AND OBSTACLES,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 2589–2593, Jan. 2011.
[30] K. Margellos and J. Lygeros, “HamiltonJacobi Formulation for ReachAvoid Differential Games,” *Transactions on Automatic Control*, vol. 56, no. 8, pp. 1849–1861, Aug. 2011.
[31] J. F. Fisac, M. , C. J. Tomlin, and S. S. Sastry, “Reach-avoid problems with time-varying dynamics, targets and constraints,” in *International Conference on Hybrid Systems Computation and Control*, 2015, pp. 11–20.
[32] J. A. Sethian, “A fast marching level set method for monotonically advancing fronts.” *National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996.
[33] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2006.
[34] I. M. Mitchell, “Application of Level Set Methods to Control and Reachability Problems in Continuous and Hybrid Systems,” Ph.D. dissertation, Stanford University, 2002.
[35] “A toolbox of level set methods,” Tech. Rep., 2007.
[36] A. M. Bayen, I. M. Mitchell, M. K. Osihi, and C. J. Tomlin, “Aircraft Autolander Safety Analysis Through Optimal Control-Based Reach Set Computation,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 68–77, Jan. 2007.
[37] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, “Reachability calculations for automated aerial refueling,” in *Conference on Decision and Control*, 2008, pp. 3706–3712.
[38] P. Bouffard, “On-board model predictive control of a quadrotor helicopter: Design, implementation, and experiments,” Master’s thesis, University of California, Berkeley, 2012.
[39] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, “A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag,” in *International Conference on Robotics and Automation*, 2011, pp. 1451–1456.
[40] M. Chen, J. F. Fisac, S. Sastry, and C. J. Tomlin, “Safe sequential path planning of multi-vehicle systems via double-obstacle Hamilton-Jacobi-Isaacs variational inequality,” in *European Control Conference*, Jul. 2015, pp. 3304–3309.
[41] S. Bansal, M. Chen, J. F. Fisac, and C. J. Tomlin, “Safe Sequential Path Planning of Multi-Vehicle Systems Under Presence of Disturbances and Imperfect Information,” in *American Control Conference (To appear)*, 2017.
[42] M. Chen, S. Bansal, J. F. Fisac, and C. J. Tomlin, “Robust sequential path planning under disturbances and adversarial intruder,” *arXiv preprint arXiv:1611.08364*, 2016.
[43] 3D Robotics, “Solo Specs: Just the facts,” 2015. [Online]. Available: https://news.3dr.com/solo-specs-just-the-facts-14480cb55722#.w7057q926
[44] New Atlas, “Amazon Prime Air.” [Online]. Available: http://newatlas.com/amazon-new-delivery-drones-us-faa-approval/36957/
[45] Wikipedia, “Beaufort scale.” [Online]. Available: https://en.wikipedia.org/wiki/Beaufort_scale#Modern_scale