Robust and Efficient Sequential Path Planning Under Adversarial Intruder

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

***Abstract*—Provably safe and scalable multi-vehicle path planning is an important and urgent problem due to the expected increase of automation in civilian airspace in the near future. Hamilton-Jacobi (HJ) reachability is an ideal tool for analyzing such safety-critical systems and has been successfully applied to several small-scale problems. However, a direct application of HJ reachability to large scale systems is often intractable because of its exponentially-scaling computation complexity with respect to system dimension, also known as the 'curse of dimensionality.' To overcome this problem, the sequential path planning (SPP) method, which assigns strict priorities to vehicles, was proposed; SPP allows multi-vehicle path planning to be done with a linearly-scaling computation complexity. 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 previous formulation requires the path planning problem to be re-solved in real time for guaranteeing goal satisfaction and safety, which is intractable for large-scale systems. In this paper, we make SPP more practical by providing a new algorithm where one 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 computational resources available during the run time. We demonstrate this algorithm in a representative simulation of the urban environment.**

I. Introduction

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]. These 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 endeavor 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

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, tomlin}@eecs.berkeley.edu

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 of vehicles. Authors in [41] extend this method to the scenarios where disturbances, such as wind, are present in the system, resolving some of the practical challenges associated with the basic SPP algorithm in [40]. However, if a vehicle not in the set of SPP vehicles enters the system, or even worse, if this vehicle has malicious intent, the original plan can lead to vehicles entering into another vehicles danger zone. Thus, 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, causing the entire SPP structure to collapse. Authors in [42], propose an SPP algorithm that accounts for such a potential intruder.

One of the limitations of the SPP algorithm proposed in [42] is that a full-scale new path planning problem is required to be solved in real time to ensure a safe transit of the SPP vehicles to their respective destinations. Since the replanning is done in real-time, the proposed algorithm is intractable for large-scale systems even with the SPP structure, rendering the method 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 if the intruder appears in the system, irrespective of the total number of SPP vehicles. Moreover, this number is a design parameter, which can be chosen beforehand based on the computation resources available during the run time, thus overcoming the limitations of the algorithm in [42]. Finally, we illustrate this algorithm through a fifty-vehicle simulation in an urban environment, demonstrating the scalability of the proposed algorithm.

Rest of the paper is organized as follows: in Section II, we formally present the SPP problem in the presence of disturbances and adversarial intruders. In Section III, we present a brief review of time-varying reachability, the basic SPP algorithms proposed in [40], [41]. 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. 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, if $(x_i, x_j) \in \mathcal{Z}_{ij}$.

In addition to the obstacles and danger zones, an intruder vehicle can also appear in the system, which can potentially have malicious intents. 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}. This assumption can be valid in situations where intruders are rare, and that some fail-safe or enforcement mechanism exists to force the intruder out of the altitude level affecting the SPP vehicles. 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.

Given the set of SPP vehicles, their targets $\mathcal{L}_i$, the static obstacles $\mathcal{O}_i^{\text{static}}$, the vehicles' danger zones with respect to each other $\mathcal{Z}_{ij}$, and the intruder dynamics $f_I(\cdot)$, 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 $\mathcal{O}_i^{\text{static}}$, the danger zones with respect to all other vehicles $\mathcal{Z}_{ij}, j \neq i$, and the intruder vehicle Q_I, irrespective of the control strategy of the intruder. 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 and the intruder vehicle. 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 $Q_j, j < i$ or the intruder vehicle Q_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 or Q_i and Q_I 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 $\mathcal{O}_i^{\text{static}}$, the danger zones of *higher-priority vehicles* $Q_j, j < i$, and the danger zone of the *intruder* Q_I irrespective of the intruder's control policy, 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.

However, when an intruder appears in the system, 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. Our goal in this work is to present an algorithm that ensures that only a fixed number of vehicles need to replan their trajectories, regardless of the total number of vehicles. Thus, the replanning time is constant and can be done in real time if this constant is small. In particular, we answer the following inter-dependent questions:

1) How can each vehicle guarantee that it will reach its target set without getting into any danger zones, despite no knowledge of the intruder initial state, the time at which it appears and its control strategy?
2) How can it be ensured that only a small number of vehicles need to replan their trajectories after the intruder disappears from the system?
3) Can we control the number the vehicles that need to replan their trajectories?

III. Background

In this section, we present the basic SPP algorithm [40] in which disturbances are ignored and perfect information of vehicles' positions is assumed. This simplification allows us to clearly present the basic SPP algorithm. We also briefly discuss the different algorithms proposed in [41] to account for disturbances in vehicles' dynamics. 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:

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

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

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

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

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

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. SPP Without Disturbances and Intruder

In this section, we give an overview of the basic SPP algorithm assuming that there is no disturbance and intruder affecting the vehicles. Although in practice, such assumptions do not hold, the description of the basic SPP algorithm will introduce the notation needed for describing the subsequent, more realistic versions of SPP. The majority of the content in this section is taken from [40].

Recall that the SPP vehicles $Q_i, i = 1,\ldots,N$, are each assigned a strict priority, with Q_j having a higher priority than Q_i if $j < i$. In the absence of disturbances, we can write the dynamics of the SPP vehicles as

$$\begin{aligned} &\dot{x}_i = f_i(x_i,u_i), t \leq t_i^{\text{STA}} \\ &u_i \in \mathcal{U}_i, \qquad i = 1\ldots,N \end{aligned} \tag{7}$$

In SPP, each vehicle Q_i plans the path to its target set $\mathcal{L}_i$ while avoiding static obstacles $\mathcal{O}_i^{\text{static}}$ and the obstacles $\mathcal{O}_i^j(t)$ induced by higher-priority vehicles $Q_j, j < i$. Path planning is done sequentially starting from the first vehicle and proceeding in descending priority, $Q_1, Q_2, \ldots, Q_N$ so that each of the path planning problems can be done in the state space of only one vehicle. During its path planning process, Q_i ignores the presence of lower-priority vehicles $Q_k, k > i$, and induces the obstacles $\mathcal{O}_k^i(t)$ for $Q_k, k > i$.

From the perspective of Q_i, each of the higher-priority vehicles $Q_j, j < i$ induces a time-varying obstacle denoted $\mathcal{O}_i^j(t)$ that Q_i needs to avoid. Therefore, each vehicle Q_i must plan its path to $\mathcal{L}_i$ while avoiding the union of all the induced obstacles as well as the static obstacles. Let $\mathcal{G}_i(t)$ be the union of all the obstacles that Q_i must avoid on its way to $\mathcal{L}_i$:

$$\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}} \cup \bigcup_{j=1}^{i-1} \mathcal{O}_i^j(t) \tag{8}$$

With full position information of higher priority vehicles, the obstacle induced for Q_i by Q_j is simply

$$\mathcal{O}_i^j(t) = \{x_i : \|p_i - p_j(t)\|_2 \leq R_c\} \tag{9}$$

Each higher priority vehicle Q_j plans its path while ignoring Q_i. Since path planning is done sequentially in descending order or priority, the vehicles $Q_j, j < i$ would have planned their paths before Q_i does. Thus, in the absence of disturbances, $p_j(t)$ is *a priori* known, and therefore $\mathcal{O}_i^j(t), j < i$ are known, deterministic moving obstacles, which means that $\mathcal{G}_i(t)$ is also known and deterministic. Therefore, the path planning problem for Q_i can be solved by first computing the BRS $\mathcal{V}_i^{\text{basic}}(t,t_i^{\text{STA}})$, defined as follows:

$$\begin{aligned} \mathcal{V}_i^{\text{basic}}(t,t_i^{\text{STA}}) = \{&y : \exists u_i(\cdot) \in \mathbb{U}_i, x_i(\cdot) \text{ satisfies (7)}, \\ &\forall s \in [t,t_i^{\text{STA}}], x_i(s) \notin \mathcal{G}_i(s), \\ &\exists s \in [t,t_i^{\text{STA}}], x_i(s) \in \mathcal{L}_i, x_i(t) = y\} \end{aligned} \tag{10}$$

The BRS $\mathcal{V}(t,t_i^{\text{STA}})$ can be obtained by solving (4) with $\mathcal{L} = \mathcal{L}_i$, $\mathcal{G}(t) = \mathcal{G}_i(t)$, and the Hamiltonian

$$H_i^{\text{basic}}(x_i,\lambda) = \min_{u_i \in \mathcal{U}_i} \lambda \cdot f_i(x_i,u_i) \tag{11}$$

The optimal control for reaching $\mathcal{L}_i$ while avoiding $\mathcal{G}_i(t)$ is then given by

$$u_i^{\text{basic}}(t,x_i) = \arg\min_{u_i \in \mathcal{U}_i} \lambda \cdot f_i(x_i,u_i) \tag{12}$$

from which the trajectory $x_i(\cdot)$ can be computed by integrating the system dynamics, which in this case are given by (7). In addition, the latest departure time t_i^{LDT} can be obtained from the BRS $\mathcal{V}(t,t_i^{\text{STA}})$ as $t_i^{\text{LDT}} = \arg\sup_t\{x_i^0 \in \mathcal{V}(t,t_i^{\text{STA}})\}$. In summary, the basic SPP algorithm is given as follows:

Algorithm 1: **Basic SPP algorithm**: Suppose we are given initial conditions x_i^0, vehicle dynamics (7), target sets $\mathcal{L}_i$, and static obstacles $\mathcal{O}_i^{\text{static}}, i = 1\ldots,N$. For each i in ascending order starting from $i = 1$ (which corresponds to descending order of priority),

1) determine the total obstacle set $\mathcal{G}_i(t)$, given in (8). In the case $i = 1$, $\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}}\ \forall t$;
2) compute the BRS $\mathcal{V}_i^{\text{basic}}(t,t_i^{\text{STA}})$ defined in (10). The latest departure time t_i^{LDT} is then given by $\arg\sup_t\{x_i^0 \in \mathcal{V}_i^{\text{basic}}(t,t_i^{\text{STA}})\}$;
3) determine the trajectory $x_i(\cdot)$ using vehicle dynamics (7), with the optimal control $u_i^{\text{basic}}(\cdot)$ given by (12);

4) given $x_i(\cdot)$, compute the induced obstacles $\mathcal{O}_k^i(t)$ for each $k > i$. In the absence of disturbances, $\mathcal{O}_k^i(t)$ is given by (9).

C. SPP With Disturbances and Without Intruder

Disturbances and incomplete information significantly complicate the SPP scheme. The main difference is that the vehicle dynamics satisfy (1) as opposed to (7). Committing to exact trajectories is therefore 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, unlike (9). In particular, a lower-priority vehicle needs to account for all possible states that the higher-priority vehicles could be in. To do this, the lower-priority vehicle needs to have some knowledge about the control policy used by each higher-priority vehicle. Three different methods are presented in [41] to address the above issues. The methods differ in terms of control policy information that is known to a lower-priority vehicle.

- **Centralized control**: A specific control strategy is enforced upon a vehicle; this can be achieved, for example, by some central agent such as an air traffic controller.
- **Least restrictive control**: A vehicle is required to arrive at its targets on time, but has no other restrictions on its control policy. When the control policy of a vehicle is unknown, but its timely arrive at its target can be assumed, the least restrictive control can be safely assumed by lower-priority vehicles.
- **Robust trajectory tracking**: A vehicle declares a nominal trajectory which can be robustly tracked under disturbances.

In each case, a vehicle Q_i can compute all possible states that a higher-priority vehicle Q_j can be in, based on the control strategy information known to the lower priority vehicle. During the path planning, Q_i uses this set of states as $\mathcal{O}_i^j(t)$. A collision avoidance between Q_i and Q_j is thus ensured. We refer to the obstacle $\mathcal{O}_i^j(t)$ as *base obstacle* here on. Note that here we only present the basic idea behind each algorithm for brevity. For further details, interested readers are referred to [41].

IV. Response to Intruders

In this section, we propose a method to allow vehicles to avoid an intruder while maintaining the SPP structure. 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.

In this work, a novel intruder avoidance algorithm is proposed, 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 at most $\bar{k}$ vehicles in a duration of t^{IAT} despite its best efforts. The proposed method builds 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, *at most* $\bar{k}$ vehicles have to replan their trajectories, 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, a space division of state-space is computed such that at most $\bar{k}$ vehicles need to apply the avoidance maneuver computed in Section IV-A, regardless of the initial state of the intruder. Trajectory planning and replanning are discussed in Sections IV-D and IV-E respectively. To keep the mathematical notations tractable, we summarize all introduced notations in Table I.

A. Optimal Avoidance Controller

In this section, our goal is to compute the control policy that a vehicle Q_i can use to avoid entering in the danger region $\mathcal{Z}_{iI}$. To compute the optimal avoidance control, 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:

$$\begin{aligned} x_{I,i} &= x_I - x_i \\ \dot{x}_{I,i} &= f_r(x_{I,i}, u_i, u_I, d_i, d_I) \end{aligned} \tag{13}$$

Given the relative dynamics, 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 is given by the backwards reachable set $\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}}),\ \tau \in [0, t^{\text{IAT}}]$:

$$\begin{aligned} \mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}}) =& \{y : \forall u_i(\cdot) \in \mathbb{U}_i, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & \exists d_I(\cdot) \in \mathbb{D}_I, x_{I,i}(\cdot) \text{ satisfies (13)}, \\ & \exists s \in [\tau, t^{\text{IAT}}], x_{I,i}(s) \in \mathcal{L}_i^{\text{A}}, x_{I,i}(\tau) = y\}, \\ \mathcal{L}_i^{\text{A}} =& \{x_{I,i} : \|p_{I,i}\|_2 \leq R_c\}. \end{aligned} \tag{14}$$

The Hamiltonian to compute $\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$ is given as:

$$H_i^{\text{A}}(x_{I,i}, \lambda) = \max_{u_i \in \mathcal{U}_i} \min_{\substack{u_I \in \mathcal{U}_I, \\ d_I \in \mathcal{D}_I, \\ d_i \in \mathcal{D}_i}} \lambda \cdot f_r(x_{I,i}, u_i, u_I, d_i, d_I). \tag{15}$$

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

$$u_i^{\text{A}} = \arg\max_{u_i \in \mathcal{U}_i} \min_{\substack{u_I \in \mathcal{U}_I, \\ d_I \in \mathcal{D}_I, \\ d_i \in \mathcal{D}_i}} \lambda \cdot f_r(x_{I,i}, u_i, u_I, d_i, d_I). \tag{16}$$

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 $\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}}), \tau < t^{\text{IAT}}$ is that if $x_{I,i}(t) \in \partial\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$, Q_i can successfully avoid the intruder for a duration of $(t^{\text{IAT}} - \tau)$ using the optimal avoidance control in (16). Moreover, if $x_{I,i}(t) \in \left(\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})\right)^C$, then Q_i and Q_I instantaneously cannot enter the danger zone $\mathcal{Z}_{iI}$, irrespective of the control applied by them at time t.

Notation	Name	Interpretation
$\mathcal{V}_i^{\mathrm{A}}(\cdot, t^{\mathrm{IAT}})$	Avoid region of Q_i	Set of relative states $x_{I,i}$ for which Q_i is forced to apply an avoidance maneuver.
$\underline{t}_i$	Avoid start time of Q_i	The first time at which Q_i is forced to apply an avoidance maneuver by Q_I.
$\mathcal{M}_j(t)$	Base obstacle induced by Q_j at time t	The set of all possible states that Q_j can be in at time t if the intruder doesn't appear in the system till time t.
$\mathcal{B}_{ij}(t)$	Buffer region between Q_j and Q_i	The set of all possible states for which the separation requirement is violated between Q_j and Q_i for some intruder strategy.
d^{A}	Detection range	Minimum detection range required to successfully avoid the intruder.
$\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$	Relative buffer region	The set of all states from which it is possible to reach the boundary of the avoid region of Q_i in a duration of t^{BRD}.
$\mathcal{S}_j(t)$	Separation region of Q_j at time t	Set of all initial states of intruder for which Q_j is forced to apply an avoidance maneuver.

TABLE I: Different mathematical notations and their interpretation.

In the worst case, Q_i might need to avoid the intruder for a duration of t^{IAT}; thus, we must have that $x_{I,i}(\underline{t}) \in \left(\mathcal{V}_i^{\mathrm{A}}(0, t^{\mathrm{IAT}})\right)^C$ to ensure successful avoidance irrespective of the intruder control strategy. Equivalently, every SPP vehicle must be able to detect the intruder at a distance of d^{A}, where

$$d^{\mathrm{A}} = \max\{\|p_i\| : \exists h_i, (p_i, h_i) \in \mathcal{V}_i^{\mathrm{A}}(0, t^{\mathrm{IAT}})\}; \tag{17}$$

otherwise, there exist an intruder control strategy such that Q_i and Q_I will collide irrespective of the control used by Q_i. Thus, d^{A} is the *minimum* detection range required by any path-planning algorithm to ensure a successful intruder avoidance for all intruder strategies. Hence, for analysis to follow, we assume that every SPP vehicle can detect the intruder at a distance of d^{A}. Interestingly, this detection range is also sufficient to avoid an intruder. In fact, the proposed algorithm does not require detection beyond 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 at most $\bar{k}$ vehicles are *forced* to apply avoidance maneuver during the duration $[\underline{t}, \bar{t}]$.

To capture this mathematically, we define

$$\mathcal{A}_m := \{t : x_{I,m}(t) \in \partial\mathcal{V}_m^{\mathrm{A}}(t - \underline{t}, t^{\mathrm{IAT}}), t \in [\underline{t}, \bar{t}]\} \tag{18}$$

$\mathcal{A}_m$ 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:

$$\underline{t}_m = \begin{cases} \min_{t \in \mathcal{A}_m} t & \text{if } \mathcal{A}_m \neq \emptyset \\ \infty & \text{otherwise} \end{cases} \tag{19}$$

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. From (18) it follows that at the avoid start time, $x_{I,m}(\underline{t}_m) \in \partial\mathcal{V}_m^{\mathrm{A}}(\underline{t}_m - \underline{t}, t^{\mathrm{IAT}})$.

If we ensure that

$$\forall i \neq j, min(\underline{t}_i, \underline{t}_j) < \infty \implies |\underline{t}_i - \underline{t}_j| \geq \frac{t^{\mathrm{IAT}}}{\bar{k}} := t^{\mathrm{BRD}}, \tag{20}$$

then at most $\bar{k}$ vehicles can be forced to apply the avoidance maneuver during the time interval $[\underline{t}, \bar{t}]$. We refer to the condition in (20) as *separation requirement* here on.

For any given time t, if we could find the set of all states of Q_i such that the separation requirement is violated for Q_i and Q_j for some $j < i$ and for some intruder strategy, then during the path planning of Q_i, it can be ensured that Q_i is not in one of these states at time t by using this set of states as "obstacles". The sequential path planning will therefore guarantee that the separation requirement holds for every SPP vehicle pair. Thus, we can focus on finding all states $x_i(t)$ such that the separation requirements is violated between vehicles Q_j and Q_i, $j < i$ at time t for some possible intruder scenario (meaning some possible $\underline{t}, \bar{t}, x_I^0$ and $u_I(\cdot)$). For this analysis, it is sufficient to consider the following two mutually exclusive and exhaustive cases:

1) Case1: $\underline{t}_j \leq \underline{t}_i, \underline{t}_j < \infty$
2) Case2: $\underline{t}_i < \underline{t}_j, \underline{t}_i < \infty$

In this section, we consider Case1. 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, i.e., $x_{I,j}(\underline{t}) \in \partial\mathcal{V}_j^{\mathrm{A}}(0, t^{\mathrm{IAT}})$. 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 in this case before it reaches the boundary of the avoid region of Q_i. Equivalently, we can assume that $\underline{t}_j = \underline{t}$. Thus, we will use $\underline{t}_j$ and $\underline{t}$ interchangeably in Case1.

1) Separation region: Consider any $\underline{t} \in \mathbb{R}$. In this section, our goal is to find the set of all states $x_I^0 := x_I(\underline{t})$ for which Q_j is forced to apply an avoidance 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 avoidance maneuver at time $\underline{t}$ only if $x_{I,j}(\underline{t}) \in \partial\mathcal{V}_j^{\mathrm{A}}(0, t^{\mathrm{IAT}})$. To compute set $\mathcal{S}_j(\underline{t})$, we thus need to translate this set in relative coordinates to a set in absolute coordinates. Therefore, if all possible states of Q_j at time $\underline{t}$ are known, then $\mathcal{S}_j(\underline{t})$ can be computed trivially.

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 $\mathcal{M}_j(\underline{t})$. Recall from the section III-C that base obstacle at time t represent all the states that Q_j can be in at time t in the presence of disturbances, but in the absence of an intruder. In particular, if the intruder doesn't appear in the system till time t, then $\mathcal{M}_j(t)$ captures all possible states of Q_j at time t. 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:

$$\mathcal{S}_j(\underline{t}) = \mathcal{M}_j(\underline{t}) + \partial\mathcal{V}_j^{\mathrm{A}}(0, t^{\mathrm{IAT}}),\ \underline{t} \in \mathbb{R}, \tag{21}$$

where the "+" in (21) 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 avoidance maneuver, we must have $x_I(\underline{t}) \in \mathcal{S}_j(\underline{t})$ in Case1.

2) Buffer Region: Recall that from the definition of avoid start time, it follows that $x_{I,i}(\underline{t}_i) \in \partial\mathcal{V}_i^{\mathrm{A}}(\underline{t}_i - \underline{t}, t^{\mathrm{IAT}})$. In this section, we first compute the set of all relative states, $\mathcal{V}_i^{\mathrm{B}}$, from which it is possible to reach $\partial\mathcal{V}_i^{\mathrm{A}}(\underline{t}_i - \underline{t}_j, t^{\mathrm{IAT}})$ in a duration of t^{BRD} for some control applied by Q_I and Q_i, where t^{BRD} is defined in (20). Further, if $\mathcal{V}_i^{\mathrm{B}}$ is augmented on $\mathcal{S}_j(\underline{t}_j)$ (to get $\mathcal{B}_{ij}(\underline{t}_j)$), then we can guarantee that Q_I cannot reach the boundary (or inside) of $\mathcal{V}_i^{\mathrm{A}}(t - \underline{t}_j, t^{\mathrm{IAT}})$ before $t = \underline{t}_j + t^{\mathrm{BRD}}$, irrespective of the control applied by Q_I and Q_i during $[\underline{t}_j, t]$, if $x_I(\underline{t}_j) \in \mathcal{S}_j(\underline{t}_j)$ and $x_i(\underline{t}_j) \in \left(\mathcal{B}_{ij}(\underline{t}_j)\right)^C$. Equivalently, $\underline{t}_i \geq \underline{t}_j + t^{\mathrm{BRD}}$, if $x_i(\underline{t}_j) \in \left(\mathcal{B}_{ij}(\underline{t}_j)\right)^C$. Note that $x_I(\underline{t}_j) \in \mathcal{S}_j(\underline{t}_j)$ is already ensured by the construction of $\mathcal{S}_j(\underline{t}_j)$.

We refer to the set $\mathcal{B}_{ij}(t)$ as *buffer region* hereon. Thus, if $x_i(t) \notin \mathcal{B}_{ij}(t)$, then the separation requirement holds in Case1. To compute $\mathcal{B}_{ij}(\underline{t})$, we first compute $\mathcal{V}_i^{\mathrm{B}}$ which is given by the following BRS:

$$\begin{aligned}
\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_i(\cdot) \in \mathbb{D}_i, \\
& \exists d_I(\cdot) \in \mathbb{D}_I, x_{i,I}(\cdot) \text{ satisfies (13)}, \\
& \exists s \in [0, t^{\mathrm{BRD}}], x_{i,I}(s) \in -\mathcal{V}_i^{\mathrm{A}}(t^{\mathrm{BRD}}, t^{\mathrm{IAT}}), \\
& x_{i,I}(t) = y\}, \\
-\mathcal{V}_i^{\mathrm{A}}(t^{\mathrm{BRD}}, t^{\mathrm{IAT}}) =& \{y : -y \in \mathcal{V}_i^{\mathrm{A}}(t^{\mathrm{BRD}}, t^{\mathrm{IAT}})\}.
\end{aligned} \tag{22}$$

The Hamiltonian to compute $\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$ is given by:

$$H_i^{\mathrm{B}}(x_{i,I}, \lambda) = \min_{\substack{u_i \in \mathcal{U}_i, u_I \in \mathcal{U}_I, \\ d_i \in \mathcal{D}_i, d_I \in \mathcal{D}_I}} \lambda \cdot f_r(x_{i,I}, u_i, u_I, d_i, d_I). \tag{23}$$

Intuitively, $\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$ represents the set of all relative states $x_{i,I}$ from which it is possible to reach the boundary of $\mathcal{V}_i^{\mathrm{A}}(t^{\mathrm{BRD}}, t^{\mathrm{IAT}})$ within a duration of t^{BRD}. Note that we use $-\mathcal{V}_i^{\mathrm{A}}(0, t^{\mathrm{IAT}})$ instead of $\mathcal{V}_i^{\mathrm{A}}(0, t^{\mathrm{IAT}})$ as the target set for our computation above because we are computing BRS $\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$ using the relative state $x_{i,I}$ and not $x_{I,i}$. We refer to set $\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$ as *relative buffer region* here on.

The buffer region can now be computed by augmenting $\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$ on the separation region of $\mathcal{S}_j(\underline{t})$:

$$\mathcal{B}_{ij}(\underline{t}) = \mathcal{S}_j(\underline{t}) + \mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}}). \tag{24}$$

Thus, we can ensure that $(\underline{t}_i - \underline{t}_j) \geq t^{\mathrm{BRD}}$ as long as $x_i(\underline{t}) \in (\mathcal{B}_{ij}(\underline{t}))^C$. Therefore, Q_I can force at most $\bar{k}$ vehicles to apply an avoidance maneuver during a duration of t^{IAT}.

3) Obstacle Computation: In sections IV-B1 and IV-B2, we computed a separation between Q_i and Q_j such that $\underline{t}_i - \underline{t}_j \geq t^{\mathrm{BRD}}$. However, it needs to be ensured that while applying an avoidance maneuver Q_i and Q_j do not enter in each other's danger zone 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 accidentally entering in $\mathcal{Z}_{ij}$ while avoiding the intruder. Since the path planning is done in a sequential fashion, Q_i being a lower priority vehicle also needs to avoid the states that can lead it to $\mathcal{Z}_{ij}$ while Q_j is avoiding the intruder. To find such states, we consider the following two exhaustive cases:

1) CaseA: intruder affects Q_j, but not Q_i, i.e., $\underline{t}_j < \infty$ and $\underline{t}_i = \infty$.
2) CaseB: intruder first affects Q_j and then Q_i, i.e., $\underline{t}_j, \underline{t}_i < \infty$.

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 ${}_1^A\mathcal{O}_i^j(\cdot)$ and ${}_1^B\mathcal{O}_i^j(\cdot)$ denote the corresponding sets of "obstacles" for the two cases. 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: In this case, only Q_j applies an avoidance maneuver. Therefore, Q_i should avoid the set of states, ${}_1^A\mathcal{O}_i^j(t)$, that can lead to a collision with Q_j at time t while Q_j is applying an avoidance maneuver. Note that since $\underline{t}_j = \underline{t}$ (by Observation 1), ${}_1^A\mathcal{O}_i^j(t)$ 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.

$$\begin{aligned}
\mathcal{W}_j^{\mathcal{O}}(\underline{t}, t) =& \{y : \exists u_j(\cdot) \in \mathbb{U}_j, \exists d_j(\cdot) \in \mathbb{D}_j, \\
& x_j(\cdot) \text{ satisfies (1)}, x_j(\underline{t}) \in \mathcal{M}_j(\underline{t}), \\
& x_j(t) = y\}.
\end{aligned} \tag{25}$$

$\mathcal{W}_j^{\mathcal{O}}(\underline{t}, t)$ 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:

$$H_j^{\mathcal{O}}(x_j, \lambda) = \max_{u_j \in \mathcal{U}_j} \max_{d_j \in \mathcal{D}_j} \lambda \cdot f_j(x_j, u_j, d_j). \tag{26}$$

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

$$\begin{aligned}
{}_1^A\mathcal{O}_i^j(t) &= \{x_i : \exists y \in \mathcal{P}_j(t), \|p_i - y\|_2 \leq R_c\} \\
\mathcal{P}_j(t) &= \{p_j : \exists h_j, (p_j, h_j) \in \bigcup_{\underline{t} \in [t - t^{\mathrm{IAT}}, t]} \mathcal{W}_j^{\mathcal{O}}(\underline{t}, t)\}
\end{aligned} \tag{27}$$

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 to the FRSs computed from two different base obstacles $\mathcal{M}_j(\tau_2)$ and $\mathcal{M}_j(\tau_1)$, i.e., $\mathcal{W}_j^{\mathcal{O}}(\tau_2, \tau_3) \subseteq \mathcal{W}_j^{\mathcal{O}}(\tau_1, \tau_3)$ if $\tau_1 < \tau_2 < \tau_3$.

Using observation 3, $\mathcal{P}_j(t)$ in (27) can be equivalently written as:

$$\mathcal{P}_j(t) = \{p_j : \exists h_j, (p_j, h_j) \in \mathcal{W}_j^{\mathcal{O}}(t - t^{\text{IAT}}, t)\}. \quad (28)$$

- CaseB: In this case, first Q_j applies an avoidance maneuver followed by Q_i. Therefore, $\underline{t}_j, \underline{t}_i \in [\underline{t}, \bar{t}]$. Once Q_j starts applying avoidance control at time $\underline{t} = \underline{t}_j$, it might deviate from its pre-planned control strategy. 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 avoidance 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 into $\mathcal{Z}_{ji}$ even when both are applying avoidance maneuver and hence can apply *any* control from each other's perspective. Thus at time t, Q_i not only need to 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 ${}_1^B\mathcal{O}_i^j(t)$, 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 ${}_1^A\mathcal{O}_i^j(t)$. 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 that it would have if $\underline{t}_i = t$.

  Due to the separation and buffer regions, we have $\underline{t}_i - \underline{t}_j \geq t^{\text{BRD}}$. This along with Observation 4 thus implies that $\underline{t}_j \leq t - t^{\text{BRD}}$. Also, from Observation 2, we have $\underline{t} = \underline{t}_j \geq t - t^{\text{IAT}}$. Thus, $\underline{t}_j \in [t - t^{\text{IAT}}, t - t^{\text{BRD}}]$. Since the intruder is present for a maximum duration of t^{IAT}, Q_j might be applying any control during $[\underline{t}_j, \underline{t}_j + t^{\text{IAT}}]$ from the perspective of Q_i. In particular, for any given $\underline{t}_j$, Q_j can reach any state in $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, t')$ at time $t' \in [\underline{t}_j, \underline{t}_j + t^{\text{IAT}}]$, 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 forward for a duration of $(t' - \underline{t}_j)$ and is given by (25).

  Taking into account all possible $\underline{t}_j \in [t - t^{\text{IAT}}, t - t^{\text{BRD}}]$, $x_j(\tau)$ is contained in the set:

$$\mathcal{K}^{\text{B1}}(\tau) = \bigcup_{\underline{t}_j \in [\tau - t^{\text{IAT}}, t - t^{\text{BRD}}]} \mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau) \quad (29)$$

  at time $\tau \in [t, t - t^{\text{BRD}} + t^{\text{IAT}}]$, where the upper bound on τ corresponds to the upper bound on $\underline{t}_j$. From Observation 3, we have $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau) \subseteq \mathcal{W}_j^{\mathcal{O}}(\tau - t^{\text{IAT}}, \tau)$ for all $\underline{t}_j \in [\tau - t^{\text{IAT}}, t - t^{\text{BRD}}]$. Therefore, $\mathcal{K}^{\text{B1}}(\tau) = \mathcal{W}_j^{\mathcal{O}}(\tau - t^{\text{IAT}}, \tau)$.

  From the perspective of Q_i, it needs to avoid all states at time t that can reach $\mathcal{K}^{\text{B1}}(\tau)$ 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:

$$\begin{aligned}\mathcal{V}_i^{\text{B1}}(t, t - t^{\text{BRD}} + t^{\text{IAT}}) = \{&y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i, \\ &x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y, \\ &\exists s \in [t, t - t^{\text{BRD}} + t^{\text{IAT}}], \\ &x_i(s) \in \tilde{\mathcal{K}}^{\text{B1}}(s)\},\end{aligned} \quad (30)$$

  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:

$$H_i^{\text{B1}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i, d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \quad (31)$$

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

$${}_1^B\mathcal{O}_i^j(t) = \mathcal{V}_i^{\text{B1}}(t, t - t^{\text{BRD}} + t^{\text{IAT}}). \quad (32)$$

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. The analysis in this case is fairly similar to Case1. There are a few differences however, which we point out wherever relevant. We start with an observation similar to Observation 1:

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 avoidance maneuver. Since, Q_j applies the avoidance maneuver after Q_i in this case, Q_j will need to avoid the intruder for a maximum duration of $t^{\text{RD}} := t^{\text{IAT}} - t^{\text{BRD}}$. This is due to the fact that our design of the buffer region in Section IV-C2 ensures that it takes Q_I at least a duration of t^{BRD} to go from the boundary of the avoid region of Q_i to that of Q_j. $\mathcal{S}_j(\underline{t}_j)$ can thus be obtained as:

$$\mathcal{S}_j(\underline{t}_j) = \mathcal{M}_j(\underline{t}_j) + \partial\mathcal{V}_j^{\text{A}}(0, t^{\text{RD}}). \quad (33)$$

2) Buffer Region: The idea behind the design of buffer region is same as that in Case1: we want to make sure that Q_I spends at least a duration of t^{BRD} to go from the boundary of the avoid region of one SPP vehicle to the boundary of the avoid region of some other SPP vehicle. Mathematically, we 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 $\mathcal{V}_j^{\text{B}}(0, t^{\text{BRD}})$:

$$\begin{aligned}\mathcal{V}_j^{\text{B}}(0, t^{\text{BRD}}) = \{&y : \exists u_j(\cdot) \in \mathbb{U}_j, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_j(\cdot) \in \mathbb{D}_j, \\ &\exists d_I(\cdot) \in \mathbb{D}_I, x_{I,j}(\cdot) \text{ satisfies (13)}, \\ &\exists s \in [0, t^{\text{BRD}}], x_{I,j}(s) \in \mathcal{V}_j^{\text{A}}(t^{\text{BRD}}, t^{\text{IAT}}), \\ &x_{I,j}(t) = y\},\end{aligned} \quad (34)$$

where

$$H_j^{\text{B}}(x_{I,j}, \lambda) = \min_{\substack{u_j \in \mathcal{U}_j, u_I \in \mathcal{U}_I, \\ d_j \in \mathcal{D}_j, d_I \in \mathcal{D}_I}} \lambda \cdot f_r(x_{I,j}, u_j, u_I, d_j, d_I) \quad (35)$$

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 cannot 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 $x_{I,i}(\underline{t}) := x_{I,i}(\underline{t}_i) \in \partial\mathcal{V}_i^{\mathrm{A}}(0, t^{\mathrm{IAT}})$ implies that $\underline{t}_j - \underline{t}_i \geq t^{\mathrm{BRD}}$. Mathematically, if we define the set,

$$\mathcal{B}_{ji}(\underline{t}) = \mathcal{M}_j(\underline{t}) + \mathcal{V}_j^{\mathrm{B}}(0, t^{\mathrm{BRD}}) + \left(-\mathcal{V}_i^{\mathrm{A}}(0, t^{\mathrm{IAT}})\right), \quad (36)$$

then $(\underline{t}_j - \underline{t}_i) \geq t^{\mathrm{BRD}}$ as long as $x_i(\underline{t}) \in (\mathcal{B}_{ji}(\underline{t}))^C$. Thus, if $x_i(\underline{t}) \in (\mathcal{B}_{ji}(\underline{t}))^C$, then the separation requirement is satisfied in Case2. Further, if $x_i(\underline{t}) \in (\mathcal{B}_{ji}(\underline{t}) \cup \mathcal{B}_{ij}(\underline{t}))^C$, then the separation requirement is satisfied regardless of *any* intruder strategy.

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 ${}^{A}_{2}\mathcal{O}_i^j(\cdot)$ and ${}^{B}_{2}\mathcal{O}_i^j(\cdot)$ 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 Q_j while it is avoiding the intruder. Since Q_j is not avoiding the intruder in this case, the set of possible states of Q_j at time t is given by $\mathcal{M}_j(t)$. To compute ${}^{A}_{2}\mathcal{O}_i^j(\cdot)$, we begin with the following observation:

  Observation 6: By Observation 2, it is sufficient to consider the scenarios where $\underline{t} = \underline{t}_i \in [t - t^{\mathrm{IAT}}, t]$. Since Q_i can be forced to apply an avoidance maneuver for the time interval $[\underline{t}_i, \underline{t}_i + t^{\mathrm{IAT}}]$, 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 $[t, \underline{t}_i + t^{\mathrm{IAT}}]$ for some avoidance control. Therefore, it is sufficient to consider the scenario $\underline{t}_i = t$ as that will maximize the avoidance duration $[\underline{t}_i, \underline{t}_i + t^{\mathrm{IAT}}]$ for the obstacle computation at time t.

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

$$\begin{aligned}\mathcal{K}^{\mathrm{A2}}(\tau) =& \tilde{\mathcal{M}}_j(\tau),\\ \tilde{\mathcal{M}}_j(s) =& \{x_j : \exists (y, h) \in \mathcal{M}_j(s), \|p_j - y\|_2 \leq R_c\}.\end{aligned} \quad (37)$$

  $\tilde{\mathcal{M}}_j(s)$ represent the set of all states that are in potential collision with Q_j at time s. 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 $\tau \in [t, t + t^{\mathrm{IAT}}]$. Avoiding $\mathcal{K}^{\mathrm{A2}}(\cdot)$ will ensure that Q_i and Q_j will not enter into each other's danger zones regardless of the avoidance maneuver applied by Q_i. The set of states that Q_i needs to avoid at time t is given by the following BRS:

$$\begin{aligned}\mathcal{V}_i^{\mathrm{A2}}(t, t + t^{\mathrm{IAT}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i,\\ & x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y,\\ & \exists s \in [t, t + t^{\mathrm{IAT}}], x_i(s) \in \mathcal{K}^{\mathrm{A2}}(s)\}.\end{aligned} \quad (38)$$

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

$$H_i^{\mathrm{A2}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i, d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \quad (39)$$

  $\mathcal{V}_i^{\mathrm{A2}}(t, t + t^{\mathrm{IAT}})$ represents the set of all states of Q_i at time t from which it is possible for Q_i to reach $\mathcal{K}^{\mathrm{A2}}(\tau)$ for some $\tau \geq t$. Thus, the induced obstacle in this case is given as

$${}^{A}_{2}\mathcal{O}_i^j(t) = \mathcal{V}_i^{\mathrm{A2}}(t, t + t^{\mathrm{IAT}}). \quad (40)$$

- CaseB: In this case, the intruder first affects Q_i and then Q_j. Q_i and Q_j apply their first avoidance maneuver at $\underline{t}_i$ and $\underline{t}_j$ respectively. Since the intruder appears for a maximum duration of t^{IAT} and $\underline{t}_i = \underline{t}$, $\bar{t} \leq \underline{t}_i + t^{\mathrm{IAT}}$. Thus, from the perspective of Q_i, Q_j can apply any control during the duration $[\underline{t}_j, \underline{t}_i + t^{\mathrm{IAT}}]$ and hence can be anywhere in the set $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau)$ at $\tau \in [\underline{t}_j, \underline{t}_i + t^{\mathrm{IAT}}]$, where $\mathcal{W}_j^{\mathcal{O}}$ denotes the FRS of base obstacle $\mathcal{M}_j(\underline{t}_j)$ computed forward for a duration of $(\underline{t}_i + t^{\mathrm{IAT}} - \underline{t}_j)$. 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, $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau)$, is induced by Q_j at τ if $\underline{t}_j$ is as early as possible. Hence, it is sufficient for Q_i to avoid this obstacle to ensure collision avoidance with Q_j at time τ. Given the separation and buffer regions between Q_i and Q_j, we must have $\underline{t}_j - \underline{t}_i \geq t^{\mathrm{BRD}}$. Hence, the biggest obstacle is induced by Q_j when $\underline{t}_j = \underline{t}_i + t^{\mathrm{BRD}}$.

  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 force Q_j to apply an avoidance maneuver after a duration of t^{BRD}. Therefore, Q_i needs to avoid $\mathcal{K}^{\mathrm{B2}}(\tau)$ at time $\tau > t$, where

$$\mathcal{K}^{\mathrm{B2}}(\tau) = \bigcup_{\underline{t}_i \in [t - t^{\mathrm{IAT}}, t], \tau - \underline{t}_i \leq t^{\mathrm{IAT}}} \mathcal{W}_j^{\mathcal{O}}(\underline{t}_i + t^{\mathrm{BRD}}, \tau), \tau > t, \quad (41)$$

  where we have substituted $\underline{t}_j = \underline{t}_i + t^{\mathrm{BRD}}$. In (41), $\underline{t} = \underline{t}_i \in [t - t^{\mathrm{IAT}}, t]$ due to Observation 2 and $\tau - \underline{t}_i \leq t^{\mathrm{IAT}}$ because the intruder can appear for a maximum duration of t^{IAT}. (41) can be equivalently written as:

$$\begin{aligned}\mathcal{K}^{\mathrm{B2}}(\tau) =& \bigcup_{\underline{t}_i \in [\tau - t^{\mathrm{IAT}}, t]} \mathcal{W}_j^{\mathcal{O}}(\underline{t}_i + t^{\mathrm{BRD}}, \tau), t < \tau \leq t + t^{\mathrm{IAT}}\\ \mathcal{K}^{\mathrm{B2}}(\tau) =& \mathcal{W}_j^{\mathcal{O}}(\tau - t^{\mathrm{IAT}} + t^{\mathrm{BRD}}, \tau), t < \tau \leq t + t^{\mathrm{IAT}},\end{aligned} \quad (42)$$

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

$$\begin{aligned}\mathcal{V}_i^{\mathrm{B2}}(t, t + t^{\mathrm{IAT}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i,\\ & x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y,\\ & \exists s \in [t + t^{\mathrm{BRD}}, t + t^{\mathrm{IAT}}], x_i(s) \in \tilde{\mathcal{K}}^{\mathrm{B2}}(s)\},\\ \tilde{\mathcal{K}}^{\mathrm{B2}}(s) =& \{x_i : \exists (y, h) \in \mathcal{K}^{\mathrm{B2}}(s), \|p_i - y\|_2 \leq R_c\}.\end{aligned} \quad (43)$$

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

$$H_i^{\mathrm{B2}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i, d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \quad (44)$$

Finally, the induced obstacle in this case is given as

$$ {}^{B}_{2}\mathcal{O}^{j}_{i}(t) = \mathcal{V}^{\text{B2}}_{i}(t, t + t^{\text{IAT}}). \tag{45} $$

D. Path Planning

In sections IV-B3 and IV-C3, we computed obstacles for Q_i such that Q_i and Q_j do not collide with each other while avoiding intruder. We next compute the states that Q_i needs to avoid to avoid a collision with static obstacles while it is applying an avoidance maneuver. Since Q_i applies avoidance maneuver for a maximum duration of t^{IAT} this set is given by the following BRS:

$$ \begin{aligned} \mathcal{V}^{\text{S}}_{i}(t, t + t^{\text{IAT}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y, \\ & \exists s \in [t, t + t^{\text{IAT}}], x_i(s) \in \mathcal{K}^{\text{S}}(s)\}, \\ \mathcal{K}^{\text{S}}(s) =& \{x_i : \exists (y, h) \in \mathcal{O}^{\text{static}}_{i}, \|p_i - y\|_2 \leq R_c\}. \end{aligned} \tag{46} $$

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

$$ H^{\text{S}}_{i}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i, d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \tag{47} $$

$\mathcal{V}^{\text{S}}_{i}(t, t + t^{\text{IAT}})$ represents the set of all states of Q_i at time t that can lead to a collision with a static obstacle for some $\tau > t$ for some control strategy of Q_i.

During the path planning of Q_i, if we use $\mathcal{B}_{ij}(t)$ and $\mathcal{B}_{ji}(t)$ as obstacles at time t, then the separation requirement is ensured between Q_i and Q_j for all intruder strategies and $\underline{t} = t$. Similarly, if obstacles computed in sections IV-B3 and IV-C3 as obstacles in path planning, then we can guarantee collision avoidance between Q_i and Q_j while they are avoiding the intruder. The overall obstacle for Q_i is thus given by:

$$ \begin{aligned} \mathcal{G}_i(t) =& \mathcal{V}^{\text{S}}_{i}(t, t + t^{\text{IAT}}) \bigcup \\ & \bigcup_{j=1}^{i-1} \left(\mathcal{B}_{ij}(t) \cup \mathcal{B}_{ji}(t) \bigcup_{k \in \{1,2\}} {}^{A}_{k}\mathcal{O}^{j}_{i}(t) \bigcup_{k \in \{1,2\}} {}^{B}_{k}\mathcal{O}^{j}_{i}(t) \right). \end{aligned} \tag{48} $$

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

$$ \begin{aligned} \mathcal{V}^{\text{PP}}_{i}(t, t^{\text{STA}}_{i}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \forall d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, \forall s \in [t, t^{\text{STA}}_{i}], x_i(s) \notin \mathcal{G}_i(s), \\ & \exists s \in [t, t^{\text{STA}}_{i}], x_i(s) \in \mathcal{L}_i, x_i(t) = y\}. \end{aligned} \tag{49} $$

The Hamiltonian H^{PP}_{i} to compute BRS in (49) is given by:

$$ H^{\text{PP}}_{i}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i} \max_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \tag{50} $$

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

$$ u^{\text{PP}}_{i}(t, x_i) = \arg \min_{u_i \in \mathcal{U}_i} \max_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \tag{51} $$

When intruder is not present in the system, Q_i applies the control u^{PP}_{i} and we get the 'nominal trajectory' of Q_i. Once intruder appears in the system, Q_i applies the avoidance control u^{A} and hence might deviate from its nominal trajectory. The overall control policy for avoiding the intruder and collision with other vehicles is thus given by:

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

Note that if Q_i starts within $\mathcal{V}^{\text{PP}}_{i}$ and use avoidance control strategy in (16), 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 $j < i$.

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 (48) by the error bound of Q_i, Ω_i (for details, see Section 4C in [41]). Also, the BRS in (49) is computed assuming no disturbance in Q_i's dynamics.

E. Replanning after intruder avoidance

As discussed in Section IV-D, the intruder can force some SPP vehicles to deviate from their planned nominal trajectory; therefore, goal satisfaction is no longer guaranteed once a vehicle is forced to apply an avoidance maneuver. Therefore, we have to replan the trajectories 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 avoidance control during $[\underline{t}, \bar{t}]$, e.g.,

$$ \mathcal{N}^{\text{RP}} = \{Q_i : \underline{t}_i < \infty, i \in \{1, \ldots, N\}\}. \tag{52} $$

Note that due to the presence of separation and buffer regions, at most $\bar{k}$ 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 $\tilde{x}^{0}_{i}$, after avoiding the intruder. Note that we can pick $\bar{k}$ beforehand and design buffer regions accordingly. Thus, by picking compatible $\bar{k}$ based on the available computation resources during run-time, we can ensure that this replanning can be done in real time.

Let the optimal control policy corresponding to this liveness controller be denoted $u^{\text{L}}_{i}(t, x_i)$. 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:

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

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

$$ \begin{aligned} \mathcal{W}^{\text{RP}}_{i}(\bar{t}, t) =& \{y \in \mathbb{R}^{n_i} : \exists u_i(\cdot) \in \mathbb{U}_i, \forall d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, x_i(\bar{t}) = \tilde{x}^{0}_{i}, \\ & x_i(t) = y, \forall s \in [\bar{t}, t], x_i(s) \notin \mathcal{G}^{\text{RP}}_{i}(s)\}, \end{aligned} \tag{53} $$

where $\tilde{x}^{0}_{i}$ represents the state of Q_i at $t = \bar{t}$; $\mathcal{G}^{\text{RP}}_{i}(\cdot)$ takes into account the fact that Q_i now needs to avoid all other vehicles

in $(\mathcal{N}^{\text{RP}})^C$ and is defined in a way analogous to (8). The FRS in (53) can be obtained by solving

$$\begin{aligned}\max\Big\{&D_t W_i^{\text{RP}}(t,x_i)+H_i^{\text{RP}}(t,x_i,\nabla W_i^{\text{RP}}(t,x_i)),\\ &-g_i^{\text{RP}}(t,x_i)-W_i^{\text{RP}}(t,x_i)\Big\}=0\\ W_i^{\text{RP}}(\underline{t},x_i)&=\max\{l_i^{\text{RP}}(x_i),-g_i^{\text{RP}}(\underline{t},x_i)\}\\ H_i^{\text{RP}}(x_i,\lambda)&=\max_{u_i\in\mathcal{U}_i}\min_{d_i\in\mathcal{D}_i}\lambda\cdot f_i(x_i,u_i,d_i)\end{aligned}\tag{54}$$

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* by designing buffer regions during replanning as well.

We conclude this section by summarizing the overall SPP algorithm:

Algorithm 2: **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) compute the avoid region of Q_i using (14). The optimal control to avoid the intruder can be obtained by using (16).
2) determine the separation region and buffer regions given by (21), (24) and (36).
3) compute the induced obstacles for Q_i by Q_j, given by (27), (32), (40), (45) and (46).
4) compute the total obstacle set $\mathcal{G}_i(t)$, given in (48). In the case $i=1$, $\mathcal{G}_i(t)=\mathcal{V}_i^{\text{S}}(t,t+t^{\text{IAT}})\ \forall t$;
5) given $\mathcal{G}_i(t)$, compute the BRS $\mathcal{V}_i^{\text{PP}}(t,t_i^{\text{STA}})$ defined in (49); the nominal control strategy is given by (51);

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 (53). 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

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 1.

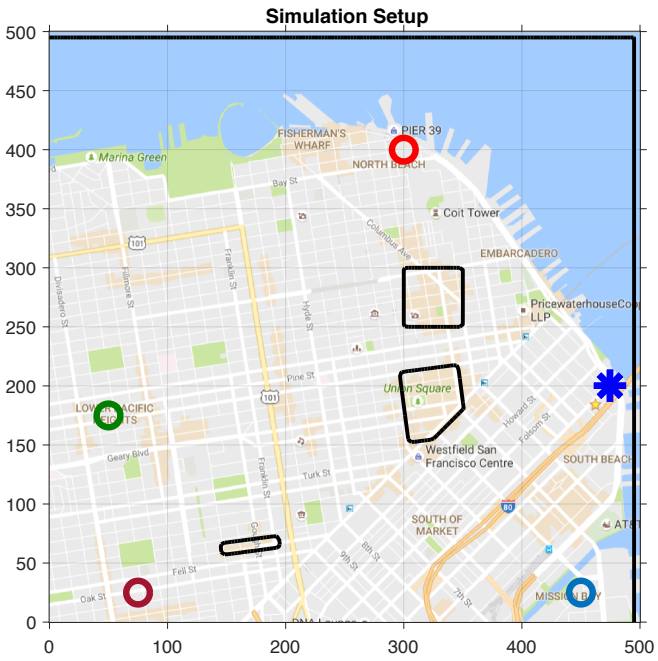

Fig. 1: Simulation setup. A $25km^2$ area of San Francisco city is used as the state-space for vehicles. SPP vehicles originate from the Blue star and go to one of the four destinations, denoted by circles. Tall buildings in the downtown area are used as static obstacles, represented by the black contours.

Each box in Figure 1 represents a $500m\times500m$ 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

$$\mathcal{L}_i=\{x_i:\|p_i-c_i\|_2\leq r\}\tag{55}$$

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 1. The destination of each vehicle is chosen randomly from these four destinations. Finally, tall buildings in downtown San Francisco are used as static obstacles for the SPP vehicles, denoted by black contours in Figure 1.

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

$$\begin{aligned}\dot{p}_{x,i}&=v_i\cos\theta_i+d_{x,i}\\ \dot{p}_{y,i}&=v_i\sin\theta_i+d_{y,i}\\ \dot{\theta}_i&=\omega_i,\\ \underline{v}\leq v_i\leq\bar{v},\ |\omega_i|\leq\bar{\omega},\ &\|(d_{x,i},d_{y,i})\|_2\leq d_r,\end{aligned}\tag{56}$$

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. 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 $d_r = 6m/s$, which corresponds to *moderate 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 (56). 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.

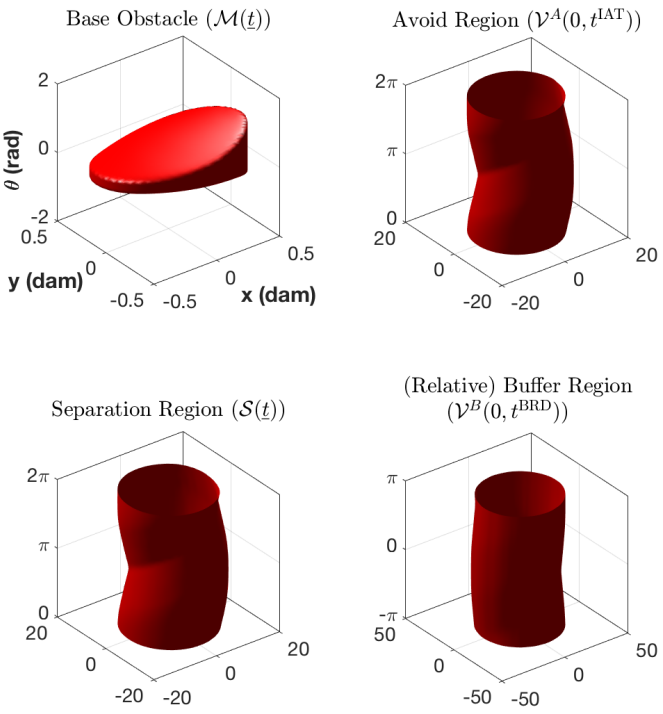

Fig. 2: Base obstacle $\mathcal{M}(t)$, Avoid region $\mathcal{V}^{\text{A}}(0, t^{\text{IAT}})$, Separation region $\mathcal{S}(t)$ and Relative buffer region $\mathcal{V}^{\text{B}}(0, t^{\text{BRD}})$ for vehicles. The three axes represent three states of the vehicles.

B. Results

In this section, we present the simulation results for $\bar{k} = 3$; occasionally, we also compare the results for different $\bar{k}$s to highlight some key points about the proposed algorithm. As per Algorithm 2, we begin with computing the avoid region $\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$. To compute the avoid region, relative dynamics between Q_i and Q_I are required. Given dynamics in (56), the relative dynamics are given by [27]:

$$\begin{aligned} \dot{p}_{x,I,i} &= v_I \cos\theta_{I,i} - v_i + \omega_i p_{y,I,i} + d_{x,i} + d_{x,I} \\ \dot{p}_{y,I,i} &= v_i \sin\theta_{I,i} - \omega_i p_{x,I,i} + d_{y,i} + d_{y,I} \\ \dot{\theta}_{I,i} &= \omega_I - \omega_i, \end{aligned} \tag{57}$$

where $x_{I,i} = (p_{x,I,i}, p_{y,I,i}, \theta_{I,i})$ is the relative state between Q_I and Q_i. Given relative dynamics, the avoid region can be computed using (14). For all the BRS and FRS computations in this simulation, we use Level Set Toolbox [35]. Also, since the vehicle dynamics' are same across all vehicles, we will omit the vehicle index from sets wherever applicable. The avoid region $\mathcal{V}^{\text{A}}(0, t^{\text{IAT}})$ for SPP vehicles is shown in Figure 2.

As long as Q_I starts outside the avoid region, Q_i is guaranteed to avoid the intruder for a duration of t^{IAT}. Given $\mathcal{V}^{\text{A}}(0, t^{\text{IAT}})$, we can compute the minimum required detection range d^{A} given by (17), which turns out to be 100m in this case. So as long as the vehicles can detect the intruder within 100m, the proposed algorithm guarantees collision avoidance with the intruder as well as a safe transit to their respective destinations.

Next, we compute the separation and buffer regions between vehicles. For the computation of base obstacles, 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]. In presence of moderate winds, the obtained error bound is 5m. This means that given any trajectory of vehicle, winds can at most cause a deviation of 5m from this trajectory. Thus, any vehicle will be within a distance of 5m from the trajectory. The overall base obstacle $\mathcal{M}$ around the point $(0, 0, 0)$ is shown in Figure 2. The base obstacles induced by a higher priority vehicle are thus given by this set augmented on the nominal trajectory, the trajectory that a vehicle will follow if the intruder never appears in the system, and is obtained by executing the control policy $u_i^{\text{PP}}(\cdot)$ in (51).

Given $\mathcal{M}$ and $\mathcal{V}^{\text{A}}(0, t^{\text{IAT}})$, we compute the separation region $\mathcal{S}$ as defined in (21). Relative buffer region $\mathcal{V}^{\text{B}}(0, t^{\text{BRD}})$, defined in (34), is similarly computed. The results are shown in Figure 2. Finally, we compute the buffer region as defined in (24). The resultant buffer region is shown in Blue in Figure 3. Thus, if Q_j is inside the base obstacle set shown in Figure 2 and Q_i is outside the Blue region in Figure 3, we can ensure that the intruder will have to spend a duration of at least $t^{\text{BRD}} = 10/3$ to go from the boundary of the avoid region of Q_j to the boundary of the avoid region of Q_i.

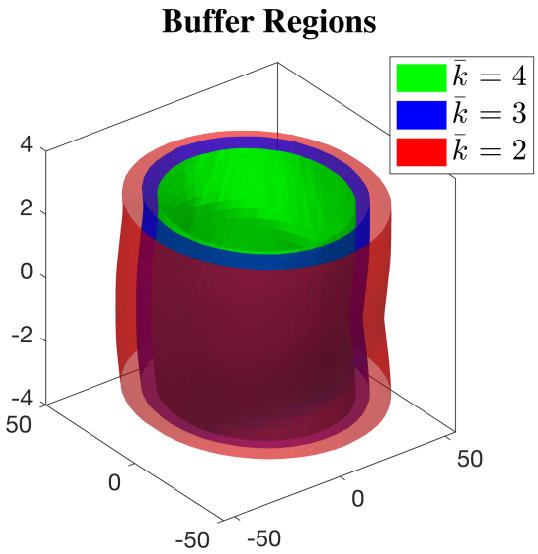

Fig. 3: Buffer regions for different $\bar{k}$ (best visualized with colors). As $\bar{k}$ decreases, a larger buffer is required between vehicles to ensure that the intruder spends more time while traveling through this buffer region so that it forces fewer vehicles to apply an avoidance maneuver.

We also computed the buffer regions for $\bar{k} = 2$ and $\bar{k} = 4$. The results are shown in Figure 3. Top-down views of these 3D sets are shown in Figure 4. As evident from the figures, a bigger buffer 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.

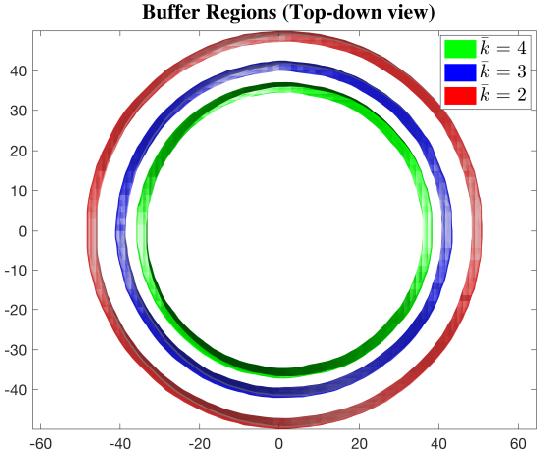

Fig. 4: Top-down view of the buffer regions for different $\bar{k}$ shown in Figure fig:buffRegions (best visualized with colors). As $\bar{k}$ decreases, t^{BRD} increase and a larger buffer is required between vehicles.

These buffer region computations along with the induced obstacle computations were similarly performed sequentially for each vehicle to obtain $\mathcal{G}(\cdot)$. This overall obstacle set was then used during their path planning and the control policy $u^{\text{PP}}(\cdot)$ was computed, as defined in (51). Finally, the corresponding nominal trajectories were obtained by executing control policy $u^{\text{PP}}(\cdot)$. The nominal trajectories and the overall obstacles for different vehicles at an arbitrary time are shown in Figure 5. The numbers in the figure represent the vehicle numbers. The nominal trajectories (solid lines) are well separated from each other to ensure collision avoidance even during a worst-case intruder "attack". This low density of vehicles is also required to ensure that the intruder cannot force more 3 vehicles to apply an avoidance maneuver. This is also evident from large obstacles induced by that vehicle for the lower priority vehicles (dashed circles). Thus, this lower density of vehicles is the price that we pay for ensuring that the replanning can be done efficiently in real-time. We discuss this further in section V-C.

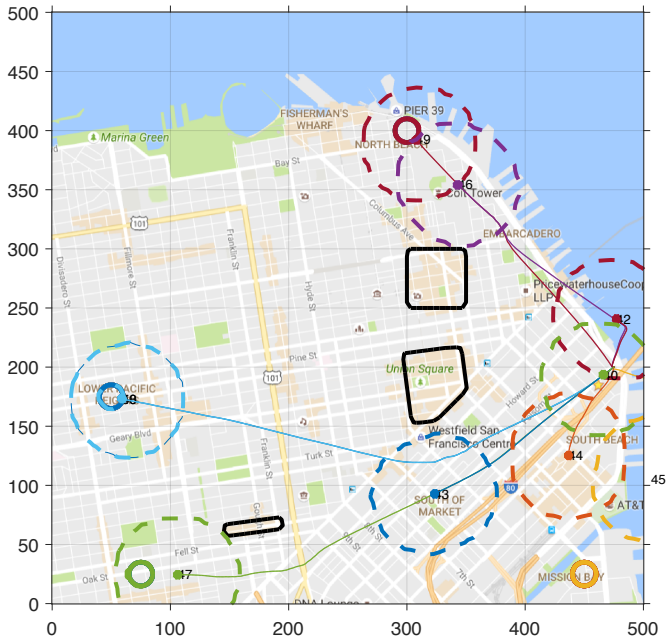

Fig. 5: Nominal trajectories and induced obstacles by different vehicles. The nominal trajectories (solid lines) are well separated from each other to ensure that the intruder cannot force more than 3 vehicles to apply an avoidance maneuver.

In the absence of an intruder the vehicles transit successfully to their destinations with control policy $u^{\text{PP}}(\cdot)$, but they can deviate from the shown nominal trajectories if an intruder appears in the system. In Figure 6, we plot the distance between an SPP vehicle and the intruder when the vehicle applies the control policy $u^{\text{PP}}(\cdot)$ (Red line) vs when it applies u^{A} (Blue line). Black dashed line represents the collision radius $r = 100$m between the vehicle and the intruder. As evident from the figure, if the vehicle continues to apply the control policy $u^{\text{PP}}(\cdot)$ in the presence of an intruder, the intruder enters in its danger zone. Thus, it is forced to apply the avoidance control, which can cause a deviation from the nominal trajectory, but will successfully avoid the intruder.

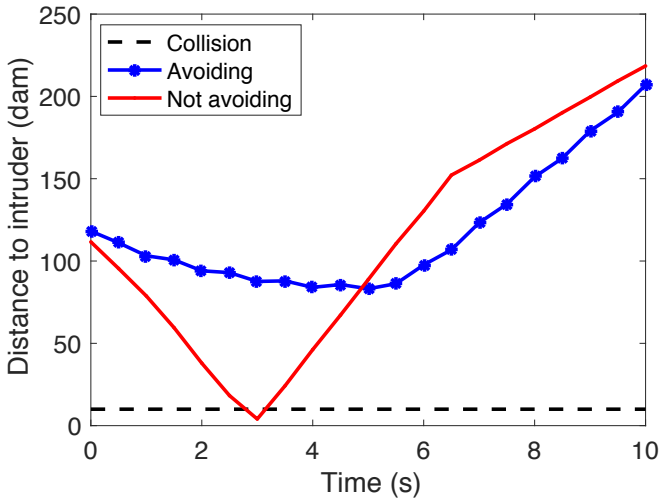

Fig. 6: The trajectory of a SPP vehicle when it applies the nominal controller vs when it applies the avoidance control. The vehicle is forced to apply the avoidance maneuver in the presence of an intruder, which can cause vehicle's deviation from its nominal trajectory.

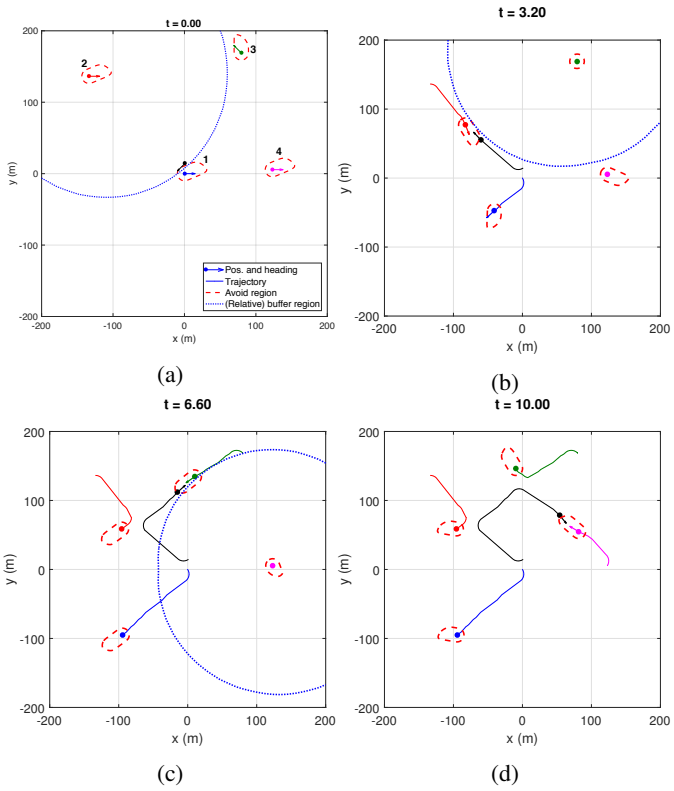

Fig. 7: Illustration of the intruder strategy to force maximum number of vehicles to apply an avoidance maneuver and hence to replan their trajectories. Q_I is able to force $\bar{k}$ vehicles to apply an avoidance control if the vehicles are applying the worst control which takes it closer to the intruder while the intruder is trying to reach its avoid region boundary.

Under the proposed algorithm, the intruder will affect the 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 reaches 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 forces maximum vehicles to apply an avoidance maneuver during a duration of t^{IAT}. This is illustrated for a small simulation of 4 vehicles in Figure 7. In this case at $\underline{t}=0$, Q_I (Black vehicle) appears at the boundary of the avoid region of Q_1 (Blue vehicle) (see Figure 7a). Immediately, it travels through the buffer region between Q_1 and Q_2 and at $t=\underline{t}+t^{\text{BRD}}=3.33$, reaches the boundary of the avoid region of Q_2 (Red vehicle), as shown in Figure 7a. The trajectories that Q_1 will follow while applying the avoidance control, Q_2 and Q_I will follow while trying to collide with each other are also shown. Following the same strategy, Q_I reaches the boundary of the avoid region of Q_3 (Green vehicle) at $t=\underline{t}+2t^{\text{BRD}}=6.67$, and will just barely reach the boundary of the avoid region of Q_4 (Pink vehicle) at $t=10$. However, it won't be able to force Q_4 to apply an avoidance maneuver as the duration of t^{IAT} will be over by then. Thus the avoid start time of the four vehicles are given as $\underline{t}_1=0$, $\underline{t}_2=3.33$, $\underline{t}_3=6.66$ and $\underline{t}_4=\infty$. The set of vehicles that will need to replan their trajectories after the intruder disappears is given by $\mathcal{N}^{\text{RP}}=\{Q_1,Q_2,Q_3\}$. As expected, $|\mathcal{N}^{\text{RP}}|\leq 3$.

The relative buffer region between vehicles is computed under the assumption that both the SPP vehicle and the intruder are trying to collide with each other; this is to ensure that the intruder will need at least a duration of t^{BRD} to reach the boundary of the avoid region of the next vehicle, irrespective of the control applied by the vehicle. However, this is not necessarily true as a vehicle will be applying the control policy $u^{\text{PP}}(\cdot)$ unless the intruder forces it to apply an avoidance maneuver, which may not necessarily correspond to the policy that the vehicle will use to *deliberately* collide with the intruder. Therefore, it is very likely that the intruder will need a bigger duration to reach the boundary of the avoid region of next vehicle, and hence it will be able to affect less than $\bar{k}$ vehicles even with its best strategy to affect maximum vehicles. This is also evident from Figure 8. In this case, Q_I again appears at the boundary of the avoid region of Q_1 at $t=0$, as shown in Figure 8a. The respective targets of the vehicles are also shown. Following its best strategy, the intruder immediately moves to travel through the buffer region between Q_1 and Q_2. However, now Q_2 is applying the control policy $u^{\text{PP}}(\cdot)$, i.e. it is trying to reach its target, unless the intruder reaches the boundary of its avoid region, which doesn't happen until $t=6.4$. Now, intruder again tries to travel through the avoid region of Q_2 and Q_3, but is not able to reach the boundary of the avoid region of Q_3 before it disappears at $t=10$. Thus, the intruder is able to force only

two vehicles to apply an avoidance maneuver. The avoid start time of the four vehicles are given as $\underline{t}_1 = 0, \underline{t}_2 = 6.4, \underline{t}_3 = \infty$ and $\underline{t}_4 = \infty$. The set of vehicles that will need to replan their trajectories is given by $\mathcal{N}^{\text{RP}} = \{Q_1, Q_2\}$. This conservatism in our method is discussed further in Section V-C.

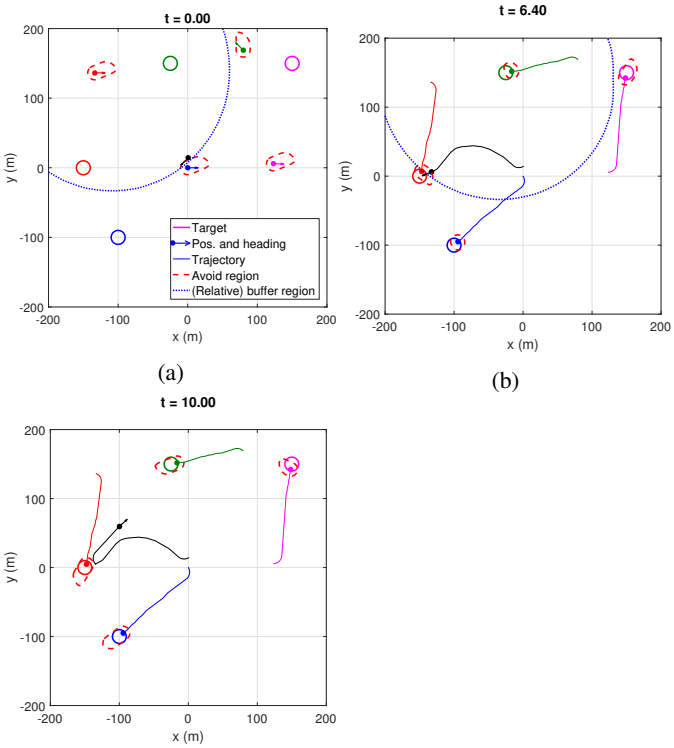

Fig. 8: Illustration of the intruder strategy to force maximum number of vehicles to apply an avoidance maneuver and hence to replan their trajectories. Since a vehicle's nominal controller might be different from the worst case controller that is assumed while computing the buffer region, Q_I is very likely to be able to force less than $\bar{k}$ vehicles to apply an avoidance maneuver despite its best strategy.

C. Discussion

The shown simulations illustrate the effectiveness of reachability in ensuring that the SPP vehicles safely reach their respective destinations even in the presence of an intruder. However, they also highlight some of the conservatism that is in-built in the reachability analysis due to the worst case analysis. For example, in the proposed algorithm, we assume the worst-case disturbances and intruder behavior while computing the buffer region and induced obstacles, which results in a large separation between vehicles and hence a lower vehicle density overall, as evident from Figure fig:trajObsSim. Similarly, while computing the relative buffer region, we assumed that a vehicle is *deliberately* trying to collide with the intruder so we once again consider the worst case scenario, as the vehicle will only be applying the nominal control strategy $u^{\text{PP}}(\cdot)$, which may not be same as the worst-case control strategy. Therefore, even though this worst-case analysis is essential to guarantee safety regardless the actions of SPP vehicles, the intruder and disturbances, a probabilistic safety analysis, that can overcome some of this conservatism, might be more suitable in practical applications.

VI. Conclusion and Future Work

We propose an algorithm to account for an adversarial intruder in sequential path planning. All vehicles are guaranteed to successfully reach their respective destinations without entering each other's danger zones despite the worst-case disturbance and the intruder attack the vehicles could experience.The proposed method ensures that only a fixed number of vehicles need to replan their trajectories once the intruder disappears, irrespective of total number of vehicles. Moreover, this fixed number is an input to the algorithm and hence can be chosen such that the replanning process is feasible in real-time. The proposed method is illustrated in a fifty-vehicle simulation, set in the urban environment of San Francisco city in California, US. Future work includes exploring methods that can account for multiple simultaneous intruders and can reduce conservatism in the current analysis.

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