Safe and Resilient Multi-vehicle Trajectory Planning Under Adversarial Intruder

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

***Abstract*—Provably safe and scalable multi-vehicle trajectory planning is an important and urgent problem. 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 multi-vehicle trajectory planning is often intractable due to the "curse of dimensionality." To overcome this problem, the sequential trajectory planning (STP) method, which assigns strict priorities to vehicles, was proposed; STP allows multi-vehicle trajectory planning to be done with a linearly-scaling computation complexity. However, if a vehicle not in the set of STP vehicles enters the system, or even worse, if this vehicle is an adversarial intruder, the previous formulation requires the entire system to perform replanning, an intractable task for large-scale systems. In this paper, we make STP more practical by providing a new algorithm where replanning is only needed only for a fixed number of vehicles, irrespective of the total number of STP vehicles. Moreover, this number is a design parameter, which can be chosen based on the computational resources available during run time. We demonstrate this algorithm in a representative simulation of an urban airspace environment.**

I. Introduction

Recently, there has been an immense surge of interest in the use of unmanned aerial systems (UASs) for civil applications [1]–[5], which will involve unmanned aerial vehicles (UAVs) flying in urban environments, potentially in close proximity to humans, other UAVs, and other important assets. As a result, new scalable ways to organize an airspace are required 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 those 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 is in 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, methods to flexibly plan provably safe and dynamically feasible trajectories without making strong assumptions on the vehicles' dynamics and other vehicles' motion are lacking. Moreover, any trajectory planning scheme that addresses collision avoidance must also guarantee both goal satisfaction and safety of UAVs despite disturbances 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.

Hamilton-Jacobi (HJ) reachability-based methods [26]–[31] are particularly suitable in the context of UAVs because of the formal guarantees provided. In this context, 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 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, trajectory planning problems, and optimal control problems [36]–[39]. However, reachable set computations involve solving a HJ partial differential equation (PDE) or variational inequality (VI) on a grid representing a discretization of the state space, resulting in an *exponential* scaling of computational complexity with respect to the system dimensionality. Therefore, reachability analysis or other dynamic programming-based methods alone 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 possible high density of vehicles that needs to be accommodated [7].

To overcome this problem, the priority-based Sequential Trajectory Planning (STP) method has been proposed [40], [41]. In this context, higher-priority vehicles plan their trajectories without taking into account the lower-priority vehicles, and 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 trajectories for each vehicle, starting from the highest-priority vehicle. Thus, the curse of dimensionality is overcome at the cost of a structural assumption, under which the computation complexity scales just *linearly* with the number of vehicles. In addition, such a structure has the potential to flexibly divide up the airspace for the use of many UAVs and allows tractable multi-vehicle trajectory-planning. Practically, different economic mechanisms can be used to establish a priority order. One example could be first-

This research is supported by NSF under CPS:ActionWebs (CNS-931843), under the CPS Frontiers VehiCal project (1545126), by the UC-Philippine-California Advanced Research Institute under project IIID-2016-005, and by the ONR MURI Embedded Humans (N00014-16-1-2206).

* 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

Algorithm 1: Overview of the proposed intruder avoidance algorithm (planning phase)

input : Set of vehicles $Q_i, i = 1, \ldots, N$ in the descending priority order;
Vehicle dynamics and initial states;
Vehicle destinations and any obstacles to avoid;
Intruder dynamics;
$\bar{k}$: Maximum number of vehicles allowed to re-plan their trajectories.

output: Provably safe vehicle trajectories to respective destinations despite disturbances and intruder;
Intruder avoidance and goal-satisfaction controller.

for $i = 1 : N$ **do**
- compute the separation region of Q_i;
- compute the required buffer region based on $\bar{k}$;
- use STP algorithm for trajectory planning of Q_i such that the buffer region is maintained between Q_i and Q_j for all $j < i$;
- output the trajectory and the optimal controller for Q_i.

come-first-serve mechanism, as highlighted in NASA's concept of operations for UAS traffic management [7].

However, if a vehicle not in the set of STP vehicles enters the system, or even worse, if this vehicle has malicious intent, the original plan can lead to a vehicle colliding with another vehicle, leading to a domino effect, causing the entire STP structure to collapse. Thus, STP vehicles must plan with an additional safety margin that takes a potential intruder into account. The authors in [42] propose an STP algorithm that accounts for such a potential intruder. However, a new full-scale trajectory planning problem is required to be solved in real time to ensure safe transit of the vehicles to their respective destinations. Since the replanning must be done in real-time, the proposed algorithm in [42] is intractable for large-scale systems even with the STP structure. In this work, we propose a novel algorithm that limits the replanning to a *fixed number of vehicles*, irrespective of the total number of STP vehicles. Moreover, this design parameter can be chosen beforehand based on the computational resources available.

Intuitively, for every vehicle, we compute a *separation region* such that the vehicle needs to account for the intruder if and only if the intruder is inside this separation region. We then compute a *buffer region* between the separation regions of any two vehicles, and ensure that this buffer is maintained as vehicles are traveling to their destinations. Thus, to intrude every additional vehicle, the intruder must travel through the buffer region. Therefore, we can design the buffer region size such that the intruder can affect at most a specified number of vehicles within some duration. A high-level overview of the proposed algorithm is provided in Algorithm 1.

In Section II, we formalize the STP problem in the presence of disturbances and adversarial intruders. In Section III, we present an overview of time-varying reachability and basic STP algorithms in [40], [41]. In Section IV, we present our proposed algorithm. Finally, we illustrate this algorithm through a fifty-vehicle simulation in an urban environment in Section V. Notations are summarized in Table I.

II. Sequential Trajectory Planning Problem

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

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

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 $\mathcal{O}_i^{\text{static}} \subset \mathbb{R}^{n_i}$, which could represent any set of states, such as positions of tall buildings, that are forbidden. Each vehicle Q_i must also avoid the danger zones with respect to every other vehicle $Q_j, j \neq i$. For simplicity, we define the danger zone of Q_i with respect to Q_j to be

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

The danger zones in general can represent any joint configurations between Q_i and Q_j that are considered to be unsafe. 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 Q_I may also appear in the system. An intruder vehicle may have malicious intent or simply be a non-participating vehicle that could accidentally collide with other vehicles. This general definition of intruder allows us to develop algorithms that can also account for vehicles who are not communicating with the STP vehicles or do not know about the STP structure. In general, the effect of intruders on vehicles in structured flight can be unpredictable, since the intruders in principle could be adversarial in nature, and the number of intruders could be arbitrary, in which case a collision avoidance problem must be solved for each STP vehicle in the joint state-space of all intruders and the STP vehicle. Therefore, to make our analysis tractable, we make the following two assumptions.

Assumption 1: At most one intruder affects the STP vehicles at any given time. The intruder is removed after a 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 planning space. For example, when STP vehicles are flying at a particular altitude level, the removal of the intruder can be achieved by forcing the intruder to exit the altitude level. Practically, over a large region of the unmanned airspace, this assumption implies that there would be one intruder vehicle per "planning region".

Each planning region would perform STP independently from the others. One would design planning regions to be an appropriate size such that it is reasonable to assume at most one intruder would appear. The entire large region would be composed of several planning regions.

Let the time at which intruder appears in the system be $\underline{t}$. Assumption 1 implies that any vehicle Q_i would need to avoid the intruder Q_I for a maximum duration of t^{IAT}. Note that we do not pose any restriction on $\underline{t}$; we only 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 we only assume that the dynamics of the intruder are known, but its initial state x_I^0, control u_I and disturbance d_I it experiences are unknown.

Given the set of STP 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 as follows. For each vehicle Q_i, 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.

Due to the high dimensionality of the joint state-space, a direct dynamic programming-based solution is intractable. Therefore, the authors in [40] proposed to assign a priority to each vehicle, and perform STP given the assigned priorities. Without loss of generality, let Q_j have a higher-priority than Q_i if $j < i$. Under the STP scheme, higher-priority vehicles can ignore the presence of lower-priority vehicles, and perform trajectory 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 straightforward 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, then the set of STP vehicles $Q_i, i = 1, \ldots, N$ would all be able to reach their targets safely. Under the STP scheme, trajectory planning can be done sequentially in descending order of vehicle priority in the state space of only a single vehicle. Thus, STP provides a solution whose complexity scales linearly with the number of vehicles. It is important to note that the trajectory planning is always feasible for the lower-priority vehicle under STP because a lower-priority vehicle can always depart early to avoid the higher-priority vehicle on its way to its destination.

When an intruder appears in the system, STP vehicles may need to avoid the intruder to ensure safety. Depending on the initial state of the intruder and its control policy, a vehicle will potentially need to apply different avoidance controls leading to different final states after avoiding the intruder. Therefore, a vehicle's control policy that ensures its successful transit to its destination needs to account for all such possible final states, which is a trajectory 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 avoids the intruder otherwise (Algorithm 1). (ii) After the intruder disappears from the system, we replan the trajectories of the affected vehicles. Following the same theme and assumptions, the authors in [42] present an algorithm to avoid an intruder in STP formulation; however, in the worst-case, the algorithm might need to replan the trajectories for *all* STP vehicles. Our goal in this work is to present an algorithm that ensures that only a *small and fixed* number of vehicles needs to replan their trajectories, regardless of the total number of vehicles, resulting in a constant replanning time. 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, its control strategy, and disturbances it experiences?
2) How can we ensure that replanning only needs to be done for at most a chosen fixed maximum number of vehicles after the intruder disappears from the system?

III. Background

In this section, we first present the basic STP algorithm [40] in which disturbances and the intruders are ignored and perfect information of vehicles' positions is assumed. We then 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 $\mathcal{G}(t)$ which captures trajectories of higher-priority vehicles, 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 $\mathcal{V}(t, t_f)$ or $\mathcal{W}(t_0, t)$ respectively. Typically, when computing the BRS, t_f will be some fixed final time, for example the scheduled time of arrival t^{STA}. When computing the FRS, t_0

TABLE I: Mathematical notation and their interpretation (in the alphabetical order of symbols).

Notation	Description	Location	Interpretation
$\mathcal{B}_{ij}(t)$	Buffer region between vehicle j and vehicle i	Beginning of Section IV-B2	The set of all possible states for which the separation requirement may be violated between vehicle j and vehicle i for some intruder strategy. If vehicle i is outside this set, then the intruder will need atleast a duration of t^{BRD} to go from the avoid region of vehicle j to the avoid region of vehicle i.
d_i	Disturbance in the dynamics of vehicle i	Beginning of Section II	-
d_I	Disturbance in the dynamics of the intruder	Assumption 2	-
f_i	Dynamics of vehicle i	Beginning of Section II	-
f_I	Dynamics of the intruder	Assumption 2	-
f_r	Relative dynamics between two vehicles	Equation (12)	-
$\mathcal{G}_i(t)$	The overall obstacle for vehicle i	Equation (7)	The set of states that vehicle i must avoid on its way to the destination.
h_i	Non-position state component of vehicle i	Beginning of Section II	-
$\bar{k}$	-	Beginning of Section IV	The maximum number of vehicles that should apply the avoidance maneuver or the maximum number of vehicles that we can replan trajectories for in real-time.
$\mathcal{L}_i$	Target set of vehicle i	Beginning of Section II	The destination of vehicle i.
$\mathcal{M}_j(t)$	Base obstacle induced by vehicle j at time t	Equations (25), (31) and (37) in [42]	The set of all possible states that vehicle j can be in at time t if the intruder does not appear in the system till time t.
N	Number of STP vehicles	Beginning of Section II	-
$\mathcal{N}^{\text{RP}}$	-	Equation (39)	The set of vehicles that need to replan their trajectories after the intruder disappears. These are also the set of vehicles that were forced to apply an avoidance maneuver.
$\mathcal{O}_i^j(t)$	Induced obstacle by vehicle j for vehicle i	After Assumption 2 in Section II	The possible states of vehicle i such that a collision between vehicle i and vehicle j or vehicle i and the intruder vehicle (if present) could occur.
$\mathcal{O}_i^{\text{static}}$	Static obstacle for vehicle i	Beginning of Section II	Obstacles that vehicle i needs to avoid on its way to destination, e.g, tall buildings.
p_i	Position of vehicle i	Beginning of Section II	-
Q_i	ith STP vehicle	Beginning of Section II	-
Q_I	The intruder vehicle	Assumption 1	-
R_c	Danger zone radius	Equation (2)	The closest distance between vehicle i and vehicle j that is considered to be safe.
$\mathcal{S}_j(t)$	Separation region of vehicle j at time t	Beginning of Section IV-B1	The set of all states of intruder at time t for which vehicle j is forced to apply an avoidance maneuver.
$\underline{t}_i$	Avoid start time of vehicle i	Equation (18)	The first time at which vehicle i is forced to apply an avoidance maneuver by the intruder vehicle. Defined to be ∞ if vehicle i never applies an avoidance maneuver.
t^{BRD}	Buffer region travel duration	Beginning of Section IV	The minimum time required for the intruder to travel through the buffer region between any pair of vehicles.
t^{IAT}	Intruder avoidance time	Assumption 1	The maximum duration for which the intruder is present in the system.
$\underline{t}$	Intruder appearance time	After Assumption 1	The time at which the intruder appears in the system.
t_i^{LDT}	Latest departure time of vehicle i	End of Section II	The latest departure time for vehicle i such that it safely reaches its destination by the scheduled time of arrival.
t_i^{STA}	Scheduled time of arrival (STA) of vehicle i	Beginning of Section II	The time by which vehicle i is required to reach its destination.
u_i	Control of vehicle i	Beginning of Section II	-
u_I	Control of the intruder	Assumption 2	-
u_i^{A}	Optimal avoidance control of vehicle i	Equation (15)	The control that vehicle i need to apply to successfully avoid the intruder once the relative state between vehicle i and the intruder reaches the boundary of the avoid region of vehicle i.
u_i^{PP}	Nominal control	Equation (37)	The nominal control for vehicle i that will ensure its successful transition to its destination if the intruder does not force it to apply an avoidance maneuver. This control law corresponds to the nominal trajectory of vehicle i.
u_i^{RP}	The overall controller for vehicle i	Equation (40)	The overall controller for vehicle i that will ensure a successful and safe transit to its destination despite the worst-case intruder strategy.
$\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$	Avoid region of vehicle i	Equation (13)	The set of relative states x_{Ii} for which the intruder can force vehicle i to enter in the danger zone $\mathcal{Z}_{iI}$ within a duration of $(t^{\text{IAT}} - \tau)$.
$\mathcal{V}_i^{\text{B}}(0, t^{\text{BRD}})$	Relative buffer region	Beginning of Section IV-B2	The set of all states from which it is possible to reach the boundary of the avoid region of vehicle i within a duration of t^{BRD}.
$\mathcal{V}_i^{\text{PP}}$	-	Equation (35)	The set of all states that vehicle i needs to avoid in order to avoid a collision with the static obstacles while applying an avoidance maneuver.
$\mathcal{V}_i^{\text{S}}$	-	Equation (32)	The set of all initial states of vehicle i from which it is guaranteed to safely reach its destination if the intruder does not force it to apply an avoidance maneuver and successfully and safely avoid the intruder in case needs it does.
x_i	State of vehicle i	Beginning of Section II	-
x_I	State of the intruder vehicle	Assumption 2	-
x_i^0	Initial state of vehicle i	Beginning of Section II	-
x_I^0	Initial state of the intruder vehicle	Assumption 2	-
x_{Ii}	Relative state between the intruder and vehicle i	Equation (12)	-
$\mathcal{Z}_{ij}$	Danger zone between vehicle i and vehicle j	Equation (2)	Set of all states of vehicle i and vehicle j which are within unsafe distance of each other. The vehicles are said to have collided if their states belong to $\mathcal{Z}_{ij}$.

will be some fixed initial time, for example the starting time or the present time.

Several formulations of reachability are able to account for time-varying obstacles [29], [31] (or state constraints in general). For our application in STP, we utilize the formulation in [31], in which a BRS is computed by solving the following *final value* double-obstacle HJ VI:

$$\begin{aligned}\max\Big\{ \min\{D_t V(t,x) + H(t,x,\nabla V(t,x)), l(x) - V(t,x)\},\\ -g(t,x) - V(t,x)\Big\} = 0, \quad t \le t_f\\ V(t_f,x) = \max\{l(x), -g(t_f,x)\}\end{aligned} \tag{3}$$

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

$$\begin{aligned}D_t W(t,x) + H(t,x,\nabla W(t,x)) = 0, \quad t \ge t_0\\ W(t_0,x) = \max\{l(x), -g(t_0,x)\}\end{aligned} \tag{4}$$

In both (3) and (4), the function $l(x)$ is the implicit surface function representing the target set $\mathcal{L} = \{x : l(x) \le 0\}$. Similarly, the function $g(t,x)$ is the implicit surface function representing the time-varying obstacles $\mathcal{G}(t) = \{x : g(t,x) \le 0\}$. The BRS $\mathcal{V}(t,t_f)$ and FRS $\mathcal{W}(t_0,t)$ are given by

$$\begin{aligned}\mathcal{V}(t,t_f) = \{x : V(t,x) \le 0\}\\ \mathcal{W}(t_0,t) = \{x : W(t,x) \le 0\}\end{aligned} \tag{5}$$

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 (3). Also, note that unlike in (3), there is no inner minimization in (4). 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.

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 explicitly on t, we will drop t from the argument. In addition, the optimization of Hamiltonian gives 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. STP Without Disturbances and Intruder

In this section, we give an overview of the basic STP algorithm assuming that there is no disturbance and no intruder affecting the vehicles. The majority of the content in this section is taken from [40].

Recall that among the STP vehicles $Q_i, i = 1,\dots,N$, Q_j has a higher priority than Q_i if $j < i$. In the absence of disturbances, we can write the dynamics of the STP vehicles as

$$\dot{x}_i = f_i(x_i,u_i), t \le t_i^{\text{STA}}, \quad u_i \in \mathcal{U}_i, \tag{6}$$

In STP, each vehicle Q_i plans its trajectory while avoiding static obstacles $\mathcal{O}_i^{\text{static}}$ and the obstacles $\mathcal{O}_i^j(t)$ induced by higher-priority vehicles $Q_j, j < i$. Trajectory planning is done sequentially in descending priority, $Q_1, Q_2, \dots, Q_N$. During its trajectory planning process, Q_i ignores the presence of lower-priority vehicles $Q_k, k > i$. 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 trajectory 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 $\mathcal{L}_i$:

$$\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}} \cup \bigcup_{j=1}^{i-1} \mathcal{O}_i^j(t) \tag{7}$$

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

$$\mathcal{O}_i^j(t) = \{x_i : \|p_i - p_j(t)\|_2 \le R_c\} \tag{8}$$

Each higher-priority vehicle Q_j ignores Q_i. Since trajectory planning is done sequentially in descending order of priority, the vehicles $Q_j, j < i$ would have planned their trajectories 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 trajectory 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:

$$\begin{aligned}\mathcal{V}_i^{\text{basic}}(t,t_i^{\text{STA}}) = \{y : \exists u_i(\cdot) \in \mathbb{U}_i, x_i(\cdot) \text{ satisfies (6)},\\ \forall s \in [t,t_i^{\text{STA}}], x_i(s) \notin \mathcal{G}_i(s),\\ \exists s \in [t,t_i^{\text{STA}}], x_i(s) \in \mathcal{L}_i, x_i(t) = y\}\end{aligned} \tag{9}$$

The BRS $\mathcal{V}(t,t_i^{\text{STA}})$ can be obtained by solving (3) with $\mathcal{L} = \mathcal{L}_i$, $\mathcal{G}(t) = \mathcal{G}_i(t)$, and the Hamiltonian

$$H_i^{\text{basic}}(x_i,\lambda) = \min_{u_i \in \mathcal{U}_i} \lambda \cdot f_i(x_i,u_i) \tag{10}$$

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

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

from which the trajectory $x_i(\cdot)$ can be computed by integrating the system dynamics, which in this case are given by (6). In addition, the latest departure time t_i^{LDT} can be obtained from the BRS $\mathcal{V}(t,t_i^{\text{STA}})$ as $t_i^{\text{LDT}} = \arg\sup_t\{x_i^0 \in \mathcal{V}(t,t_i^{\text{STA}})\}$. The basic STP algorithm is summarized in Algorithm 2.

C. STP With Disturbances and Without Intruder

Disturbances and incomplete information significantly complicate the STP scheme. The main difference is that the vehicle dynamics satisfy (1) as opposed to (6). Committing to exact trajectories is therefore no longer possible, since

Algorithm 2: STP algorithm in the absence of disturbances and intruders

input : STP vehicles Q_i, their dynamics (6), initial states x_i^0, destinations $\mathcal{L}_i$, static obstacles $\mathcal{O}_i^{\text{static}}$

output: Provably safe trajectories to destinations and goal-satisfaction controllers $u^{\text{basic}}(\cdot)$

for $i = 1 : N$ **do**
- **Trajectory planning for** Q_i
- compute the total obstacle set $\mathcal{G}_i(t)$ given by (7). If $i = 1$, $\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}}\ \forall t$;
- compute the BRS $\mathcal{V}_i^{\text{basic}}(t, t_i^{\text{STA}})$ defined in (9);
- **Trajectory and controller of** Q_i
- compute the optimal controller $u_i^{\text{basic}}(\cdot)$ given by (11);
- determine the trajectory $x_i(\cdot)$ using vehicle dynamics (6) and the control $u_i^{\text{basic}}(\cdot)$;
- output the trajectory and optimal controller for Q_i.
- **Obstacles induced by** Q_i
- given the trajectory $x_i(\cdot)$, compute the induced obstacles $\mathcal{O}_k^i(t)$ given by (8) for all $k > i$.

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 in (8). 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, 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 $\mathcal{O}_i^j(t)$ that a higher-priority vehicle Q_j can be in based on the control strategy information known to the lower priority vehicle. A collision avoidance between Q_i and Q_j is thus ensured. We refer to the obstacle $\mathcal{O}_i^j(t)$, induced in the presence of disturbances but in the absence of intruders, as *base obstacle* and denote it as $\mathcal{M}_j(t)$ from here on. Further details of each algorithm are presented in [41].

IV. Response to Intruders

In this section, we propose a method to allow vehicles to avoid an intruder while maintaining the STP structure. Our goal is to design a control policy for each vehicle that ensures separation with the intruder and other STP vehicles, and ensures a successful transit to the destination.

As discussed in Section II, depending on the initial state of the intruder and its control policy, a vehicle may arrive at different states after avoiding the intruder. To make sure that the vehicle still reaches its destination, a replanning of vehicle's trajectory is required. Since the replanning must be done in real-time, we also need to ensure that only a small number of vehicles require replanning. In this work, a novel intruder avoidance algorithm is proposed, which will need to replan trajectories only for a *small fixed* number of vehicles, irrespective of the total number of STP vehicles. Moreover, this number is a design parameter, which can be chosen based on the resources available during run time.

Let $\bar{k}$ denote the maximum number of vehicles that we can replan the trajectories for in real-time. Also, let $t^{\text{BRD}} = \frac{t^{\text{IAT}}}{\bar{k}}$. We divide our algorithm in two parts: the planning phase and the replanning phase. In the planning phase, our goal is to divide the flight space of vehicles such that at any given time, any two vehicles are far enough from each other so that an intruder needs at least a duration of t^{BRD} to travel from the vicinity of one vehicle to that of another. Since the intruder is present for a total duration of t^{IAT}, this division ensures that it can only affect at most $\bar{k}$ vehicles despite its best efforts. In particular, we compute a separation region for each vehicle such that the vehicle needs to account for the intruder if and only if the intruder is inside this separation region. We then compute a buffer region between the separation regions of any two vehicles such that the intruder requires atleast a duration of t^{BRD} to travel through this region. A high-level overview of the planning phase is presented in Algorithm 1. The planning phase ensures that after the intruder disappears, *at most* $\bar{k}$ vehicles have to replan their trajectories. In the replanning phase, we re-plan the trajectories of affected vehicles so that they reach their destinations safely.

Note that our theory assumes worst-case scenarios in terms of the behavior of the intruder, the effect of disturbances, and the planned trajectories of each STP vehicle. This way, we are able to guarantee safety and goal satisfaction of all vehicles in all possible scenarios given the bounds on intruder dynamics and disturbances. To achieve denser operation of STP vehicles, known information about the intruder, disturbances, and specifies of STP vehicle trajectories may be incorporated; however, these considerations are out of the scope of this paper.

The rest of the section is organized as follows. In Sections IV-A, we discuss the intruder avoidance control that a vehicle needs to apply within the separation region. In Sections IV-B and IV-C, we compute the separation and buffer regions for vehicles. Trajectory planning that maintains the buffer region between every pair of vehicles is discussed in Section IV-D. Finally, the replanning of the trajectories of the affected vehicles is discussed in Section IV-E.

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 $\mathcal{Z}_{iI}$. We also compute the set of states from which the joint states of Q_I and Q_i can enter the danger zone $\mathcal{Z}_{iI}$ despite the best efforts of Q_i to avoid Q_I, which is then used to compute the separation region of Q_i in Section IV-B1.

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

$$x_{Ii} = x_I - x_i, \qquad \dot{x}_{Ii} = f_r(x_{Ii}, u_i, u_I, d_i, d_I) \tag{12}$$

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 backward reachable set $\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}}),\ \tau \in [0, t^{\text{IAT}}]$:

$$\begin{aligned}\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}}) =& \{y : \forall u_i(\cdot) \in \mathbb{U}_i, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_i(\cdot) \in \mathbb{D}_i,\\ & \exists d_I(\cdot) \in \mathbb{D}_I, x_{Ii}(\cdot) \text{ satisfies (12)},\\ & \exists s \in [\tau, t^{\text{IAT}}], x_{Ii}(s) \in \mathcal{L}_i^{\text{A}}, x_{Ii}(\tau) = y\},\\ \mathcal{L}_i^{\text{A}} =& \{x_{Ii} : \|p_{Ii}\|_2 \le R_c\}.\end{aligned} \tag{13}$$

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

$$H_i^{\text{A}}(x_{Ii}, \lambda) = \max_{u_i \in \mathcal{U}_i} \min_{\substack{u_I \in \mathcal{U}_I,\\ d_I \in \mathcal{D}_I,\\ d_i \in \mathcal{D}_i}} \lambda \cdot f_r(x_{Ii}, u_i, u_I, d_i, d_I). \tag{14}$$

We refer to $\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$ as *avoid region* from here on. The interpretation of $\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$ is that if Q_i starts inside this set, i.e., $x_{Ii}(t) \in \mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$, then the intruder can force Q_i to enter the danger zone $\mathcal{Z}_{iI}$ within a duration of $(t^{\text{IAT}} - \tau)$, regardless of the control applied by the vehicle. If Q_i starts at the boundary of this set (denoted as $\partial\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$), i.e., $x_{Ii}(t) \in \partial\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})$, it can *barely* successfully avoid the intruder for a duration of $(t^{\text{IAT}} - \tau)$ using the optimal avoidance control u_i^{A} (referred to as *avoidance maneuver* from here on)

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

Finally, if Q_i starts outside this set, i.e., $x_{Ii}(t) \in \left(\mathcal{V}_i^{\text{A}}(\tau, t^{\text{IAT}})\right)^C$, then Q_i and Q_I cannot instantaneously enter the danger zone $\mathcal{Z}_{iI}$, irrespective of the control applied by them at time t. In fact, Q_i can safely apply *any* control as long as it is outside the boundary of this set, but will have to apply the avoidance maneuver to avoid the intruder once it reaches the boundary.

In the worst case, Q_i might need to avoid the intruder for a duration of t^{IAT} starting at $t = \underline{t}$; thus, the least we must have is that $x_{Ii}(\underline{t}) \in \left(\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})\right)^C$ to ensure successful avoidance. Otherwise, regardless of what control a vehicle applies, the intruder can force it to enter the danger zone $\mathcal{Z}_{iI}$.

Assumption 3: $x_{Ii}(\underline{t}) \in \left(\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})\right)^C \forall i \in \{1, \ldots, N\}$.

Intuitively, assumption 3 enforces a condition on the detection of the intruder by STP vehicles. For example, if STP vehicles are equipped with circular sensors, then assumption 3 implies that STP vehicle must be able to detect a intruder that is within a distance of d^{A}, where

$$d^{\text{A}} = \max\{\|p_i\|_2 : \exists h_i, (p_i, h_i) \in \mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})\}; \tag{16}$$

otherwise, there exists 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 trajectory-planning algorithm to ensure a successful intruder avoidance for all intruder strategies. In general, assumption 3 is required to ensure that the intruder gives the STP vehicles "a chance" to react and avoid it. Hence, for analysis to follow, we assume that assumption 3 holds.

Note that although (15) gives us a provably successful avoidance control for avoiding the intruder if $x_{Ii}(\underline{t}) \in \left(\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})\right)^C$, the vehicle may not be able to apply this control because it may lead to a collision with other STP vehicles. Thus, in general, assumption 3 is *only necessary not sufficient* to guarantee intruder avoidance. However, we ensure that the STP vehicles are always separated enough from each other so that any vehicle can apply avoidance maneuver if need be. Thus, for the proposed algorithm, assumption 3 will also be sufficient for a successful intruder avoidance.

B. Separation and Buffer Regions - Case 1

In the next two sections, our goal is to compute separation and buffer regions for STP vehicles so that at most $\bar{k}$ vehicles are *forced* to apply an avoidance maneuver during the duration $[\underline{t}, \underline{t} + t^{\text{IAT}}]$.

Intuitively, we divide the duration of t^{IAT} into $\bar{k}$ intervals and ensure that atmost one vehicle can be forced to apply the avoidance maneuver during each interval. Thus, by construction, it is guaranteed that at most $\bar{k}$ vehicles apply the avoidance maneuver during the time interval $[\underline{t}, \underline{t} + t^{\text{IAT}}]$. We refer to this structure as the *separation requirement* from here on. To ensure this requirement, we use the reachability theory to find the set of all states of Q_i such that the separation requirement can be violated between the vehicle pair (Q_i, Q_j), $j < i$, at time t. During the trajectory planning of Q_i, we then ensure that the vehicle is not in one of these states at time t by using this set of states as "obstacle". The sequential trajectory planning will therefore guarantee that the separation requirement holds for every STP vehicle pair.

Mathematically, we define

$$\mathcal{A}_i := \{t : x_{Ii}(t) \in \partial\mathcal{V}_i^{\text{A}}(t - \underline{t}, t^{\text{IAT}}), t \in [\underline{t}, \underline{t} + t^{\text{IAT}}]\}, \tag{17}$$

and the *avoid start time* $\underline{t}_i$

$$\underline{t}_i = \begin{cases} \min_{t \in \mathcal{A}_i} t & \text{if } \mathcal{A}_i \neq \emptyset \\ \infty & \text{otherwise} \end{cases} \tag{18}$$

By definition of $\mathcal{V}_i^{\text{A}}(\cdot, t^{\text{IAT}})$, $\mathcal{A}_i$ is the set of all times at which Q_i must apply an avoidance maneuver and $\underline{t}_i$ denotes the *first* such time. The separation requirement can thus be written as

$$\forall i \neq j, \min(\underline{t}_i, \underline{t}_j) < \infty \implies |\underline{t}_i - \underline{t}_j| \ge t^{\text{BRD}} := \frac{t^{\text{IAT}}}{\bar{k}}. \tag{19}$$

The separation requirement essentially implies that the intruder requires a time duration of atleast t^{BRD} before it can force any additional vehicle to apply an avoidance maneuver. Thus, any two STP vehicles should be "separated" enough from each other at any given time for such a requirement to hold.

We now focus on finding what that "separation set" (also referred to as the buffer region) between (Q_i, Q_j). Since the path planning is done in a sequential order, we assume that we have already planned a path for Q_j and compute the buffer region that Q_i needs to maintain to ensure that the separation requirement is satisfied. For this computation, it is sufficient to consider the following two mutually exclusive and exhaustive cases:

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

In this section, we consider Case 1. Case 2 is discussed in the next section.

In Case 1, the intruder forces Q_j, the higher-priority vehicle, to apply avoidance control before or at the same time as Q_i, the lower-priority vehicle. 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* in the system at the boundary of the avoid region of Q_j, i.e., $x_{Ij}(\underline{t}) \in \partial\mathcal{V}_j^{\mathrm{A}}(0, t^{\mathrm{IAT}})$. Equivalently, we can assume that $\underline{t}_j = \underline{t}$.

Since $\underline{t}_j \leq \underline{t}_i$, Q_I reaches the boundary of the avoid region of Q_j before it reaches the boundary of the avoid region of Q_i. Furthermore, by the definition of the avoid region, vehicles Q_j and Q_i need not account for the intruder until it reaches the boundary of the avoid region of Q_j. Thus, it is sufficient to consider the worst case $\underline{t}_j = \underline{t}$.

1) Separation region: Recall that the separation region denotes the set of states of the intruder for which a vehicle is forced to apply an avoidance maneuver. In this section our goal is to find $\mathcal{S}_j(\underline{t}_j)$, the separation region of Q_j at the avoid start time. By virtue of Observation 1, we will use $\underline{t}_j$ and $\underline{t}$ interchangeably here on.

As discussed in Section IV-A, Q_j needs to apply avoidance maneuver at time $\underline{t}$ only if $x_{Ij}(\underline{t}) \in \partial\mathcal{V}_j^{\mathrm{A}}(0, t^{\mathrm{IAT}})$. To compute set $\mathcal{S}_j(\underline{t})$, we thus need to translate these relative states to a set in the state space of the intruder. Therefore, if all possible states of Q_j at time $\underline{t}$ are known, then $\mathcal{S}_j(\underline{t})$ can be trivially computed.

Recall from Section III-C that the base obstacle $\mathcal{M}_j(t)$ at time t represents all possible states of Q_j at time t, if the intruder doesn't appear in the system until that time. This is precisely the set that we are interested in to compute the separation region. 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] (and Section III-C of this paper) to compute the base obstacles $\mathcal{M}_j(\underline{t})$. In particular, the base obstacles are respectively given by equations (25), (31) and (37) in [42] for the centralized control, the least restrictive control and the robust trajectory tracking algorithms (the three proposed algorithms to account for disturbances in STP). We will explain the computation of the base obstacles further in Section IV-D.

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

$$\mathcal{S}_j(\underline{t}) = \mathcal{M}_j(\underline{t}) + \partial\mathcal{V}_j^{\mathrm{A}}(0, t^{\mathrm{IAT}}),\ \underline{t} \in \mathbb{R}, \tag{20}$$

where the "+" in (20) denotes the Minkowski sum. Since $\mathcal{S}_j(\underline{t})$ represents the set of all states of Q_I for which Q_j must apply an avoidance maneuver, Observation 1 implies that it is sufficient to consider the scenarios where $x_I^0 := x_I(\underline{t}) \in \mathcal{S}_j(\underline{t})$.

2) Buffer Region: We are now ready to compute the buffer region $\mathcal{B}_{ij}$, the set of states for which the separation requirement can be violated between the vehicle pair (Q_i, Q_j). Conversely, if Q_i is outside the buffer region, the separation requirement is satisfied between (Q_i, Q_j). We start with recalling the following three results/facts

1) the separation requirement is equivalent to $|\underline{t}_i - \underline{t}| \geq t^{\mathrm{BRD}}$ (see 19)
2) $x_I(\underline{t}) \in \mathcal{S}_j(\underline{t})$ (Section IV-B1)
3) $x_{Ii}(\underline{t}_i) \in \partial\mathcal{V}_i^{\mathrm{A}}(\underline{t}_i - \underline{t}, t^{\mathrm{IAT}})$ (Definition of $\underline{t}_i$).

To compute $\mathcal{B}_{ij}$, we first compute $\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$, the set of all states x_{Ii} that can reach the set $\mathcal{V}_i^{\mathrm{A}}(t^{\mathrm{BRD}}, t^{\mathrm{IAT}})$ within a duration of t^{BRD}. Ensuring that the intruder is outside this set at time $\underline{t}$ guarantees that it will need a duration of atleast t^{BRD} before it can force Q_i to apply an avoidance maneuver (equivalently, reach the avoid region of Q_i). Since the possible states of the intruder at $\underline{t}$ is given by $\mathcal{S}_j(\underline{t})$, we thus simply need to ensure that $\mathcal{S}_j(\underline{t})$ is outside $\mathcal{V}_i^{\mathrm{B}}$. Thus, the minimum buffer region is given by:

$$\mathcal{B}_{ij}(\underline{t}) = \mathcal{S}_j(\underline{t}) + \mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}}). \tag{21}$$

We refer to $\mathcal{V}_i^{\mathrm{B}}$ as the *relative buffer region* here on, which is given by the following BRS:

$$\begin{aligned}\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_i(\cdot) \in \mathbb{D}_i,\\ & \exists d_I(\cdot) \in \mathbb{D}_I, x_{iI}(\cdot) \text{ satisfies (12)},\\ & \exists s \in [0, t^{\mathrm{BRD}}], x_{iI}(s) \in -\mathcal{V}_i^{\mathrm{A}}(t^{\mathrm{BRD}}, t^{\mathrm{IAT}}),\\ & x_{iI}(t) = y\},\\ -\mathcal{V}_i^{\mathrm{A}}(t^{\mathrm{BRD}}, t^{\mathrm{IAT}}) =& \{y : -y \in \mathcal{V}_i^{\mathrm{A}}(t^{\mathrm{BRD}}, t^{\mathrm{IAT}})\}.\end{aligned} \tag{22}$$

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

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

Intuitively, $\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$ represents the set of all relative states x_{iI} 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 the BRS $\mathcal{V}_i^{\mathrm{B}}(0, t^{\mathrm{BRD}})$ is computed using the relative state x_{iI} (and not x_{Ii}.)

To summarize, we can ensure that $(\underline{t}_i - \underline{t}_j) \geq t^{\mathrm{BRD}}$ as long as $x_i(\underline{t}) \in (\mathcal{B}_{ij}(\underline{t}))^C$. This will be ensured by using $\mathcal{B}_{ij}$ as an obstacle during the path planning of Q_i (see Section IV-D). Consequently, 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 buffer region between Q_i and Q_j such that the separation requirement is satisfied. However, it still needs to be ensured that a vehicle does not collide with other vehicles while applying an avoidance maneuver. In this section, we find the set of states that Q_i needs to avoid to avoid accidentally entering in $\mathcal{Z}_{ij}$ during an avoidance maneuver. Since the trajectory planning is done in a sequential fashion, being a lower priority vehicle, Q_i also needs to avoid the states that can lead it to $\mathcal{Z}_{ij}$ while Q_j is avoiding the intruder. These sets of states are then used as obstacles during the path planning of Q_i, which ensures that it never enters these "potentially unsafe" states.

To find this obstacle set, we consider the following two exhaustive cases:

1) Case A: The intruder affects Q_j, but not Q_i, i.e., $\underline{t}_j < \infty$ and $\underline{t}_i = \infty$.
2) Case B: The 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 $\mathcal{Z}_{ij}$ eventually. Let ${}^{A}_{1}\mathcal{O}^{j}_{i}(\cdot)$ and ${}^{B}_{1}\mathcal{O}^{j}_{i}(\cdot)$ 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 $\underline{t} \in [t - t^{\text{IAT}}, t]$. This is because if $\underline{t} < t - t^{\text{IAT}}$, 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.

- Case A: In this case, only Q_j applies an avoidance maneuver; therefore, Q_i should avoid the set of states 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), ${}^{A}_{1}\mathcal{O}^{j}_{i}(t)$ is given by the states that Q_j can reach while avoiding the intruder, starting from some state in the base obstacle, $\mathcal{M}_j(\underline{t}), \underline{t} \in [t - t^{\text{IAT}}, t]$. These states can be obtained by computing a FRS from the base obstacles.

$$\begin{aligned}\mathcal{W}^{\mathcal{O}}_{j}(\underline{t}, t) = \{y : \exists u_j(\cdot) \in \mathbb{U}_j, \exists d_j(\cdot) \in \mathbb{D}_j,\\ x_j(\cdot) \text{ satisfies (1)}, x_j(\underline{t}) \in \mathcal{M}_j(\underline{t}), x_j(t) = y\}.\end{aligned} \tag{24}$$

$\mathcal{W}^{\mathcal{O}}_{j}(\underline{t}, t)$ 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 (4) with the following Hamiltonian:

$$H^{\mathcal{O}}_{j}(x_j, \lambda) = \max_{u_j \in \mathcal{U}_j} \max_{d_j \in \mathcal{D}_j} \lambda \cdot f_j(x_j, u_j, d_j). \tag{25}$$

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

$$\begin{aligned}{}^{A}_{1}\mathcal{O}^{j}_{i}(t) &= \{x_i : \exists y \in \mathcal{P}_j(t), \|p_i - y\|_2 \le R_c\}\\ \mathcal{P}_j(t) &= \{p_j : \exists h_j, (p_j, h_j) \in \bigcup_{\underline{t} \in [t - t^{\text{IAT}}, t]} \mathcal{W}^{\mathcal{O}}_{j}(\underline{t}, t)\}\end{aligned} \tag{26}$$

Observation 3: Since the base obstacles represent all possible states of a vehicle 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 earlier time $\tau_1 < \tau_2$, computed forward for a duration of $(\tau_2 - \tau_1)$. That is, $\mathcal{M}_j(\tau_2) \subseteq \mathcal{W}^{\mathcal{O}}_{j}(\tau_1, \tau_2)$, where $\mathcal{W}^{\mathcal{O}}_{j}(\tau_1, \tau_2)$, 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., $\mathcal{W}^{\mathcal{O}}_{j}(\tau_2, \tau_3) \subseteq \mathcal{W}^{\mathcal{O}}_{j}(\tau_1, \tau_3)$ if $\tau_1 < \tau_2 < \tau_3$.

Using observation 3, $\mathcal{P}_j(t)$ in (26) can be equivalently written as

$$\mathcal{P}_j(t) = \{p_j : \exists h_j, (p_j, h_j) \in \mathcal{W}^{\mathcal{O}}_{j}(t - t^{\text{IAT}}, t)\}. \tag{27}$$

- Case B: In this case, first Q_j applies an avoidance maneuver followed by Q_i. 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 $[\underline{t}, \underline{t} + t^{\text{IAT}}]$. Furthermore, Q_i itself must apply avoidance maneuver during $[\underline{t}_i, \underline{t} + t^{\text{IAT}}]$. Thus, the main challenge in this case is to ensure that Q_i and Q_j do not enter into $\mathcal{Z}_{ij}$ even when both vehicles are applying avoidance maneuver and hence can apply *any* control from each other's perspective. Thus at time t, Q_i not only needs to avoid the states that Q_j could be in at time t, but also all the states that could lead it to $\mathcal{Z}_{ij}$ *in future* under some control actions of Q_i and Q_j. To compute this set of states, we make the following key observation:

Observation 4: For computing ${}^{B}_{1}\mathcal{O}^{j}_{i}(t)$, it is sufficient to consider $\underline{t}_i = t$. If $\underline{t}_i > t$, then Q_i is not applying any avoidance maneuver at time t and hence should only avoid the states that Q_j could be in at time t. However, this is already ensured during computation of ${}^{A}_{1}\mathcal{O}^{j}_{i}(t)$. 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 $\underline{t}_i - \underline{t}_j \ge t^{\text{BRD}}$. This along with Observation 4 implies that $\underline{t}_j \le t - t^{\text{BRD}}$. Also, from Observation 2, we have $\underline{t} = \underline{t}_j \ge t - t^{\text{IAT}}$. 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 $\mathcal{W}^{\mathcal{O}}_{j}(\underline{t}_j, t')$ 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, $\mathcal{W}^{\mathcal{O}}_{j}(\underline{t}_j, t')$ represents the FRS of $\mathcal{M}_j(\underline{t}_j)$ computed forward for a duration of $(t' - \underline{t}_j)$ and is given by (24).

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:

$$\mathcal{K}^{\text{B1}}(\tau) = \bigcup_{\underline{t}_j \in [\tau - t^{\text{IAT}}, t - t^{\text{BRD}}]} \mathcal{W}^{\mathcal{O}}_{j}(\underline{t}_j, \tau) \tag{28}$$

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 $\mathcal{W}^{\mathcal{O}}_{j}(\underline{t}_j, \tau) \subseteq \mathcal{W}^{\mathcal{O}}_{j}(\tau - t^{\text{IAT}}, \tau)$ for all $\underline{t}_j \in [\tau - t^{\text{IAT}}, t - t^{\text{BRD}}]$. Therefore, $\mathcal{K}^{\text{B1}}(\tau) = \mathcal{W}^{\mathcal{O}}_{j}(\tau - t^{\text{IAT}}, \tau)$.

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:

$$\begin{aligned}\mathcal{V}^{\text{B1}}_{i}(t, t - t^{\text{BRD}} + t^{\text{IAT}}) = \{&y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i,\\ &x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y,\\ &\exists s \in [t, t - t^{\text{BRD}} + t^{\text{IAT}}],\\ &x_i(s) \in \tilde{\mathcal{K}}^{\text{B1}}(s)\},\end{aligned} \tag{29}$$

where

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

The Hamiltonian H^{B1}_{i} to compute $\mathcal{V}^{\text{B1}}_{i}(\cdot)$ is given by

$$H^{\text{B1}}_{i}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i, d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \tag{30}$$

Finally, the induced obstacle in this case is given by

$$ {}^{B}_{1}\mathcal{O}^{j}_{i}(t) = \mathcal{V}^{\text{B1}}_{i}(t, t - t^{\text{BRD}} + t^{\text{IAT}}). \tag{31} $$

C. Separation and Buffer Regions - Case 2

We now consider Case 2: $\underline{t}_i < \underline{t}_j, \underline{t}_i < \infty$. In this case, the intruder forces Q_i, the lower-priority vehicle, to apply avoidance control before Q_j, the higher-priority vehicle. The separation region, the buffer region and the obstacles in this case can be computed in a similar manner to that in Case 1. The buffer region in this case is denoted as $\mathcal{B}_{ji}(t)$ to differentiate it from Case 1 (i.e., the order of i and j indexes has been switched). Similarly, the obstacles for Case 2 are denoted as ${}^{A}_{2}\mathcal{O}^{j}_{i}(t)$ and ${}^{B}_{2}\mathcal{O}^{j}_{i}(t)$, corresponding to the two cases similar to that in Section (IV-B3). For brevity purposes, this computation is presented in the Appendix.

D. Trajectory Planning

In this section, our goal is to plan the trajectory of each vehicle such that it is guaranteed to safely reach its target in the absence of an intruder, and to ensure collision avoidance with vehicles or obstacles if forced to apply an avoidance maneuver. We also need to make sure that the trajectories of the vehicles are such that the separation requirement is satisfied at all times. To obtain such a trajectory, we take into account all the "obstacles" computed in previous sections which ensure that the vehicle Q_i will not collide with any other vehicle, as long as it is outside these obstacles.

Before, we plan such a trajectory, we need to compute one final set of obstacles. In particular, we need to compute the set of states states that Q_i needs to avoid in order 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:

$$ \begin{aligned} \mathcal{V}^{\text{S}}_{i}(t, t + t^{\text{IAT}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y, \\ & \exists s \in [t, t + t^{\text{IAT}}], x_i(s) \in \mathcal{K}^{\text{S}}(s)\}, \\ \mathcal{K}^{\text{S}}(s) =& \{x_i : \exists (y, h) \in \mathcal{O}^{\text{static}}_{i}, \|p_i - y\|_2 \leq R_c\}. \end{aligned} \tag{32} $$

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

$$ H^{\text{S}}_{i}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i, d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \tag{33} $$

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

During the trajectory 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-C are used as obstacles in trajectory planning, then we can guarantee collision avoidance between Q_i and Q_j while they are avoiding the intruder. Thus, the overall obstacle for Q_i is given by:

$$ \begin{aligned} \mathcal{G}_i(t) =& \mathcal{V}^{\text{S}}_{i}(t, t + t^{\text{IAT}}) \bigcup \\ & \bigcup_{j=1}^{i-1} \left(\mathcal{B}_{ij}(t) \cup \mathcal{B}_{ji}(t) \bigcup_{k \in \{1,2\}} {}^{A}_{k}\mathcal{O}^{j}_{i}(t) \bigcup_{k \in \{1,2\}} {}^{B}_{k}\mathcal{O}^{j}_{i}(t) \right). \end{aligned} \tag{34} $$

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

$$ \begin{aligned} \mathcal{V}^{\text{PP}}_{i}(t, t^{\text{STA}}_{i}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \forall d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, \forall s \in [t, t^{\text{STA}}_{i}], x_i(s) \notin \mathcal{G}_i(s), \\ & \exists s \in [t, t^{\text{STA}}_{i}], x_i(s) \in \mathcal{L}_i, x_i(t) = y\}. \end{aligned} \tag{35} $$

The Hamiltonian H^{PP}_{i} to compute the BRS in (35) is given by:

$$ H^{\text{PP}}_{i}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i} \max_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \tag{36} $$

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:

$$ u^{\text{PP}}_{i}(t, x_i) = \arg \min_{u_i \in \mathcal{U}_i} \max_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \tag{37} $$

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}_{i} 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:

$$ u^{*}_{i}(t) = \begin{cases} u^{\text{PP}}_{i}(t) & t \leq \underline{t}_i \\ u^{\text{A}}_{i}(t) & \underline{t}_i \leq t \leq \underline{t} + t^{\text{IAT}} \end{cases} \tag{38} $$

If Q_i starts within $\mathcal{V}^{\text{PP}}_{i}$ and uses the control u^{*}_{i}, it is guaranteed to avoid collision with the intruder and other STP vehicles, regardless of the control strategy of Q_I. Finally, since we use separation and buffer regions as obstacles during the trajectory planning of Q_i, it is guaranteed that $|\underline{t}_i - \underline{t}_j| \geq t^{\text{BRD}}$ for all $j < i$. Therefore, atmost $\bar{k}$ vehicles are forced to apply an avoidance maneuver. The planning phase is summarized in Algorithm 3.

Remark 1: Note that given $\mathcal{V}^{\text{PP}}_{i}$ and u^{PP}_{i}, the base obstacles required for the computation of the separation region in Section IV-B1 can be computed using equations (25), (31) and (37) in [42]. Also, if we use the robust trajectory tracking method to compute the base obstacles, we would need to augment the obstacles in (34) by the error bound of Q_i, Ω_i (for details, see Section 5A-3 in [42]). The BRS in (35) in this case 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 STP 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 replanning is

Algorithm 3: The intruder avoidance algorithm: Planning-phase (offline planning)

input : Set of vehicles Q_i in the descending priority order, their dynamics (1) and initial states x_i^0; Vehicle destinations $\mathcal{L}_i$ and static obstacles $\mathcal{O}_i^{\text{static}}$; Intruder dynamics f_I and the maximum avoidance time t^{IAT} ; Maximum number of vehicles allowed to re-plan their trajectories $\bar{k}$.

output: The nominal controller u^{PP} and the avoidance controller u^{A} for all vehicles.

for $i = 1 : N$ **do**
- **Avoid region and avoidance control for** Q_i
- compute the avoid region $\mathcal{V}_i^{\text{A}}$ using (13);
- compute the avoidance controller u_i^{A} using (15);
- output the optimal avoidance controller u_i^{A} for Q_i.
- **if** $i \neq 1$ **then**
  - **Computation of separation region for** Q_i
  - **for** $j = 1 : i-1$ **do**
    - given the base obstacles $\mathcal{M}_j(\cdot)$ and the avoid region $\mathcal{V}_j^{\text{A}}$, compute the separation regions in (20) and (45);
  - **Computation of buffer region for** Q_i
  - **for** $j = 1 : i-1$ **do**
    - given the separation regions, compute the relative buffer regions $\mathcal{V}^{\text{B}}$ in (22) and (46);
    - given the relative buffer regions, compute the buffer regions in (21) and (48);
- **Computation of obstacles for** Q_i
- **if** $i \neq 1$ **then**
  - **for** $j = 1 : i-1$ **do**
    - given the base obstacles $\mathcal{M}_j(\cdot)$, compute the obstacles ${}_1^A\mathcal{O}_i^j(t)$ in (26), ${}_1^B\mathcal{O}_i^j(t)$ in (31), ${}_2^A\mathcal{O}_i^j(t)$ in (52), and ${}_2^B\mathcal{O}_i^j(t)$ in (57);
- compute the effective static obstacle to avoid ($\mathcal{V}_i^{\text{S}}$) using (32);
- **Trajectory planning for** Q_i
- compute the total obstacle set $\mathcal{G}_i(t)$ given by (34);
- compute the BRS $\mathcal{V}_i^{\text{PP}}(t, t_i^{\text{STA}})$ defined in (35);
- **The nominal controller of** Q_i
- compute the nominal controller $u_i^{\text{PP}}(\cdot)$ given by (37);
- output the nominal controller for Q_i.
- **Base obstacle induced by** Q_i
- given the nominal controller $u_i^{\text{PP}}(\cdot)$ and the BRS $\mathcal{V}_i^{\text{PP}}(t, t_i^{\text{STA}})$, compute the base obstacles $\mathcal{M}_i(\cdot)$ using equations (25), (31) or (37) in [42], depending on the information assumed to be known about the higher-priority vehicles.

required, $\mathcal{N}^{\text{RP}}$, can be obtained by checking if a vehicle Q_i applied any avoidance control during $[\underline{t}, \underline{t} + t^{\text{IAT}}]$, i.e.,

$$\mathcal{N}^{\text{RP}} = \{Q_i : \underline{t}_i < \infty, i \in \{1, \dots, N\}\}. \tag{39}$$

Note that due to the presence of separation and buffer regions, at most $\bar{k}$ vehicles can be affected by Q_I, i.e., $|\mathcal{N}^{\text{RP}}| \leq \bar{k}$. Goal satisfaction controllers which ensure that these vehicles reach their destinations can be obtained by solving a new STP problem, where the starting states of the vehicles are now given by the states they end up in, denoted $\tilde{x}_i^0$, after avoiding the intruder. 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. Moreover, flexible trajectory-planning algorithms such as FaSTrack [43] can be used that can perform replanning efficiently in real-time.

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

$$u_i^{\text{RP}}(t) = \begin{cases} u_i^*(t, x_i) & t \leq \underline{t} + t^{\text{IAT}} \\ u_i^{\text{L}}(t, x_i) & t > \underline{t} + t^{\text{IAT}} \end{cases} \tag{40}$$

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

$$\begin{aligned} \mathcal{W}_i^{\text{RP}}(\bar{t}, t) = \{&y \in \mathbb{R}^{n_i} : \exists u_i(\cdot) \in \mathbb{U}_i, \forall d_i(\cdot) \in \mathbb{D}_i, \\ &x_i(\cdot) \text{ satisfies (1)}, x_i(\bar{t}) = \tilde{x}_i^0, \\ &x_i(t) = y, \forall s \in [\bar{t}, t], x_i(s) \notin \mathcal{G}_i^{\text{RP}}(s)\}, \end{aligned} \tag{41}$$

where $\tilde{x}_i^0$ represents the state of Q_i at $t = \underline{t} + t^{\text{IAT}}$; $\mathcal{G}_i^{\text{RP}}(\cdot)$ 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 (7). The FRS in (41) can be obtained by solving

$$\begin{aligned} \max \Big\{ &D_t W_i^{\text{RP}}(t, x_i) + H_i^{\text{RP}}(t, x_i, \nabla W_i^{\text{RP}}(t, x_i)), \\ &- g_i^{\text{RP}}(t, x_i) - W_i^{\text{RP}}(t, x_i) \Big\} = 0 \\ W_i^{\text{RP}}(\underline{t}, x_i) &= \max\{l_i^{\text{RP}}(x_i), -g_i^{\text{RP}}(\underline{t}, x_i)\} \\ H_i^{\text{RP}}(x_i, \lambda) &= \max_{u_i \in \mathcal{U}_i} \min_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \end{aligned} \tag{42}$$

where $W_i^{\text{RP}}, g_i^{\text{RP}}, l_i^{\text{RP}}$ represent the FRS, obstacles during replanning, 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. The replanning phase is summarized in Algorithm 4.

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*.

V. Simulations

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

Algorithm 4: The intruder avoidance algorithm: Replanning-phase (real-time planning)

input : Set of vehicles $Q_i \in \mathcal{N}^{\text{RP}}$ that require replanning;
Vehicle dynamics (1) and new initial states $\tilde{x}_i^0$;
Vehicle destinations $\mathcal{L}_i$ and static obstacles $\mathcal{O}_i^{\text{static}}$;
Total base obstacle set $\mathcal{G}_i^{\text{RP}}(\cdot)$ induced by all other vehicles in $(\mathcal{N}^{\text{RP}})^C$.

output: The updated nominal controller u_i^{L} for all vehicles in $\mathcal{N}^{\text{RP}}$.

for $Q_i \in \mathcal{N}^{RP}$ **do**
  Computation of the updated t_i^{STA} for Q_i
  given the obstacle set $\mathcal{G}_i^{\text{RP}}(\cdot)$, compute the FRS $\mathcal{W}_i^{\text{RP}}(\bar{t}, t)$ in (41);
  the updated t_i^{STA} for Q_i is given by $\arg\inf_t\{\mathcal{W}_i^{\text{RP}}(\bar{t}, t) \cap \mathcal{L}_i \neq \emptyset\}$.
  Trajectory and controller of Q_i
  given the updated STA t_i^{STA}, the initial state $\tilde{x}_i^0$, the total obstacle set $\mathcal{G}_i^{\text{RP}}(\cdot)$, the vehicle dynamics (1) and the target set $\mathcal{L}_i$, use Algorithm 3 to replan the nominal trajectory and controller.

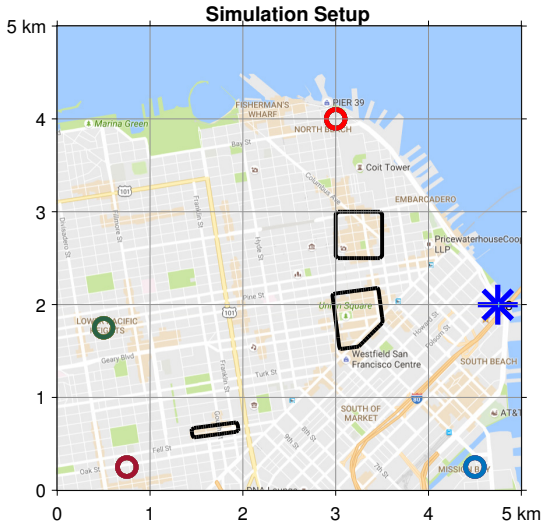

Fig. 1: Simulation setup. A 25 km^2 area of San Francisco city is used as the state-space for vehicles. STP vehicles originate from the Blue star and go to one of the four destinations, denoted by the circles. Tall buildings in the downtown area are used as static obstacles, represented by the black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

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 use the city of San Francisco (SF), California, USA as our planning region, as shown in Figure 1. Practically speaking, depending on the UAV density, it may be desirable to have smaller planning regions that together cover the SF area; however, such considerations are out of the scope of this paper.

Each box in Figure 1 represents a 500 m $\times$ 500 m 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 $\mathcal{L}_i = \{x_i : \|p_i - c_i\|_2 \leq 100\text{m}\}$, where c_i are centers of the target circles. 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, denoted by black contours in Figure 1. We use the following dynamics for each vehicle:

$$\begin{aligned} \dot{p}_{x,i} &= v_i \cos\theta_i + d_{x,i} \\ \dot{p}_{y,i} &= v_i \sin\theta_i + d_{y,i} \\ \dot{\theta}_i &= \omega_i, \\ \underline{v} \leq v_i \leq \bar{v},\ |\omega_i| &\leq \bar{\omega},\ \|(d_{x,i}, d_{y,i})\|_2 \leq d_r, \end{aligned} \tag{43}$$

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 and θ_i is the heading. $d = (d_{x,i}, d_{y,i})$ 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 $\underline{v}$ =0 m s^{-1}, $\bar{v}$ =25 m s^{-1}, $\bar{\omega}$ =2 rad s^{-1}, aligned with the modern UAV specifications [44], [45]. The disturbance bound is chosen as d_r =6 m s^{-1}, which corresponds to *moderate winds* on the Beaufort wind force scale [46]. 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^{IAT} =10 s, whose dynamics are given by (43). The joint state space of this fifty-vehicle system is 150-dimensional (150D); therefore, we assign a priority order to vehicles and solve the trajectory 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 STP vehicles so that they do not collide with each other or the intruder.

B. Results

In this section, we present the simulation results for $\bar{k} = 3$; occasionally, we also compare the results for different values of $\bar{k}$ to highlight some key insights about the proposed algorithm. As per Algorithm 3, 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 the dynamics in (43), the relative dynamics are given by [27]:

$$\begin{aligned} \dot{p}_{x,I,i} &= v_I \cos\theta_{I,i} - v_i + \omega_i p_{y,I,i} + d_{x,i} + d_{x,I} \\ \dot{p}_{y,I,i} &= v_i \sin\theta_{I,i} - \omega_i p_{x,I,i} + d_{y,i} + d_{y,I} \\ \dot{\theta}_{I,i} &= \omega_I - \omega_i, \end{aligned} \tag{44}$$

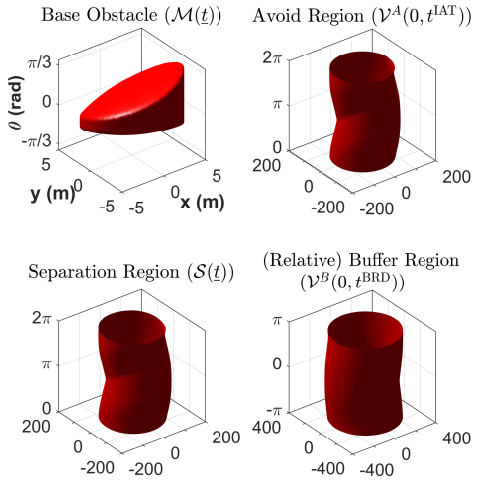

Fig. 2: Base obstacle $\mathcal{M}(t)$, Avoid region $\mathcal{V}^{\text{A}}(0, t^{\text{IAT}})$, Separation region $\mathcal{S}(t)$ and Relative buffer region $\mathcal{V}^{\text{B}}(0, t^{\text{BRD}})$ for vehicles. The three axes represent three states of the vehicles.

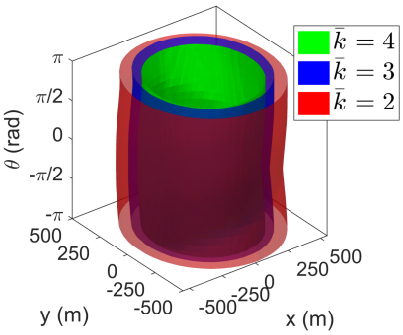

Fig. 3: Buffer regions for different $\bar{k}$ (best visualized with colors). As $\bar{k}$ decreases, a larger buffer is required between vehicles to ensure that the intruder spends more time traveling through this buffer region so that it forces fewer vehicles to apply an avoidance maneuver.

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 the relative dynamics, the avoid region can be computed using (13). 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 STP vehicles is shown in the top-right plot of Figure 2.

As long as Q_I starts outside the avoid region, Q_i is guaranteed to be able 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 (16) for the circular sensors, which turns out to be $100\,\text{m}$ in this case, corresponding to a detection of $4\,\text{s}$ in advance (given the speed of $25\,\text{m}\,\text{s}^{-1}$). So as long as the vehicles can detect the intruder within $100\,\text{m}$, 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 $5\,\text{m}$. This means that given any trajectory of vehicle, winds can at most cause a deviation of $5\,\text{m}$ from this trajectory. The overall base obstacle $\mathcal{M}$ around the point $(0, 0, 0)$ is shown in the top-left plot of 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 $u^{\text{PP}}(\cdot)$ in (37) for the higher-priority vehicles.

Given $\mathcal{M}$ of the higher-priority vehicles and $\mathcal{V}^{\text{A}}(0, t^{\text{IAT}})$, we compute the separation region $\mathcal{S}$ as defined in (20). Relative buffer region $\mathcal{V}^{\text{B}}(0, t^{\text{BRD}})$, defined in (22), is similarly computed. The results are shown in the bottom two plots of Figure 2. Finally, we compute the buffer region as defined in (21). The resultant buffer region is shown in Blue in Figure 3. If Q_j is inside the base obstacle set and Q_i is outside the buffer region, we can ensure that the intruder will have to spend a duration of at least $t^{\text{BRD}} = 10/3\,\text{s}$ to go from the boundary of the avoid region of Q_j to the boundary of the avoid region of Q_i.

For the comparison purposes, we also computed the buffer regions for $\bar{k} = 2$ and $\bar{k} = 4$. As shown in Figure 3, 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 in the same duration.

These buffer region computations along with the induced obstacle computations were similarly performed sequentially for each vehicle to obtain $\mathcal{G}(\cdot)$ in (34). This overall obstacle set was then used during their trajectory planning and the control policy $u^{\text{PP}}(\cdot)$ was computed, as defined in (37). 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 are shown in Figure 4. 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". At any given time, the vehicle density is low to ensure that the intruder cannot force more than three vehicles to apply an avoidance maneuver. This is also evident from large obstacles induced by vehicles for the lower priority vehicles (dashed circles). 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 trade-off further in section V-C.

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 particular, if a vehicle continues to apply the control policy $u^{\text{PP}}(\cdot)$ in the presence of an intruder, it might lead to a collision. In Figure 5, we plot the distance between an STP vehicle and the intruder when the vehicle applies the control policy $u^{\text{PP}}(\cdot)$ (Red line) vs when it applies

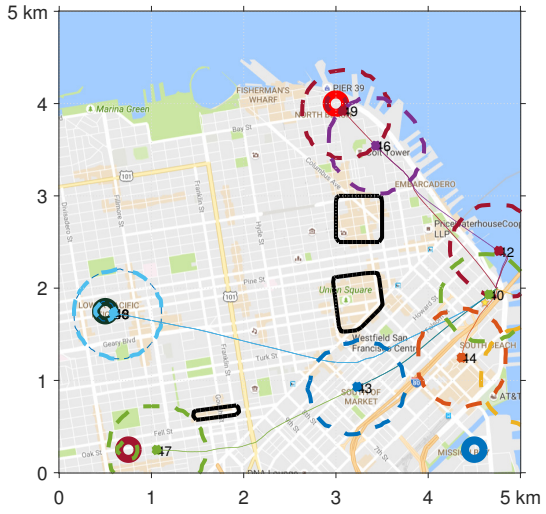

Fig. 4: Nominal trajectories and induced obstacles by different vehicles. The nominal trajectories (solid lines) are well separated from each other to ensure that the intruder cannot force more than 3 vehicles to apply an avoidance maneuver.

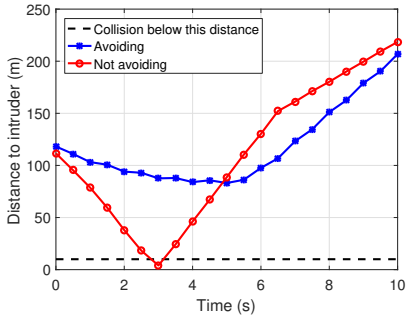

Fig. 5: The trajectory of a STP vehicle when it applies the nominal controller vs when it applies the avoidance control. The vehicle is forced to apply the avoidance maneuver in the presence of an intruder, which can cause vehicle's deviation from its nominal trajectory.

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.

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 6. 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 6a). Immediately, it travels through the buffer region between Q_1 and Q_2 and at

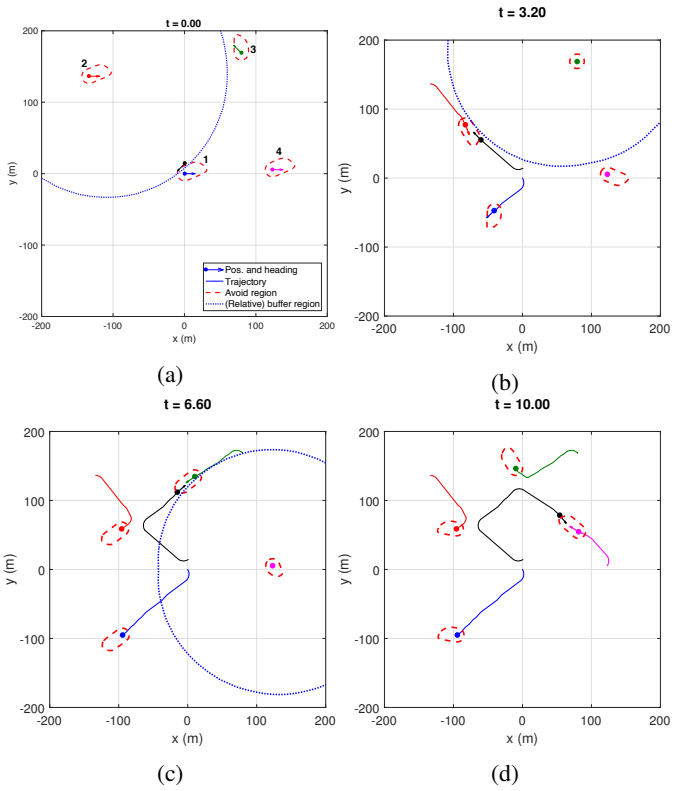

Fig. 6: Illustration of the intruder strategy to force maximum number of vehicles to apply an avoidance maneuver and hence to replan their trajectories. Q_I is able to force $\bar{k}=3$ vehicles to apply an avoidance control if the vehicles are applying the *worst control* which takes it closer to the intruder while the intruder is trying to reach its avoid region boundary.

$t=\underline{t}+t^{\text{BRD}}$ =3.33 s, reaches the boundary of the avoid region of Q_2 (Red vehicle), as shown in Figure 6a. The trajectories that Q_1 will follow while applying the avoidance control, and 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 s, and will just barely reach the boundary of the avoid region of Q_4 (Pink vehicle) at t =10 s. 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 s, $\underline{t}_2$ =3.33 s, $\underline{t}_3$ =6.67 s 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 STP 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, 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 larger 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 7. In this case, Q_I again appears at the boundary of the avoid region of Q_1 at $t = 0$, as shown in Figure 7a. 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, Q_2 now applies 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 does not happen until t =6.4 s. 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 is removed from the system at $t = t^{\text{IAT}}$ =10 s. 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 s, $\underline{t}_2$ =6.4 s, $\underline{t}_3 = \infty$ 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.

The time for planning and replanning for each vehicle is approximately 15 minutes on a MATLAB implementation on a desktop computer with a Core i7 5820K processor. With a GPU-parallelized CUDA implementation in C++ using two GeForce GTX Titan X graphics processing units, this computation time is reduced to approximately 9 seconds per vehicle. So for $\bar{k} = 3$, replanning would take less than 30 seconds. Reachability computations are highly parallelizable, and with more computational resources, replanning should be possible to do within a fraction of seconds.

C. Discussion

The simulations illustrate the effectiveness of reachability in ensuring that the STP vehicles safely reach their respective destinations even in the presence of an intruder. However, they also highlight some of the conservatism in the worst-case reachability analysis. For example, in the proposed algorithm, we assume the worst-case disturbances and intruder behavior while computing the buffer region and the induced obstacles, which results in a large separation between vehicles and hence a lower vehicle density overall, as evident from Figure 4. 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, even though the vehicle will only be applying the nominal control strategy $u^{\text{PP}}(\cdot)$, which is usually not be same as the worst-case control strategy. This worst-case analysis is essential to guarantee safety regardless of the actions of STP vehicles, the intruder, and disturbances, given no other information about the intruder's intentions and no model of disturbances except for the bounds. However, the conservatism of our results illustrates the need and the utility of acquiring more information about the intruder and disturbances, and of incorporating knowledge of the nominal strategy $u^{\text{PP}}(\cdot)$ in future work.

VI. Conclusion and Future Work

We propose an algorithm to account for an adversarial intruder in sequential trajectory 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 the 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, USA. Future work includes exploring methods that can account for multiple simultaneous intruders and reduce conservatism in the current analysis.

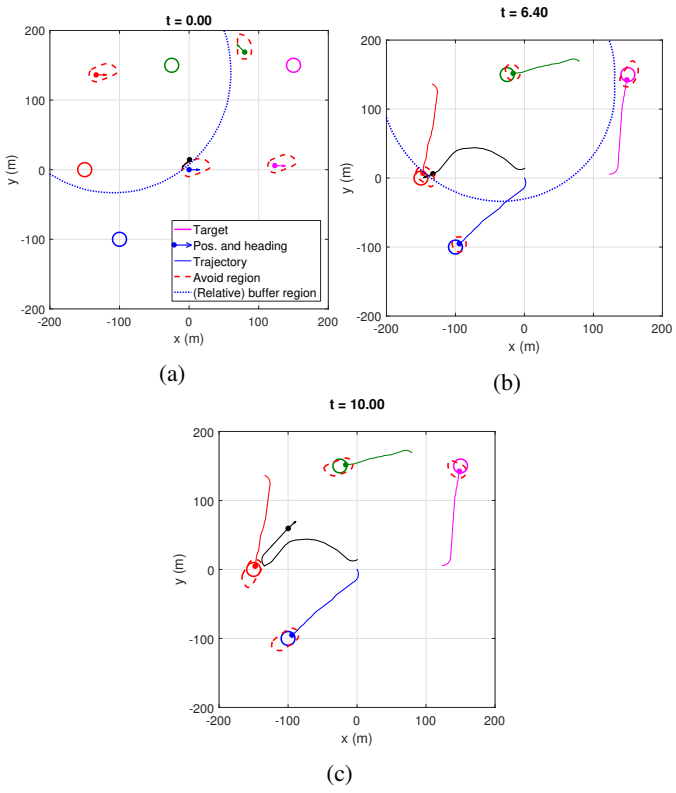

Fig. 7: Illustration of the intruder strategy to force the maximum number of vehicles to apply an avoidance maneuver and hence to replan their trajectories. Since a vehicle's nominal controller might be different from the worst case controller that is assumed while computing the buffer region, Q_I is very likely to be able to force less than $\bar{k}$ vehicles to apply an avoidance maneuver despite its best strategy.

References

[1] B. 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 *Infotech@ Aerospace Conferences*, 2010.

[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. Robinson III, M. Johnson, and J. Jung, "UAS Traffic Management (UTM) concept of operations to safely enable low altitude flight operations," in *Proc. AIAA Aviation Technol., Integration, and Operations Conf.*, 2016.
[8] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. 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 *Proc. Amer. Control Conf.*, 2005.
[10] J. Van den Berg, L. Ming, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2008.
[11] A. Wu and J. How, "Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles," *Autonomous Robots*, vol. 32, no. 3, pp. 227–242, 2012.
[12] R. Olfati-Saber and R. 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. Chuang, Y. Huang, M. D'Orsogna, and A. Bertozzi, "Multi-vehicle flocking: Scalability of cooperative control algorithms using pairwise potentials," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2007.
[14] F. Lian and R. Murray, "Real-time trajectory generation for the cooperative path planning of multi-vehicle systems," in *Proc. IEEE Conf. Decision and Control*, 2002.
[15] A. Ahmadzadeh, N. Motee, A. Jadbabaie, and G. Pappas, "Multi-vehicle path planning in dynamically changing environments," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2009.
[16] J. Bellingham, M. Tillerson, M. Alighanbari, and J. How, "Cooperative path planning for multiple UAVs in dynamic and uncertain environments," in *Proc. IEEE Conf. Decision and Control*, 2002.
[17] R. Beard and T. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *Proc. IEEE Conf. Decision and Control*, 2003.
[18] T. Schouwenaars and E. Feron, "Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees," in *Proc. AIAA Guidance, Navigation and Control Conf.*, 2004.
[19] D. Stipanovic, P. Hokayem, M. Spong, and D. Siljak, "Cooperative avoidance control for multiagent systems," *ASME J. 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 *Proc. Int. Conf. Engineering of Complex Computer Systems*, 2001.
[21] M. Althoff and J. Dolan, "Set-based computation of vehicle behaviors for the online verification of autonomous vehicles," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, 2011.
[22] Y. Lin and S. Saripalli, "Collision avoidance for UAVs using reachable sets," in *Proc. Int. Conf. Unmanned Aircraft Systems*, 2015.
[23] E. Lalish, K. Morgansen, and T. Tsukamaki, "Decentralized reactive collision avoidance for multiple unicycle-type vehicles," in *Proc. Amer. Control Conf.*, 2008.
[24] G. Hoffmann and C. Tomlin, "Decentralized cooperative collision avoidance for acceleration constrained vehicles," in *Proc. IEEE Conf. Decision and Control*, 2008.
[25] M. Chen, J. Shih, and C. Tomlin, "Multi-vehicle collision avoidance via Hamilton-Jacobi reachability and mixed integer programming," in *Proc. IEEE Conf. Decision and Control*, 2016.
[26] E. Barron, "Differential games with maximum cost," *Nonlinear analysis: Theory, methods & applications*, vol. 14, no. 11, pp. 971–989, 1990.
[27] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 947–957, 2005.
[28] O. Bokanowski, N. Forcadel, and H. Zidani, "Reachability and minimal times for state constrained nonlinear problems without any controllability assumption," *J. Control and Optimization*, vol. 48, no. 7, pp. 4292–4316, 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, 2011.
[30] K. Margellos and J. Lygeros, "HamiltonJacobi formulation for reachavoid differential games," *IEEE Trans. Autom. Control*, vol. 56, no. 8, pp. 1849–1861, 2011.
[31] J. Fisac, M. Chen, C. Tomlin, and S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proc. ACM Int. Conf. Hybrid Systems: Computation and Control*, 2015.
[32] J. Sethian, "A fast marching level set method for monotonically advancing fronts," *National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.
[33] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2006.
[34] I. 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," *Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, http://www. cs. ubc. ca/˜ mitchell/ToolboxLS/toolboxLS.pdf, Tech. Rep. TR-2004-09*, 2004.
[36] A. Bayen, I. Mitchell, M. Osihi, and C. Tomlin, "Aircraft autolander safety analysis through optimal control-based reach set computation," *AIAA J. Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 68–77, 2007.
[37] J. Ding, J. Sprinkle, S. Sastry, and C. Tomlin, "Reachability calculations for automated aerial refueling," in *Proc. IEEE Conf. Decision and Control*, 2008.
[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. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2011.
[40] M. Chen, J. Fisac, S. Sastry, and C. Tomlin, "Safe sequential path planning of multi-vehicle systems via double-obstacle Hamilton-Jacobi-Isaacs variational inequality," in *Proc. European Control Conf.*, 2015.
[41] S. Bansal, M. Chen, J. Fisac, and C. Tomlin, "Safe sequential path planning of multi-vehicle systems under presence of disturbances and imperfect information," in *Proc. Amer. Control Conf.*, 2017.
[42] M. Chen, S. Bansal, J. Fisac, and C. Tomlin, "Robust Sequential Path Planning Under Disturbances and Adversarial Intruder," *IEEE Trans. Control Syst. Technol.*, to appear.
[43] S. Herbert, M. Chen, S. Han, S. Bansal, J. Fisac, and C. Tomlin, "FaSTrack: a modular framework for fast and guaranteed safe motion planning," *Proc. IEEE Conf. Decision and Control*, 2017.
[44] 3D Robotics, "Solo specs: Just the facts," 2015. [Online]. Available: https://news.3dr.com/solo-specs-just-the-facts-14480cb55722#.w7057q926
[45] New Atlas, "Amazon Prime Air." [Online]. Available: http://newatlas.com/amazon-new-delivery-drones-us-faa-approval/36957/
[46] Wikipedia, "Beaufort scale." [Online]. Available: https://en.wikipedia.org/wiki/Beaufort_scale#Modern_scale

VII. Appendix

A. Separation and Buffer Regions - Case 2

In this section, we consider Case 2: $\underline{t}_i < \underline{t}_j, \underline{t}_i < \infty$. In this case, the intruder forces Q_i, the lower-priority vehicle, to apply an avoidance maneuver before Q_j, the higher-priority vehicle. The analysis in this case is similar to that of Case 1 (Section IV-B). However, there are a few subtle differences, which we point out wherever relevant. We start our analysis with an observation similar to Observation 1:

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

1) Separation region: Similar to Section IV-B1, we want to compute the set of all states of the intruder 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 $t^{\mathrm{RD}} := t^{\mathrm{IAT}} - t^{\mathrm{BRD}}$. This is due to the fact that our design of the buffer region in Section VII-A2 ensures that it takes the intruder 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:

$$\mathcal{S}_j(\underline{t}_j) = \mathcal{M}_j(\underline{t}_j) + \partial\mathcal{V}_j^{\mathrm{A}}(0, t^{\mathrm{RD}}). \tag{45}$$

2) Buffer Region: The idea behind the design of buffer region is same as that in Case 1: 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 STP vehicle to the boundary of the avoid region of some other STP 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 $\mathcal{V}_j^{\text{B}}(0, t^{\text{BRD}})$:

$$\begin{aligned}\mathcal{V}_j^{\text{B}}(0,t^{\text{BRD}}) =&\{y : \exists u_j(\cdot) \in \mathbb{U}_j, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_j(\cdot) \in \mathbb{D}_j,\\ &\exists d_I(\cdot) \in \mathbb{D}_I, x_{I,j}(\cdot) \text{ satisfies (12)},\\ &\exists s \in [0, t^{\text{BRD}}], x_{I,j}(s) \in \mathcal{V}_j^{\text{A}}(t^{\text{BRD}}, t^{\text{IAT}}),\\ &x_{I,j}(t) = y\},\end{aligned} \tag{46}$$

where

$$H_j^{\text{B}}(x_{I,j}, \lambda) = \min_{\substack{u_j \in \mathcal{U}_j, u_I \in \mathcal{U}_I,\\ d_j \in \mathcal{D}_j, d_I \in \mathcal{D}_I}} \lambda \cdot f_r(x_{I,j}, u_j, u_I, d_j, d_I) \tag{47}$$

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 $x_{I,i}(\underline{t}_i) \in \partial\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$ implies that $\underline{t}_j - \underline{t}_i \geq t^{\text{BRD}}$. Mathematically, if we define the set,

$$\mathcal{B}_{ji}(\underline{t}) = \mathcal{M}_j(\underline{t}) + \mathcal{V}_j^{\text{B}}(0, t^{\text{BRD}}) + \left(-\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})\right), \tag{48}$$

then $(\underline{t}_j - \underline{t}_i) \geq t^{\text{BRD}}$ as long as $x_i(\underline{t}) \in (\mathcal{B}_{ji}(\underline{t}))^C$. Thus, if $x_i(\underline{t}) \in (\mathcal{B}_{ji}(\underline{t}))^C$, then the separation requirement (19) is satisfied for Case 2. 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.

Note that we use $-\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$ instead of $\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$ in (48) because $\mathcal{V}_i^{\text{A}}(0, t^{\text{IAT}})$ is computed using the relative state $x_{I,i}$ and we are interested in finding the "unsafe" states for Q_i when the intruder is outside $\tilde{\mathcal{B}}_{ji}(t)$.

3) Obstacle Computation: We now compute the set of states that Q_i needs to avoid in order to avoid entering in the danger zone of Q_j eventually. We consider the following two mutually exclusive and exhaustive cases:

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

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

- Case A: 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 particular case, the set of possible states of Q_j at time t is given by $\mathcal{M}_j(t)$. To compute ${}_2^A\mathcal{O}_i^j(\cdot)$, we begin with the following observation:

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

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

$$\begin{aligned}\mathcal{K}^{\text{A2}}(\tau) =&\tilde{\mathcal{M}}_j(\tau),\\ \tilde{\mathcal{M}}_j(s) =&\{x_j : \exists (y, h) \in \mathcal{M}_j(s), \|p_j - y\|_2 \leq R_c\}.\end{aligned} \tag{49}$$

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

$$\begin{aligned}\mathcal{V}_i^{\text{A2}}(t, t + t^{\text{IAT}}) =&\{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i,\\ &x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y,\\ &\exists s \in [t, t + t^{\text{IAT}}], x_i(s) \in \mathcal{K}^{\text{A2}}(s)\}.\end{aligned} \tag{50}$$

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

$$H_i^{\text{A2}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i, d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \tag{51}$$

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

$${}_2^A\mathcal{O}_i^j(t) = \mathcal{V}_i^{\text{A2}}(t, t + t^{\text{IAT}}). \tag{52}$$

- Case B: In this case, the intruder first affects Q_i and then Q_j. Recall that 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 $\underline{t}_i = \underline{t}$, from the perspective of Q_i, Q_j can apply any control during the duration $[\underline{t}_j, \underline{t}_i + t^{\text{IAT}}]$ and hence can be anywhere in the set $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau)$ at $\tau \in [\underline{t}_j, \underline{t}_i + t^{\text{IAT}}]$, where $\mathcal{W}_j^{\mathcal{O}}$ denotes the FRS of base obstacle $\mathcal{M}_j(\underline{t}_j)$ computed forward for a duration of $(\underline{t}_i + t^{\text{IAT}} - \underline{t}_j)$. Q_i thus needs to make sure that it avoids all states at time t that can reach $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau)$, regardless of the control applied by Q_i during $[t, \tau]$. We now make the following key observation:

  Observation 7: Observation 3 implies that $\mathcal{W}_j^{\mathcal{O}}(\tau_2, \tau) \subseteq \mathcal{W}_j^{\mathcal{O}}(\tau_1, \tau)$ if $\tau > \tau_2 > \tau_1$. Therefore, the biggest obstacle, $\mathcal{W}_j^{\mathcal{O}}(\underline{t}_j, \tau)$, 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 biggest 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 $\underline{t}_j - \underline{t}_i \geq t^{\text{BRD}}$. Hence, the biggest obstacle is induced by Q_j when $\underline{t}_j = \underline{t}_i + t^{\text{BRD}}$.

Intuitively, Observation 7 implies that the biggest obstacle is induced by Q_j when intruder forces Q_i to apply the avoidance maneuver and *immediately* begins traveling 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 $\mathcal{K}^{\text{B2}}(\tau)$ at time $\tau > t$, where

$$\mathcal{K}^{\text{B2}}(\tau) = \bigcup_{\underline{t}_i \in [t-t^{\text{IAT}}, t], \tau \leq \underline{t}_i + t^{\text{IAT}}} \mathcal{W}_j^{\mathcal{O}}(\underline{t}_i + t^{\text{BRD}}, \tau), \tau > t, \tag{53}$$

where we have substituted $\underline{t}_j = \underline{t}_i + t^{\text{BRD}}$. In (53), $\underline{t} = \underline{t}_i \in [t - t^{\text{IAT}}, t]$ due to Observation 2 and $\tau \leq \underline{t}_i + t^{\text{IAT}}$ because the intruder can appear for a maximum duration of t^{IAT}. Equation (53) can be equivalently written as:

$$\begin{aligned}\mathcal{K}^{\text{B2}}(\tau) =& \bigcup_{\underline{t}_i \in [\tau - t^{\text{IAT}}, t]} \mathcal{W}_j^{\mathcal{O}}(\underline{t}_i + t^{\text{BRD}}, \tau), t < \tau \leq t + t^{\text{IAT}} \\ \mathcal{K}^{\text{B2}}(\tau) =& \mathcal{W}_j^{\mathcal{O}}(\tau - t^{\text{IAT}} + t^{\text{BRD}}, \tau), t < \tau \leq t + t^{\text{IAT}},\end{aligned} \tag{54}$$

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:

$$\begin{aligned}\mathcal{V}_i^{\text{B2}}(t, t + t^{\text{IAT}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y, \\ & \exists s \in [t + t^{\text{BRD}}, t + t^{\text{IAT}}], x_i(s) \in \tilde{\mathcal{K}}^{\text{B2}}(s)\}, \\ \tilde{\mathcal{K}}^{\text{B2}}(s) =& \{x_i : \exists (y, h) \in \mathcal{K}^{\text{B2}}(s), \|p_i - y\|_2 \leq R_c\}.\end{aligned} \tag{55}$$

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

$$H_i^{\text{B2}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i, d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i). \tag{56}$$

Finally, the induced obstacle in this case is given as

$${}_2^B\mathcal{O}_i^j(t) = \mathcal{V}_i^{\text{B2}}(t, t + t^{\text{IAT}}). \tag{57}$$