Robust Sequential Trajectory Planning Under Disturbances and Adversarial Intruder

Mo Chen, Somil Bansal, Jaime F. Fisac, Claire J. Tomlin

***Abstract*—Provably safe and scalable multi-vehicle trajectory planning is an important and urgent problem due to the expected increase of automation in civilian airspace in the near future. Although this problem has been studied in the past, there has not been a method that guarantees both goal satisfaction and safety for vehicles with general nonlinear dynamics while taking into account disturbances and potential adversarial agents, to the best of our knowledge. Hamilton-Jacobi (HJ) reachability is the ideal tool for guaranteeing goal satisfaction and safety under such scenarios, and has been successfully applied to many small-scale problems. However, a direct application of HJ reachability in most cases becomes intractable when there are more than two vehicles due to the exponentially scaling computational complexity with respect to system dimension. In this paper, we take advantage of the guarantees HJ reachability provides, and eliminate the computation burden by assigning a strict priority ordering to the vehicles under consideration. Under this sequential trajectory planning (STP) scheme, vehicles reserve "space-time" portions in the airspace, and the space-time portions guarantee dynamic feasibility, collision avoidance, and optimality of the trajectories given the priority ordering. With a computation complexity that scales quadratically when accounting for both disturbances and an intruder, and *linearly* when accounting for only disturbances, STP can tractably solve the multi-vehicle trajectory planning problem for vehicles with general nonlinear dynamics in a practical setting. We demonstrate our theory in representative simulations.**

I. Introduction

Recently, there has been an immense surge of interest in the use of unmanned aerial systems (UASs) in urban environments. UASs have great potential in civil applications such as package delivery, aerial surveillance, disaster response, among many others (Removed 5 references). Unlike previous uses of UASs for military purposes, civil applications will involve unmanned aerial vehicles (UAVs) flying in urban environments, potentially in close proximity of humans, other UAVs, and other important assets. As a result, government agencies such as the Federal Aviation Administration (FAA) and National Aeronautics and Space Administration (NASA) of the United States are urgently trying to develop new scalable ways to organize an airspace in which potentially thousands of UAVs can fly simultaneously in the same region [1], [2].

One essential problem that needs to be addressed is the safe multi-vehicle trajectory planning problem: how a group of vehicles in the same vicinity can reach their destinations while avoiding situations which are considered dangerous, such as collisions. In many previous studies that address this problem, specific control strategies for the vehicles are assumed, and approaches such as those involving induced velocity obstacles [3]–[5] and involving virtual potential fields to maintain collision [6], [7] have been used. Other analyses of multi-vehicle systems include methods for real-time trajectory generation [8], for path planning for vehicles with linear dynamics in the presence of obstacles with known motion [9], and for cooperative path planning via waypoints which do not account for vehicle dynamics [10]. Other related work include those which consider only the collision avoidance problem without path planning. These results include those that assume the system has a linear model [11]–[13], rely on a linearization of the system model [14], [15], assume a simple positional state space [16], and many others [17]–[19].

However, the capability to flexibly plan provably safe and dynamically feasible trajectories without making strong assumptions on the vehicles' dynamics and other vehicles' motion is essential for dense groups of UAVs to safely fly in each other's vicinity. In addition, in a practical setting, any trajectory or path planning scheme that also addresses collision avoidance must guarantee both goal satisfaction and safety of UAVs despite disturbances such as weather effects and communication faults [2]. Furthermore, unexpected scenarios such as UAV malfunctions or even UAVs with malicious intent need to be accounted for.

The problem of trajectory planning and collision avoidance under disturbances in safety-critical systems has been studied using Hamilton-Jacobi (HJ) reachability analysis, which provides guarantees on goal satisfaction and safety of optimal system trajectories [20]–[25]. Reachability-based methods are particular suitable in the context of UAVs because of the hard guarantees that are provided. In reachability analysis, one computes the reachable set, defined as the set of states from which the system can be driven to a target set. Many numerical tools are available for computing various definitions of reachable sets [26]–[29], and reachability analysis has been successfully used in applications involving systems with no more than two vehicles, such as pairwise collision avoidance [21], automated in-flight refueling [30], and many others [31], [32].

One of the main challenges of managing the next generation of airspace is the density of vehicles that needs to be accommodated [2]. Such a large-scale system has a high-dimensional joint state space, making a direct application of dynamic programming-based approaches such as reachability analysis intractable. In particular, reachable set computations involve solving a HJ partial differential equation (PDE) or variational inequality (VI) on a grid representing a discretization of

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

All authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. {mochen72, somil, jfisac, tomlin}@eecs.berkeley.edu

the state space, causing computational complexity to scale *exponentially* with system dimension.

A. Contributions and Outline

In this paper, we propose the sequential trajectory planning (STP) method to tackle the multi-vehicle trajectory planning problem. Our approach is similar to the approaches of [33], [34], but provides hard guarantees on both the goal satisfaction and safety of all vehicles even in the presence of disturbances and a single intruder vehicle that could potentially be adversarial. In addition, our method scales only *linearly* with the number of vehicles when there is no intruder, and *quadratically* with the number of vehicles when there is a single intruder. On a high level, the STP method assigns a strict priority ordering to the vehicles under consideration. Higher-priority vehicles plan their trajectories without taking into account the lower-priority vehicles. Lower-priority vehicles treat higher-priority vehicles as moving obstacles. Under this assumption, time-varying formulations of reachability [23], [25] 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 for the multi-vehicle trajectory planning problem at the cost of a mild structural assumption.

In a sense, the STP method reserves a portion of "space-time" in the airspace for each vehicle. The reserved space-time portion is recorded so that lower-priority vehicles can take it into account. Besides planning around the reserved space-time portions of higher-priority vehicles, no other communication between the vehicles is needed at execution time, even when disturbances and an intruder are present. Besides reducing computational complexity, a priority-based approach makes practical sense as priorities can be assigned in a first-come-first-serve basis by, for example, an air navigation service provider [2], based on flight requests from various companies and agencies sharing the airspace.

Almost all computation is done *offline* to produce a value function, corresponding to an appropriate optimal control problem or differential game, for each vehicle. The gradient of the value function can be stored in a look-up table, which is used *online* to synthesize the optimal controller. Controller synthesis amounts to evaluating an analytic expression, is the only online computation, and therefore can be done in real-time.

In the absence of disturbances and intruders, and assuming each vehicle has perfect information about other vehicles' positions, each vehicle may plan and commit to an exact trajectory, with the reserved space-time being the collision set around the trajectory at every point in time. This basic concept of STP is formally presented in Section IV.

When the vehicles are affected by disturbances, exact trajectories cannot be known *a priori*, and thus the basic STP algorithm cannot be directly applied. Fortunately, reachability analysis allows us to determine, at no additional computation cost, all possible states of each vehicle over time under the worst-case disturbance, given a control strategy. In addition, we can also determine suitable portions of space-time for each vehicle depending on the available information about the control strategies of higher-priority vehicles. STP under disturbances and three different assumptions on the information available about the control strategy of other vehicles is formally presented in Section V.

In scenarios where there could potentially be single, possibly adversarial intruder in the airspace, each vehicle needs extra space around other vehicles in order to be able to perform avoidance maneuvers. Assuming the intruder may be present for some maximum duration, we use use reachability analysis to determine precisely the amount of space-time needed for each vehicle to be able to avoid the intruder under the presence of disturbances, making our proposed method sufficiently robust to most practical scenarios. STP in the presence of a single intruder is formally presented in Section VI.

II. Problem Formulation

Consider N vehicles which participate in the STP process and denote these vehicles as the *STP vehicles* $Q_i, i = 1, \dots, N$. We assume their dynamics are given by

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

where $x_i \in \mathbb{R}^{n_i}$ represents the state of vehicle Q_i, $u_i \in \mathcal{U}_i$ the control of Q_i, and $d_i \in \mathcal{D}_i$ the disturbance experienced by Q_i. For convenience, 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 assume that the control functions $u_i(\cdot), d_i(\cdot)$ are drawn from the set of measurable functions[1]. For convenience, 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.

We further assume that the flow field $f_i : \mathbb{R}^{n_i} \times \mathcal{U}_i \times \mathcal{D}_i \to \mathbb{R}^{n_i}$ is uniformly continuous, bounded, and Lipschitz continuous in x_i for fixed u_i and d_i. With this assumption, given $u_i(\cdot) \in \mathbb{U}_i, d_i(\cdot) \in \mathbb{D}_i$, there exists a unique trajectory solving (1) [35].

In addition, we assume that the disturbances $d_i(\cdot)$ are drawn from the set of non-anticipative strategies [21] Γ, defined as follows:

$$\begin{aligned} \Gamma := \{&\mathcal{N} : \mathbb{U}_i \to \mathbb{D}_i : u_i(r) = \hat{u}_i(r) \text{ a. e. } r \in [t, s] \\ &\Rightarrow \mathcal{N}[u_i](r) = \mathcal{N}[\hat{u}_i](r) \text{ a. e. } r \in [t, s]\} \end{aligned} \tag{2}$$

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. In some situations, we may find that it is infeasible for Q_i to get to $\mathcal{L}_i$ at or before t_i^{STA}. Whenever unsure, we may first determine the earliest feasible t_i^{STA} as described in Section VI.

On its way to $\mathcal{L}_i$, Q_i must avoid a set of static obstacles $\mathcal{O}_i^{\text{static}} \subset \mathbb{R}^{n_i}$. The interpretation of $\mathcal{O}_i^{\text{static}}$ could be a tall

[1] A function $f : X \to Y$ between two measurable spaces (X, Σ_X) and (Y, Σ_Y) is said to be measurable if the preimage of a measurable set in Y is a measurable set in X, that is: $\forall V \in \Sigma_Y, f^{-1}(V) \in \Sigma_X$, with Σ_X, Σ_Y σ-algebras on X,Y.

building, a region around an airport, or any set of states that are forbidden for each STP vehicle.

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

$$\mathcal{Z}_{ij} = \{(x_i, x_j) : \|p_i - p_j\|_2 \leq R_c\} \tag{3}$$

whose interpretation is that Q_i and Q_j are considered to be in an unsafe configuration when they are within a distance of R_c of each other. For concreteness, we will call $\mathcal{Z}_{ij}$ the collision set, and if $(x_i, x_j) \in \mathcal{Z}_{ij}$, then Q_i and Q_j are said to have collided. Note that, for practical system, there might exists an inevitable danger zone given the position-based danger zone $\mathcal{Z}_{ij}$ (for example the states from which wind can push an aerial vehicle in $\mathcal{Z}_{ij}$ irrespective of control applied by the vehicle). However, the reachability analysis presented in this paper can successfully avoid such inevitable danger zones. Finally, if the vehicles are not particles (as often is the case), the danger zone $\mathcal{Z}_{ij}$ can be defined to include the effective sizes of the vehicles.

Given the set of STP vehicles, their targets $\mathcal{L}_i$, the static obstacles $\mathcal{O}_i^{\text{static}}$, and the vehicles' danger zones with respect to each other $\mathcal{Z}_{ij}$, we would like, for each vehicle Q_i, to synthesize a controller which guarantees that Q_i reaches its target $\mathcal{L}_i$ at or before the scheduled time of arrival t_i^{STA}, while avoiding the static obstacles $\mathcal{O}_i^{\text{static}}$ as well as the danger zones with respect to all other vehicles $\mathcal{Z}_{ij}, j \neq i$. In addition, we would like to obtain the latest departure time t_i^{LDT} such that Q_i can still arrive at $\mathcal{L}_i$ on time. Note that as long as it is feasible a vehicle to reach its target in the *absence* of all other vehicles, producing a safe and timely trajectory is *always* feasible using our proposed algorithms, since the latest departure time t_i^{LDT} is obtained. Indeed, if the environment is expected to be crowded, a vehicle can simply depart early enough (and potentially arrive very early) to "avoid traffic". In practice, if departing at or before the latest departure time t_i^{LDT} is not possible, then arriving on time is infeasible.

In general, the above optimal trajectory planning problem must be solved in the joint space of all N STP vehicles. However, due to the high joint dimensionality, a direct dynamic programming-based solution is intractable. Therefore, we propose 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 $Q_j, j < i$; each higher-priority vehicle Q_j induces a set of time-varying obstacles $\mathcal{O}_i^j(t)$, which represents the possible states of Q_i such that a collision between Q_i and Q_j could occur.

It is straight-forward to see that if each vehicle Q_i is able to plan a trajectory that takes it to $\mathcal{L}_i$ while avoiding the static obstacles $\mathcal{O}_i^{\text{static}}$ and the danger zones of *higher-priority vehicles* $Q_j, j < i$, then the set of STP vehicles $Q_i, i = 1, \ldots, N$ would all be able to reach their targets safely. With the STP scheme, the additional structure provided by the vehicle priorities allows us to reduce the complexity of the joint trajectory planning problem. As we will see, 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 in the presence of disturbances, as opposed to exponentially with a direct application of dynamic programming approaches. In the presence of a single intruder, the computation complexity scaling becomes quadratic. As mentioned earlier, priorities of vehicles may be assigned in a first-come-first-serve basis through, for example, an air navigation service provider [2] which manages a region of the low altitude airspace.

In the following sections, we will explore STP under different assumptions. We begin with the basic STP algorithm in which disturbances are ignored and perfect information of vehicles' positions is assumed. This simplification allows us to clearly establish the basic STP algorithm. Next, we show how the basic STP approach can be made robust to disturbances as well as an imperfect knowledge of other vehicles' positions. Finally, we further robustify the STP approach by considering how the set of STP vehicles may respond to the presence of an intruder vehicle which may be adversarial. All of our methods use time-varying reachability analysis to provide goal satisfaction and safety guarantees.

III. Time-Varying Reachability Background

For the safe trajectory planning, we will use reachability theory. In particular, we will be using reachability analysis to compute either a backward reachable set (BRS) $\mathcal{V}$, a forward reachable set (FRS) $\mathcal{W}$, or a sequence of BRSs and FRSs, given some target set $\mathcal{L}$, time-varying obstacle $\mathcal{G}(t)$, and the Hamiltonian function H which captures the system dynamics as well as the roles of the control and disturbance. Intuitively, BRS is the set of states such that the system trajectories that start from this set can reach some given target set without colliding with obstacles, despite the worst-case disturbance. Similarly, FRS is the set of states that can be reached by a system starting from a target set. The BRS $\mathcal{V}$ in a time interval $[t, t_f]$ or FRS $\mathcal{W}$ in a time interval $[t_0, t]$ will be denoted by

$$\begin{aligned} &\mathcal{V}(t, t_f) \quad \text{(backward reachable set)} \\ &\mathcal{W}(t_0, t) \quad \text{(forward reachable set)} \end{aligned} \tag{4}$$

The computation of BRS (or FRS) can be formulated as a differential game between the control and disturbance (or as an optimal control problem when no disturbance is not involved), which can be solved using the principle of dynamic programming. Ultimately, this results in solving a HJ PDE. For further details on this formulation, we refer the interested readers to [36]. Traditionally, only static obstacles had been considered in the reachability formulation. However, in the STP scheme, a lower-priority vehicle must avoid a set of moving obstacles (i.e., time-varying obstacles) on its way to

the target. Several formulations of reachability are able to perform optimal trajectory planning with hard guarantees on safety and performance under disturbances in such a scenario [23], [25]. For our application in STP, we utilize the time-varying formulation in [25], which accounts for the time-varying nature of systems without requiring augmentation of the state space with the time variable. In the formulation in [25], a BRS is computed by solving the following *final value* double-obstacle HJ VI:

$$\begin{aligned}\max\Big\{\min\{D_tV(t,x)+H(t,x,\nabla V(t,x)),l(x)-V(t,x)\},\\ -g(t,x)-V(t,x)\Big\}=0,\quad t\le t_f\\ V(t_f,x)=\max\{l(x),-g(t_f,x)\}\end{aligned} \tag{5}$$

Here, $D_tV(t,x)$ and $\nabla V(t,x)$ denote the time and space derivative of the value function V respectively. In a similar fashion, the FRS is computed by solving the following *initial value* HJ PDE:

$$\begin{aligned}D_tW(t,x)+H(t,x,\nabla W(t,x))=0,\quad t\ge t_0\\ W(t_0,x)=\max\{l(x),-g(t_0,x)\}\end{aligned} \tag{6}$$

In both (5) and (6), 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

$$\begin{aligned}\mathcal{V}(t,t_f)=\{x:V(t,x)\le 0\}\\ \mathcal{W}(t_0,t)=\{x:W(t,x)\le 0\}\end{aligned} \tag{7}$$

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 (5). Also, note that unlike in (5), there is no inner minimization in (6). As we will see later, we will be using the BRS to determine all states that can reach some target set *within the time horizon* $[t,t_f]$, whereas we will be using the FRS to determine where a vehicle could be *at some particular time* t. In addition, (6) has no outer maximum, since the FRSs that we will compute will not involve any obstacles.

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

IV. STP Without Disturbances and With Perfect Information

In this section, we introduce the basic STP algorithm assuming that there is no disturbance affecting the vehicles, and that each vehicle knows the exact position of higher-priority vehicles. Although in practice, such assumptions do not hold, the description of the basic STP algorithm will introduce the notation needed for describing the subsequent, more realistic versions of STP. We also show simulation results for the basic STP algorithm. The majority of the content in this section is taken from [37].

A. Theory

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

$$\begin{aligned}\dot{x}_i=f_i(x_i,u_i), t\le t_i^{\text{STA}}\\ u_i\in\mathcal{U}_i,\qquad i=1\dots,N\end{aligned} \tag{8}$$

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

From the perspective of Q_i, each of the higher-priority vehicles $Q_j, j<i$ induces a time-varying obstacle denoted $\mathcal{O}_i^j(t)$ that Q_i needs to avoid[2]. Therefore, each vehicle Q_i must plan its trajectory to $\mathcal{L}_i$ while avoiding the union of all the induced obstacles as well as the static obstacles. Let $\mathcal{G}_i(t)$ be the union of all the obstacles that Q_i must avoid on its way to $\mathcal{L}_i$:

$$\mathcal{G}_i(t)=\mathcal{O}_i^{\text{static}}\cup\bigcup_{j=1}^{i-1}\mathcal{O}_i^j(t) \tag{9}$$

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

$$\mathcal{O}_i^j(t)=\{x_i:\|p_i-p_j(t)\|_2\le R_c\} \tag{10}$$

Each higher priority vehicle Q_j plans its trajectory while ignoring Q_i. Since trajectory planning is done sequentially in descending order or 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:

$$\begin{aligned}\mathcal{V}_i^{\text{basic}}(t,t_i^{\text{STA}})=\{y:\exists u_i(\cdot)\in\mathbb{U}_i, x_i(\cdot)\text{ satisfies (8)},\\ \forall s\in[t,t_i^{\text{STA}}], x_i(s)\notin\mathcal{G}_i(s),\\ \exists s\in[t,t_i^{\text{STA}}], x_i(s)\in\mathcal{L}_i, x_i(t)=y\}\end{aligned} \tag{11}$$

[2]Note that the index k in $\mathcal{O}_k^i$ denotes vehicles with lower priority than Q_i, and the index j in $\mathcal{O}_i^j(t)$ denotes vehicles with higher priority than Q_i.

The BRS $\mathcal{V}(t, t_i^{\text{STA}})$ can be obtained by solving (5) with $\mathcal{L} = \mathcal{L}_i$, $\mathcal{G}(t) = \mathcal{G}_i(t)$, and the Hamiltonian

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

Note that $\mathcal{V}(t, t_i^{\text{STA}})$, by definition, does not contain any states from which it is inevitable to avoid the danger zone $\mathcal{Z}_{ij}$ (and $\mathcal{G}_i$ in general). Given $\mathcal{V}(t, t_i^{\text{STA}})$, the optimal control for reaching $\mathcal{L}_i$ while avoiding $\mathcal{G}_i(t)$ is then given by

$$u_i^{\text{basic}}(t, x_i) = \arg\min_{u_i \in \mathcal{U}_i} \lambda \cdot f_i(x_i, u_i) \tag{13}$$

from which the trajectory $x_i(\cdot)$ can be computed by integrating the system dynamics, which in this case are given by (8). In addition, the latest departure time t_i^{LDT} can be obtained from the BRS $\mathcal{V}(t, t_i^{\text{STA}})$ as $t_i^{\text{LDT}} = \arg\sup_t \{x_i^0 \in \mathcal{V}(t, t_i^{\text{STA}})\}$. In summary, the basic STP algorithm is given as follows:

Algorithm 1: **Basic STP algorithm**: Suppose we are given initial conditions x_i^0, vehicle dynamics (8), target sets $\mathcal{L}_i$, and static obstacles $\mathcal{O}_i^{\text{static}}, i = 1 \ldots, N$. For each i in ascending order starting from $i = 1$ (which corresponds to descending order of priority),

1) determine the total obstacle set $\mathcal{G}_i(t)$, given in (9). In the case $i = 1$, $\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}}\ \forall t$;
2) compute the BRS $\mathcal{V}_i^{\text{basic}}(t, t_i^{\text{STA}})$ defined in (11). The latest departure time t_i^{LDT} is then given by $\arg\sup_t \{x_i^0 \in \mathcal{V}_i^{\text{basic}}(t, t_i^{\text{STA}})\}$;
3) determine the trajectory $x_i(\cdot)$ using vehicle dynamics (8), with the optimal control $u_i^{\text{basic}}(\cdot)$ given by (13);
4) given $x_i(\cdot)$, compute the induced obstacles $\mathcal{O}_k^i(t)$ for each $k > i$. In the absence of disturbances, $\mathcal{O}_k^i(t)$ is given by (10).

Note that Step 1, which determines the total obstacle set, can be updated in a recursive manner by adding a new set of induced obstacles for each next vehicle: $\mathcal{G}_{i+1}(t) = \mathcal{G}_i(t) \cup \mathcal{O}_{i+1}^i(t)$. In addition, in implementation, Step 4 can be simplified by storing $\mathcal{G}_i(t)$ as a look-up table with the maximum dimensionality across all vehicle state spaces. When a vehicle plans its trajectory, irrelevant dimensions of $\mathcal{G}_i(t)$ can be ignored. This observation keeps the computational complexity of our algorithm linear with respect to the number of vehicles.

As previously mentioned, the basic STP algorithm, as well as all subsequent variants of STP algorithms, will *always* return a feasible trajectory that arrives at the target on time, as long as a feasible trajectory exists in the *absence* of other vehicles. This is because a vehicle can simply depart early enough to avoid being blocked by higher-priority vehicles. In fact, the latest departure time t_i^{LDT} quantifies exactly when each vehicle needs to depart to arrive on time.

B. Numerical Simulations

We now illustrate the basic STP algorithm using a four-vehicle example. In this example, we will use the following dynamics for each vehicle:

$$\begin{aligned} \dot{p}_{x,i} &= v_i \cos\theta_i \\ \dot{p}_{y,i} &= v_i \sin\theta_i \\ \dot{\theta}_i &= \omega_i \\ |\omega_i| &\le \bar{\omega} \end{aligned} \tag{14}$$

where $x_i = (p_{x,i}, p_{y,i}, \theta_i)$ is the state of vehicle Q_i, $p_i = (p_{x,i}, p_{y,i})$ is the position, θ_i is the heading, v_i is the speed, and ω_i is the turn rate. In this example, we assume that the vehicles have constant speed $v_i = 1\ \forall i$, and that the control of each vehicle Q_i is given by $u_i = \omega_i$ with $|\omega_i| \le \bar{\omega} = 1\ \forall i$. We have chosen these dynamics for clarity of illustration; the STP algorithm can handle more general systems of the form in which the vehicles have different control bounds and dynamics.

For this example, the target sets $\mathcal{L}_i$ of the vehicles are circles of radius r in the position space; 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

$$\mathcal{L}_i = \{x_i : \|p_i - c_i\|_2 \le r\} \tag{15}$$

where c_i are centers of the target circles. For the simulation of the basic STP algorithm, we used $r = 0.1$. The vehicles have target centers c_i, initial conditions x_i^0, and scheduled times of arrivals t_i^{STA} as follows:

$$\begin{aligned} c_1 &= (0.7, 0.2), & x_1^0 &= (-0.5, 0, 0), & t_1^{\text{STA}} &= 0 \\ c_2 &= (-0.7, 0.2), & x_2^0 &= (0.5, 0, \pi), & t_2^{\text{STA}} &= 0.2 \\ c_3 &= (0.7, -0.7), & x_3^0 &= (-0.6, 0.6, 7\pi/4)\,, & t_3^{\text{STA}} &= 0.4 \\ c_4 &= (-0.7, -0.7), & x_4^0 &= (0.6, 0.6, 5\pi/4)\,, & t_4^{\text{STA}} &= 0.6 \end{aligned} \tag{16}$$

The setup for this example is shown in Fig. 1, which also shows the static obstacles as the black rectangles around the center of the domain.

The joint state space of this four-vehicle system is twelve-dimensional (12D), making the joint trajectory planning and collision avoidance problem intractable for direct analysis using HJ reachability. Therefore, we apply the STP algorithm described in Algorithm 1 and repeatedly solve the double-obstacle HJ VI in (5) to obtain the optimal control for each vehicle to reach its target while avoiding higher-priority vehicles. In addition, due to the flexibility of the HJ VI with respect to time-varying systems, the different scheduled times of arrival t_i^{STA} can be trivially incorporated.

Fig. 2, 3, and 4 show the simulation results. Since the state space of each vehicle is 3D, each of the BRSs $\mathcal{V}_i^{\text{basic}}(t, t_i^{\text{STA}})$ is also 3D. To visualize the results, we slice the BRSs at the initial heading angles θ_i^0. Fig. 2 shows the 2D BRS slices for each vehicle at its latest departure times $t_1^{\text{LDT}} = -1.12, t_2^{\text{LDT}} = -0.94, t_3^{\text{LDT}} = -1.48, t_4^{\text{LDT}} = -1.44$ determined from our method. The obstacles in the domain $\mathcal{O}_i^{\text{static}}$ and the obstacles induced by higher-priority vehicles $\mathcal{O}_i^j(t)$ inhibit the evolution of the BRSs, carving out thin "channels" that separate the BRSs into different "islands". One can see how these "channels" and "islands" form by examining the time evolution of the BRS, shown in Fig. 3 for vehicle Q_3.

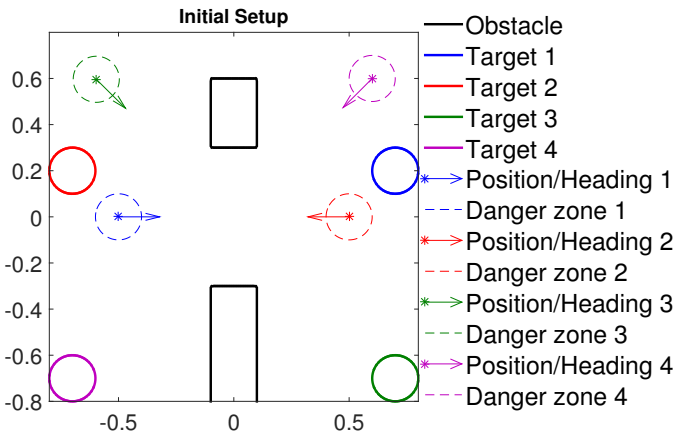

Fig. 1: Initial configuration of the four-vehicle example.

Finally, Fig. 4 shows the resulting trajectories of the four vehicles. Most interestingly, the subplot labeled $t = -0.55$ shows all four vehicles in close proximity without collision: each vehicle is outside of the danger zone of all other vehicles (although the danger zones may overlap). This close proximity is an indication of the optimality of the basic STP algorithm given the assigned priority ordering. Since no disturbances are present, getting as close to other vehicles' danger zones as possible without entering the danger zones intuitively results in short transit times.

The actual arrival times of vehicles $Q_i, i = 1, 2, 3, 4$ are $0, 0.19, 0.34, 0.31$, respectively. It is interesting to note that for some vehicles, the actual arrival times are earlier than the scheduled times of arrivals t_i^{STA}. This is because in order to arrive at the target by t_i^{STA}, these vehicles must depart early enough to avoid major delays resulting from the induced obstacles of other vehicles; these delays would have led to a late arrival if vehicle Q_i departed after t_i^{LDT}.

Computations were done on a desktop computer with a Core i7 5820K processor and two GeForce GTX Titan X graphics processing units. The average computation time per vehicle is approximately 1 second using CUDA and GPU parallelization. Note that for the simulations in this and subsequent sections, almost all of the computation is done offline. Only a look-up table query to obtain the gradient of the value function corresponding to a BRS, and evaluation of the optimal control in, for example (13), is needed online. For control affine systems, controller synthesis amounts to evaluation of an analytic expression.

V. STP With Disturbances and Incomplete Information

Disturbances and incomplete information significantly complicate the STP scheme. The main difference is that the vehicle dynamics satisfy (1) as opposed to (8). Committing to exact trajectories is therefore no longer possible, since the disturbance $d_i(\cdot)$ is *a priori* unknown. Thus, the induced obstacles $\mathcal{O}_i^j(t)$ are no longer just the danger zones centered around positions.

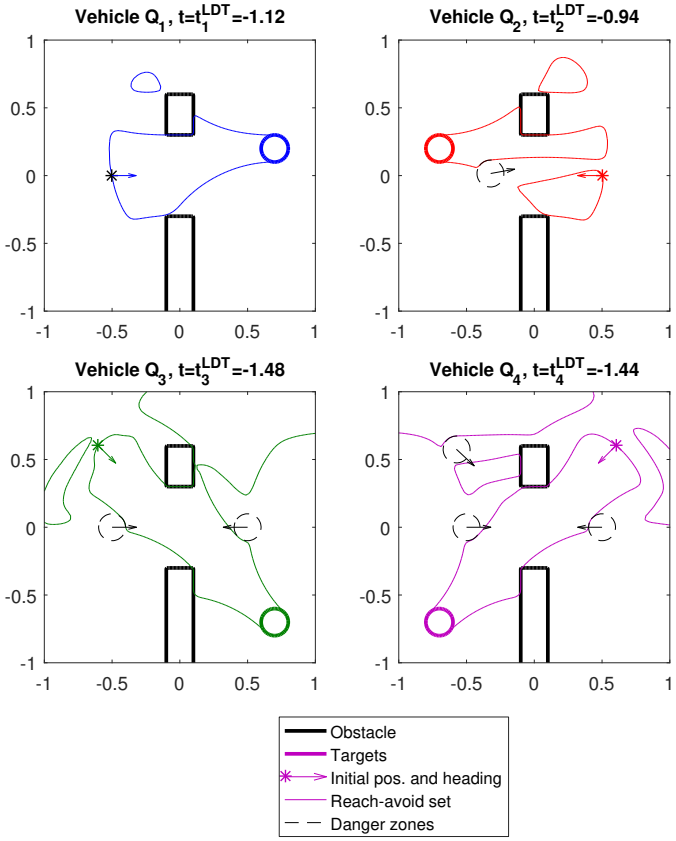

Fig. 2: BRSs at $t = t_i^{\text{LDT}}$ for vehicles $1, 2, 3, 4$, sliced at initial headings θ_i^0. Black arrows indicate direction of obstacle motion. Due to the turn rate constraint, the presence of static obstacles $\mathcal{O}_i^{\text{static}}$ and time-varying obstacles induced by higher-priority vehicles $\mathcal{O}_i^j(t)$ carve "channels" in the BRS, dividing it up into multiple "islands".

A. Theory

We present three methods to address the above issues. The methods differ in terms of control policy information that is known to a lower-priority vehicle, and have their relative advantages and disadvantages depending on the situation. The three methods are as follows:

- **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 target on time, but has no other restrictions on its control policy. When the control policy of a vehicle is unknown, but its timely arrive at its target can be assumed, the least restrictive control can be safely assumed by lower-priority vehicles.
- **Robust trajectory tracking**: A vehicle declares a nominal trajectory which can be robustly tracked under disturbances.

In general, the above methods can be used in combination in a single trajectory planning problem, with each vehicle independently having different control policies. Lower-priority vehicles would then plan their trajectories while taking into account the control policy information known for each higher-

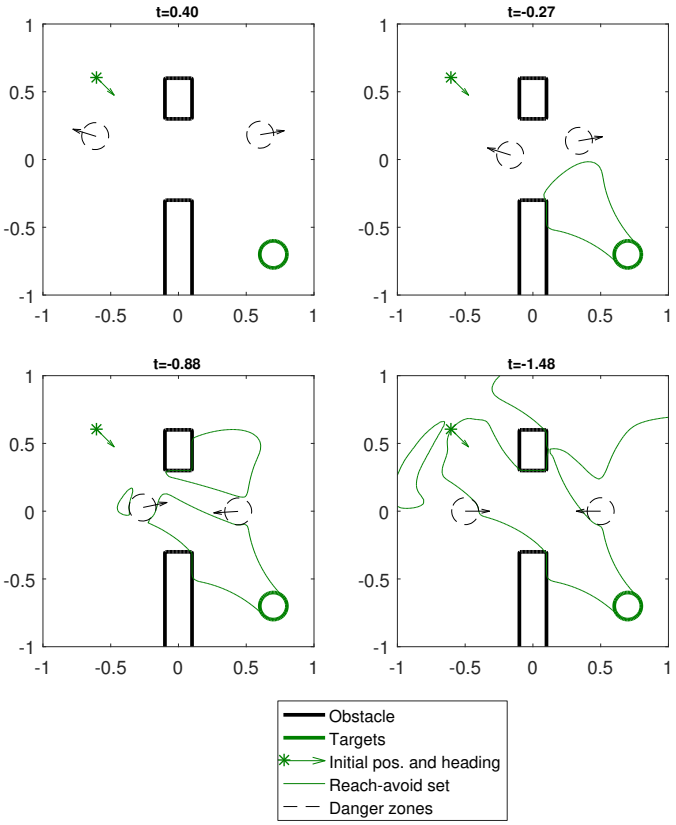

Fig. 3: Time evolution of the BRS for vehicle Q_3, sliced at its initial heading $\theta_3^0 = \frac{7\pi}{4}$. Black arrows indicate direction of obstacle motion. Top row: the BRS grows unobstructed by obstacles (time-varying and static). Bottom row: the static obstacles $\mathcal{O}_i^{\text{static}}$ and the induced obstacles $\mathcal{O}_3^1, \mathcal{O}_3^2$, carve out "channels" in the BRS.

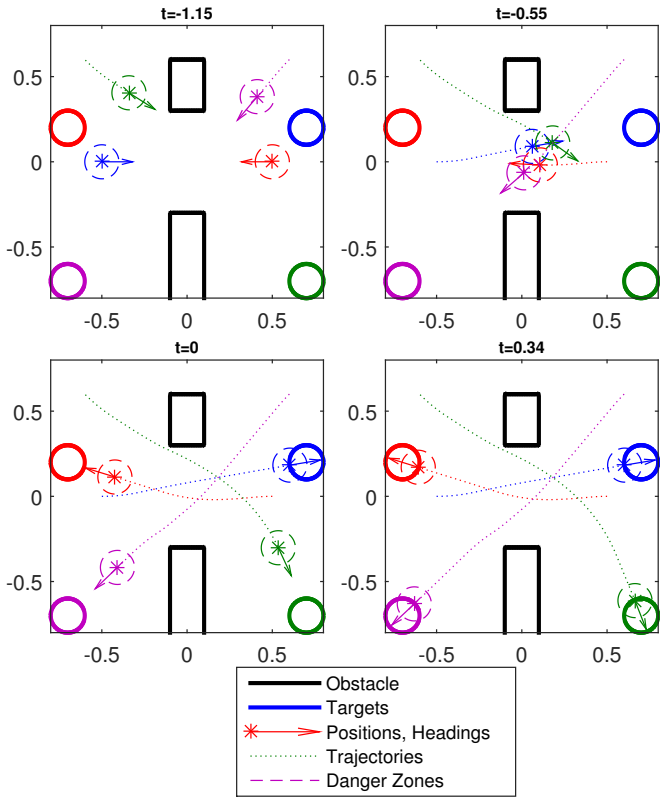

Fig. 4: The planned trajectories of the four vehicles. Top left: only vehicles Q_3 (green) and Q_4 (purple) have started moving, showing t_i^{LDT} is not common across the vehicles. Top right: all vehicles have come within very close proximity, but none is in the danger zone of another. Bottom left: vehicle Q_1 (blue) arrives at $\mathcal{L}_1$ at $t = 0$. Bottom right: all vehicles have reached their destination, some ahead of t_i^{STA}.

priority vehicle. For clarity, we will present each method as if all vehicles are using the same method of trajectory planning.

In addition, for simplicity of explanation, we will assume that no static obstacles exist. In the situations where static obstacles do exist, the time-varying obstacles $\mathcal{G}_i(t)$ simply become the union of the induced obstacles $\mathcal{O}_i^j(t)$ in (10) and the static obstacles. The material in this section is taken partially from [38].

1) Centralized Control: The highest-priority vehicle Q_1 first plans its trajectory by computing the BRS (with $i = 1$)

$$\begin{aligned}\mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}}) = \{&y : \exists u_i(\cdot) \in \mathbb{U}_i, \forall d_i(\cdot) \in \mathbb{D}_i, x_i(\cdot) \text{ satisfies (1)},\\ &\forall s \in [t, t_i^{\text{STA}}], x_i(s) \notin \mathcal{G}_i(s), x_i(t) = y\\ &\exists s \in [t, t_i^{\text{STA}}], x_i(s) \in \mathcal{L}_i\}\end{aligned} \tag{17}$$

Since we have assumed no static obstacles exist, we have that for $Q_1, \mathcal{G}_1(s) = \emptyset \ \forall s \leq t_i^{\text{STA}}$, and thus the above BRS is well-defined. This BRS can be computed by solving the HJ VI (5) with the following Hamiltonian:

$$H_i^{\text{dstb}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i} \max_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \tag{18}$$

From the BRS, we can obtain the optimal control

$$u_i^{\text{dstb}}(t, x_i) = \arg \min_{u_i \in \mathcal{U}_i} \max_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \tag{19}$$

Here, as well as in the other two methods, the latest departure time t_i^{LDT} is then given by $\arg\sup_t x_i^0 \in \mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}})$.

If there is a central agent directly controlling each of the N vehicles, then the control law of each vehicle can be enforced. In this case, lower-priority vehicles can safely assume that higher-priority vehicles are applying the enforced control law. In particular, the optimal controller for getting to the target, $u_i^{\text{dstb}}(t, x_i)$, can be enforced. In this case, the dynamics of each vehicle becomes

$$\begin{aligned}\dot{x}_i &= f_i^{\text{cc}}(t, x_i, d_i) = f_i(x_i, u_i^{\text{dstb}}(t, x_i), d_i)\\ d_i &\in \mathcal{D}_i, \quad i = 1, \ldots, N, \quad t \in [t_i^{\text{LDT}}, t_i^{\text{STA}}]\end{aligned} \tag{20}$$

where u_i no longer appears explicitly in the dynamics.

From the perspective of a lower-priority vehicle Q_i, a higher-priority vehicle $Q_j, j < i$ induces a time-varying obstacle that represents the positions that could possibly be within the collision radius R_c of Q_j under the dynamics $f_j^{\text{cc}}(t, x_j, d_j)$. Determining this obstacle involves computing

an FRS of Q_j starting from[3] $x_j(t_j^{\text{LDT}}) = x_j^0$. The FRS $\mathcal{W}_j^{\text{cc}}(t_j^{\text{LDT}}, t)$ is defined as follows:

$$\begin{aligned}\mathcal{W}_j^{\text{cc}}(t_j^{\text{LDT}}, t) =& \{y : \exists d_j(\cdot) \in \mathbb{D}_j, x_j(\cdot) \text{ satisfies (20)}, \\ & x_j(t_j^{\text{LDT}}) = x_j^0, x_j(t) = y\}.\end{aligned} \tag{21}$$

This FRS can be computed using (6) with the Hamiltonian

$$H_j^{\text{cc}}(t, x_j, \lambda) = \max_{d_j \in \mathcal{D}_j} \lambda \cdot f_j^{\text{cc}}(t, x_j, d_j) \tag{22}$$

The FRS $\mathcal{W}_j^{\text{cc}}(t_j^{\text{LDT}}, t)$ represents the set of possible states at time t of a higher-priority vehicle Q_j given all possible disturbances $d_j(\cdot)$ and given that Q_j uses the feedback controller $u_j^{\text{dstb}}(t, x_j)$. In order for a lower-priority vehicle Q_i to guarantee that it does not go within a distance of R_c to Q_j, Q_i must stay a distance of at least R_c away from the FRS $\mathcal{W}_j^{\text{cc}}(t_j^{\text{LDT}}, t)$ for all possible values of the non-position states h_j. This gives the obstacle induced by a higher-priority vehicle Q_j for a lower-priority vehicle Q_i as follows:

$$\mathcal{O}_i^j(t) = \{x_i : \exists y \in \mathcal{P}_j(t), \|p_i - y\|_2 \leq R_c\} \tag{23}$$

where the set $\mathcal{P}_j(t)$ is the set of states in the FRS $\mathcal{W}_j^{\text{cc}}(t_j^{\text{LDT}}, t)$ projected onto the states representing position p_j, and disregarding the non-position dimensions h_j:

$$\mathcal{P}_j(t) = \{p_j : \exists h_j, (p_j, h_j) \in \mathcal{B}_j(t)\}, \tag{24}$$

$$\mathcal{B}_j(t) = \mathcal{W}_j^{\text{cc}}(t_j^{\text{LDT}}, t). \tag{25}$$

Finally, taking the union of the induced obstacles $\mathcal{O}_i^j(t)$ as in (9) gives us the time-varying obstacles $\mathcal{G}_i(t)$ needed to define and determine the BRS $\mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}})$ in (17). Repeating this process, all vehicles will be able to plan trajectories that guarantee the vehicles' timely and safe arrival. The centralized control algorithm can be summarized as follows:

Algorithm 2: **Centralized control algorithm**: Given initial conditions x_i^0, vehicle dynamics (1), target set $\mathcal{L}_i$, and static obstacles $\mathcal{O}_i^{\text{static}}, i = 1 \ldots, N$, for each i,

1) determine the total obstacle set $\mathcal{G}_i(t)$, given in (9). In the case $i = 1$, $\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}}\ \forall t$;
2) compute the BRS $\mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}})$ defined in (17). The latest departure time t_i^{LDT} is then given by $\arg\sup_t x_i^0 \in \mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}})$;
3) compute the optimal control $u_i^{\text{dstb}}(t, x_i)$ corresponding to $\mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}})$ given by (19). Given $u_i^{\text{dstb}}(t, x_i)$, compute the FRS $\mathcal{W}_i^{\text{cc}}(t_i^{\text{LDT}}, t)$ in (21);
4) finally, compute the induced obstacles $\mathcal{O}_k^i(t)$ for each $k > i$. In the centralized control method, $\mathcal{O}_k^i(t)$ is computed using (23) where $\mathcal{P}_i(t)$ is given by (24).

2) Least Restrictive Control: Here, we again begin with the highest-priority vehicle Q_1 planning its trajectory by computing the BRS $\mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}})$ in (17). However, if there is no centralized controller to enforce the control policy for higher-priority vehicles, weaker assumptions must be made by the lower-priority vehicles to ensure collision avoidance. One reasonable assumption is that all higher-priority vehicles follow the least restrictive control that would take them to their targets. This control would be given by

$$u_j^{\text{lrc}}(t, x_j) \in \begin{cases} \{u_j^{\text{dstb}}(t, x_j) \text{ in (19)}\} \text{ if } x_j(t) \in \partial\mathcal{V}_j^{\text{dstb}}(t, t_j^{\text{STA}}), \\ \mathcal{U}_j \text{ otherwise} \end{cases} \tag{26}$$

Such a controller allows each higher-priority vehicle to use any controller it desires, except when it is on the boundary of the BRS, $\partial\mathcal{V}_j^{\text{dstb}}(t, t_j^{\text{STA}})$, in which case the optimal control $u_j^{\text{dstb}}(t, x_j)$ given by (19) must be used to get to the target safely and on time. This assumption is the weakest assumption that could be made by lower-priority vehicles given that the higher-priority vehicles will get to their targets on time.

Suppose a lower-priority vehicle Q_i assumes that higher-priority vehicles $Q_j, j < i$ use the least restrictive control strategy $u_j^{\text{lrc}}(t, x_j)$ in (26). From the perspective of the lower-priority vehicle Q_i, a higher-priority vehicle Q_j could be in any state that is reachable from Q_j's initial state $x_j(t_j^{\text{LDT}}) = x_j^0$ and from which the target $\mathcal{L}_j$ can be reached. Mathematically, this is defined by the intersection of an FRS $\mathcal{W}_j^{\text{lrc}}(t_j^{\text{LDT}}, t)$ from the initial state $x_j(t_j^{\text{LDT}}) = x_j^0$ and the BRS $\mathcal{V}_j^{\text{dstb}}(t, t_j^{\text{STA}})$ defined in (17) from the target set $\mathcal{L}_j$, $\mathcal{W}_j^{\text{lrc}}(t_j^{\text{LDT}}, t) \cap \mathcal{V}_j^{\text{dstb}}(t, t_j^{\text{STA}})$. In this situation, since Q_j cannot be assumed to be using any particular feedback control, $\mathcal{W}_j^{\text{lrc}}(t_j^{\text{LDT}}, t)$ is defined as

$$\begin{aligned}\mathcal{W}_j^{\text{lrc}}(t_j^{\text{LDT}}, t) =& \{y : \exists u_j(\cdot) \in \mathbb{U}_j, \exists d_j(\cdot) \in \mathbb{D}_j, \\ & x_j(\cdot) \text{ satisfies (1)}, x_j(t_j^{\text{LDT}}) = x_j^0, \\ & x_j(t) = y\}.\end{aligned} \tag{27}$$

This FRS can be computed by solving (6) with the Hamiltonian

$$H_j^{\text{lrc}}(x_j, \lambda) = \max_{u_j \in \mathcal{U}_j} \max_{d_j \in \mathcal{D}_j} \lambda \cdot f_j(x_j, u_j, d_j) \tag{28}$$

In turn, the obstacle induced by a higher-priority Q_j for a lower-priority vehicle Q_i is as follows:

$$\mathcal{O}_i^j(t) = \{x_i : \exists y \in \mathcal{P}_j(t), \|p_i - y\|_2 \leq R_c\}, \text{ where} \tag{29}$$

$$\mathcal{P}_j(t) = \{p_j : \exists h_j, (p_j, h_j) \in \mathcal{B}_j(t)\}, \tag{30}$$

$$\mathcal{B}_j(t) = \mathcal{W}_j^{\text{lrc}}(t_j^{\text{LDT}}, t) \cap \mathcal{V}_j^{\text{dstb}}(t, t_j^{\text{STA}}). \tag{31}$$

The least restrictive control method can be summarized as follows:

Algorithm 3: **Least restrictive control algorithm**: Given initial conditions x_i^0, vehicle dynamics (1), target set $\mathcal{L}_i$, and static obstacles $\mathcal{O}_i^{\text{static}}, i = 1 \ldots, N$, for each i,

1) determine the total obstacle set $\mathcal{G}_i(t)$, given in (9). In the case $i = 1$, $\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}}\ \forall t$;
2) compute the BRS $\mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}})$ defined in (17). The latest departure time t_i^{LDT} is then given by $\arg\sup_t x_i^0 \in \mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}})$;

[3]In practice, we define the target set to be a small region around the vehicle's initial state for computational reasons.

3) compute the FRS $\mathcal{W}_i^{\text{lrc}}(t)$ in (27). Given $\mathcal{W}_i^{\text{lrc}}(t_i^{\text{LDT}}, t)$ and $\mathcal{V}_i^{\text{dstb}}(t, t_i^{\text{STA}})$, compute the positions that Q_i could be in. The set of these positions is given by (30);
4) compute the induced obstacles $\mathcal{O}_k^i(t)$ for each $k > i$ using (29).

Remark 1: The centralized control method described in the previous section can be thought of as the "most restrictive control" method, in which all vehicles must use the optimal controller at all times, while the least restrictive control method allows vehicles to use any suboptimal controller that allows them to arrive at the target on time. These two methods can be considered two extremes of a spectrum in which varying degrees of optimality is assumed for higher-priority vehicles. Vehicles can also choose a control strategy in the middle of the two extremes, and for example use a control within some range around the optimal control, or use the optimal control unless some condition is met. The induced obstacles and the BRS can then be similarly computed using the corresponding control strategy.

3) Robust Trajectory Tracking: Even though it is impossible to commit to and track an exact trajectory in the presence of disturbances, it may still be possible to instead *robustly* track a feasible *nominal* trajectory with a bounded error at all times. If this can be done, then the tracking error bound can be used to determine the induced obstacles. Here, computation is done in two phases: the *planning phase* and the *disturbance rejection phase*. In the planning phase, we compute a nominal trajectory $x_{r,j}(\cdot)$ that is feasible in the absence of disturbances. In the disturbance rejection phase, we compute a bound on the tracking error.

It is important to note that the planning phase does not make full use of a vehicle's control authority, as some margin is needed to reject unexpected disturbances while tracking the nominal trajectory. Therefore, in this method, planning is done for a reduced control set $\mathcal{U}^p \subset \mathcal{U}$. The resulting trajectory reference will not utilize the vehicle's full control capability; additional maneuverability is available at execution time to counteract external disturbances.

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

Let x_j and $x_{r,j}$ denote the states of the actual vehicle Q_j and the virtual evader, respectively, and define the tracking error $e_j = x_j - x_{r,j}$. When the error dynamics are independent of the absolute state as in (32) (and also (7) in [21]), we can obtain error dynamics of the form

$$\begin{aligned} \dot{e_j} &= f_{e_j}(e_j, u_j, u_{r,j}, d_j), \\ u_j &\in \mathcal{U}_j, u_{r,j} \in \mathcal{U}_j^p, d_j \in \mathcal{D}_j, \quad t \leq 0 \end{aligned} \tag{32}$$

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

$$\mathcal{E}_j = \{e_j : \|p_{e_j}\|_2 \leq R_{\text{EB}}\}, \tag{33}$$

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

$$\begin{aligned} \mathcal{V}_j^{\text{EB}}(t,0) = \{&y : \forall u_j(\cdot) \in \mathbb{U}_j, \exists u_{r,j}(\cdot) \in \mathbb{U}_j^p, \exists d_j(\cdot) \in \mathbb{D}_i, \\ &e_j(\cdot) \text{ satisfies (32)}, e_j(t) = y, \\ &\exists s \in [t, 0], e_j(s) \in \mathcal{E}_j^c\}, \end{aligned} \tag{34}$$

where the Hamiltonian to compute the BRS is given by:

$$H_j^{\text{EB}}(e_j, \lambda) = \max_{u_j \in \mathcal{U}_j} \min_{u_r \in \mathcal{U}_j^p, d_j \in \mathcal{D}_j} \lambda \cdot f_{e_j}(e_j, u_j, u_{r,j}, d_j). \tag{35}$$

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

$$\kappa_j(e_j) = \arg \max_{u_j \in \mathcal{U}_j} \min_{u_r \in \mathcal{U}_j^p, d_j \in \mathcal{D}_j} \lambda \cdot f_{e_j}(e_j, u_j, u_{r,j}, d_j). \tag{36}$$

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

$$\begin{aligned} \mathcal{O}_i^j(t) &= \{x_i : \exists y \in \mathcal{P}_j(t), \|p_i - y\|_2 \leq R_c\} \\ \mathcal{P}_j(t) &= \{p_j : \exists h_j, (p_j, h_j) \in \mathcal{B}_j(t)\} \\ \mathcal{B}_j(t) &= \Omega_j + x_{r,j}(t), \end{aligned} \tag{37}$$

where the "+" in (37) denotes the Minkowski sum[4]. Intuitively, if Q_j is tracking $x_{r,j}(t)$, then it will remain within the error bound Ω_j around $x_{r,j}(t)$ $\forall t$. This set is mathematically given by the "tube" obtained by augmenting the error bound Ω_j around the refernece trajectory $x_{r,j}(\cdot)$. This is precisely the set $\mathcal{B}_j(t)$ (and $\mathcal{P}_j(t)$ for the position states). The induced obstacles can then be obtained by augmenting a danger zone around this set. Finally, we can obtain the total obstacle set $\mathcal{G}_i(t)$ using (9).

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

$$\tilde{\mathcal{G}}_i(t) = \mathcal{G}_i(t) + \Omega_i. \tag{38}$$

Finally, given Ω_i, we can guarantee that Q_i will reach its target $\mathcal{L}_i$ if $\Omega_i \subseteq \mathcal{L}_i$; thus, in the trajectory planning phase, we modify $\mathcal{L}_i$ to be $\tilde{\mathcal{L}}_i := \{x_i : \Omega_i + x_i \subseteq \mathcal{L}_i\}$, and compute

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

a BRS, with the control authority $\mathcal{U}_i^p$, that contains the initial state of the vehicle. Mathematically,

$$\begin{aligned}\mathcal{V}_i^{\text{rtt}}(t, t_i^{\text{STA}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i^p, x_i(\cdot) \text{ satisfies (8)}, \\ & \forall s \in [t, t_i^{\text{STA}}], x_i(s) \notin \tilde{\mathcal{G}}_i(t), \\ & \exists s \in [t, t_i^{\text{STA}}], x_i(s) \in \tilde{\mathcal{L}}_i, x_i(t) = y\}\end{aligned} \tag{39}$$

The BRS $\mathcal{V}_i^{\text{rtt}}(t, t_i^{\text{STA}})$ can be obtained by solving (5) using the Hamiltonian:

$$H_i^{\text{rtt}}(x_i, \lambda) = \min_{u_i \in \mathcal{U}_i^p} \lambda \cdot f_i(x_i, u_i) \tag{40}$$

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

$$u_i^{\text{rtt}}(t) = \arg\min_{u_i \in \mathcal{U}_i^p} \lambda \cdot f_i(x_i, u_i). \tag{41}$$

The nominal trajectory $x_{r,i}(\cdot)$ can thus be obtained by using vehicle dynamics (8), with the optimal control $u_i^{\text{rtt}}(\cdot)$ given by (41). From the resulting nominal trajectory $x_{r,i}(\cdot)$, the overall control policy to reach $\mathcal{L}_i$ can be obtained via (36). The robust trajectory tracking method can be summarized as follows:

Algorithm 4: **Robust trajectory tracking algorithm**: Given initial conditions x_i^0, vehicle dynamics (1), target sets $\mathcal{L}_i$, and static obstacles $\mathcal{O}_i^{\text{static}}, i = 1 \ldots, N$, for each i,

1) determine the total obstacle set $\mathcal{G}_i(t)$, given in (9). In the case $i = 1$, $\mathcal{G}_i(t) = \mathcal{O}_i^{\text{static}}$ $\forall t$;
2) decide on a reduced control authority $\mathcal{U}_i^p$ for the planning phase, and choose a parameter R_{EB} to conservatively bound the tracking error;
3) compute the BRS $\mathcal{V}_i^{\text{EB}}(t, 0)$ using (34) and make sure that $\Omega_i \neq \emptyset$. Given R_{EB}, the error bound on the tracking error is given by Ω_i;
4) using Ω_i, determine the augmented obstacle set $\tilde{\mathcal{G}}_i(t)$, given in (38);
5) compute the BRS $\mathcal{V}_i^{\text{rtt}}(t, t_i^{\text{STA}})$ as described in (39) using the reduced target set $\tilde{\mathcal{L}}_i$, $\tilde{\mathcal{G}}_i(t)$ as obstacles, and the control authority $\mathcal{U}_i^p$. The latest departure time t_i^{LDT} is then given by $\arg\sup_t x_i^0 \in \mathcal{V}_i^{\text{rtt}}(t, t_i^{\text{STA}})$;
6) compute the nominal trajectory $x_{r,i}(\cdot)$ for Q_i in the absence of disturbances, which can be obtained using the vehicle dynamics in (8) and the optimal control given in (41);
7) the induced obstacles $\mathcal{O}_k^i(t)$ for each $k > i$ can be computed using Ω_i and $x_{r,i}(\cdot)$ via (37).

B. Numerical Simulations

We demonstrate our proposed methods for accounting for disturbances and incomplete information using a four-vehicle example. Each vehicle has the simple kinematics model in (14) but with disturbances added to the evolution of each state:

$$\begin{aligned}\dot{p}_{x,i} &= v_i \cos\theta_i + d_{x,i} \\ \dot{p}_{y,i} &= v_i \sin\theta_i + d_{y,i} \\ \dot{\theta}_i &= \omega_i + d_{\theta,i}, \\ \underline{v} \leq v_i \leq \bar{v}&, |\omega_i| \leq \bar{\omega}, \\ \|(d_{x,i}, d_{y,i})\|_2 \leq d_r&, |d_{\theta,i}| \leq \bar{d}_\theta\end{aligned} \tag{42}$$

where $d = (d_{x,i}, d_{y,i}, d_{\theta,i})$ represents Q_i's disturbances in the three states. 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. For illustration purposes, we choose $\underline{v} = 0.5, \bar{v} = 1, \bar{\omega} = 1$; however, our method can easily handle the case in which these inputs differ across vehicles. The disturbance bounds are chosen as $d_r = 0.1, \bar{d}_\theta = 0.2$, which correspond to a 10% uncertainty in the dynamics.

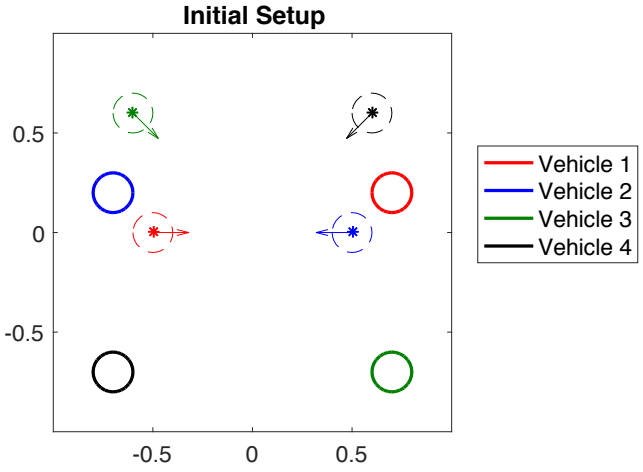

Fig. 5: Initial configuration of the four-vehicle example in the presence of disturbances.

For this example, we have chosen scheduled times of arrival $t_i^{\text{STA}} = 0$ $\forall i$ for simplicity. Each vehicle aims to get to a target set of the form (15) with target radius $r = 0.1$. The vehicles' target centers c_i and initial conditions x_i^0 are given by (16).

These parameters are the same as the example in Section IV-B, except that the t_i^{STA} values are the same for all vehicles, and that there are no static obstacles. The problem setup for this example is shown in Fig. 5.

With the above parameters, we obtain $t_i^{\text{LDT}}, i = 1, 2, 3, 4$. Note that even though t_i^{STA} is assumed to be same for all vehicles in this example for simplicity, our method can easily handle the case in which t_i^{STA} is different for each vehicle as we have already shown in Section IV-B.

For each proposed method of computing induced obstacles, we show the vehicles' entire trajectories (colored dotted lines), and overlay their positions (colored asterisks) and headings (arrows) at a point in time in which they are in a relatively dense configuration. In all cases, the vehicles are able to avoid each other's danger zones (colored dashed circles) while getting to their target sets in minimum time. In addition, we show the evolution of the BRS over time for Q_3 (green boundaries) as well as the obstacles induced by the higher-priority vehicles (black boundaries).

Offline computations were done on a desktop computer with a Core i7 5820K processor and two GeForce GTX Titan X graphics processing units. The average computation time per vehicle is approximately 2 second using CUDA and GPU parallelization.

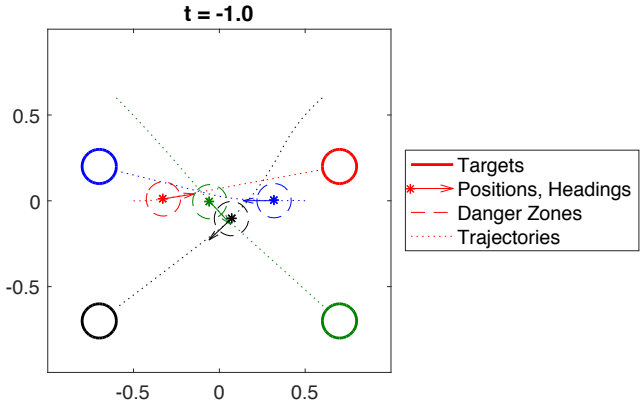

Fig. 6: Simulated trajectories in the centralized control method. Since the higher priority vehicles induce relatively small obstacles in this case, vehicles do not deviate much from a straight line trajectory towards their respective targets, and arrive at a dense configuration similar to that in Fig. 4.

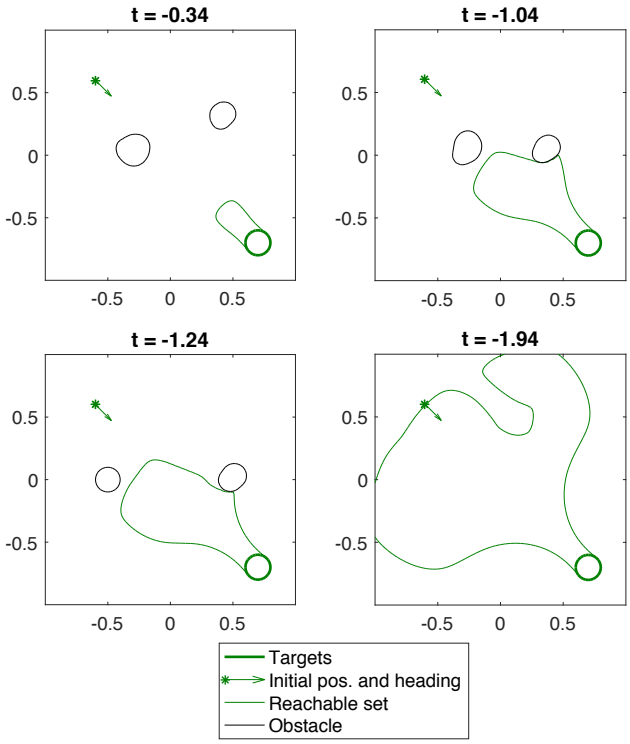

Fig. 7: Evolution of the BRS and the obstacles induced by Q_1 and Q_2 for Q_3 in the centralized control method. Since vehicles apply the optimal control at all times, the obstacle sizes are only slightly bigger than those in Fig. 2 and 3.

1) Centralized Control: Fig. 6 shows the simulated trajectories in the situation where a centralized controller enforces each vehicle to use the optimal controller $u_i^{\text{dstb}}(t, x_i)$ according to (19), as described in Section V-A1. In this case, vehicles appear to deviate slightly from a straight line trajectory towards their respective targets, just enough to avoid higher-priority vehicles. The deviation is small since the centralized controller is quite restrictive, making the possible positions of higher-priority vehicles cover a small area. In the dense configuration at $t = -1.0$, the vehicles are close to each other but still outside each other's danger zones.

Fig. 7 shows the evolution of the BRS for Q_3 (green boundary), as well as the obstacles (black boundary) induced by the higher-priority vehicles Q_1 and Q_2. The locations of the induced obstacles at different time points include the actual positions of Q_1 and Q_2 at those times, and the sizes of obstacles remain relatively small. The t_i^{LDT} values for the four vehicles (in order) in this case are $-1.35, -1.37, -1.94$ and -2.04, relatively close for vehicles pairs (Q_1, Q_2) and (Q_3, Q_4), because the obstacles generated by higher-priority vehicles are small and hence do not affect the t_i^{LDT} of lower-priority vehicles significantly.

2) Least Restrictive Control: Fig. 8 shows the simulated trajectories in the situation where each vehicle assumes that higher-priority vehicles use the least restrictive control to reach their targets, as described in V-A2. Fig. 9 shows the BRS and induced obstacles for Q_3.

Q_1 (red) takes a relatively straight path to reach its target. From the perspective of all other vehicles, large obstacles are induced by Q_1, since lower-priority vehicles make the weak assumption that higher-priority vehicles are using the least restrictive control. Because the obstacles induced by higher-priority vehicles are so large, it is faster for lower-priority vehicles to wait until higher-priority vehicles pass by than to move around the higher-priority vehicles. As a result, the vehicles never form a dense configuration, and their trajectories are all relatively straight, indicating that they end up taking a short path to the target after higher-priority vehicles pass by. This is also indicated by the early t_i^{LDT}

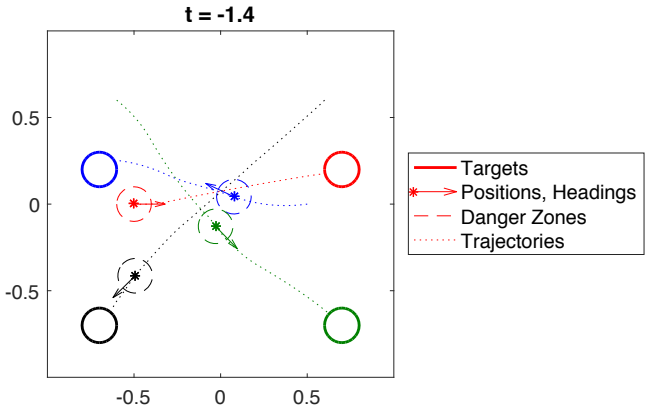

Fig. 8: Simulated trajectories in the least restrictive control method. All vehicles start moving before Q_1 starts, because the large obstacles make it optimal to wait until higher priority vehicles pass by, leading to earlier t_i^{LDT}'s.

values for the four vehicles, $-1.35, -1.97, -2.66$ and -3.39, respectively. Compared to the centralized control method, t_i^{LDT}'s are significantly earlier for all vehicles, except Q_1, the highest-priority vehicle, since it need not account for any moving obstacles.

From Q_3's (green) perspective, the large obstacles induced by Q_1 and Q_2 are shown in Fig. 9 as the black boundary. As the BRS (green boundary) evolves over time, its growth gets inhibited by the large obstacles for a long time, from $t = -0.89$ to $t = -1.39$. Eventually, the boundary of the BRS reaches the initial state of Q_3 at $t = t_3^{\text{LDT}} = -2.66$.

3) Robust Trajectory Tracking: In the planning phase, we reduced the maximum turn rate of the vehicles from 1 to 0.6, and the speed range from $[0.5, 1]$ to exactly 0.75 (constant speed). With these reduced control authorities, we determined from the disturbance rejection phase that a nominal trajectory from the planning phase can be robustly tracked within a distance of $R_{\text{EB}} = 0.075$.

Fig. 10 shows the vehicle trajectories in the situation where each vehicle robustly tracks a pre-specified trajectory and is guaranteed to stay inside a "bubble" around the trajectory. Fig. 11 shows the evolution of BRS and induced obstacles for vehicle Q_3. The obstacles induced by other vehicles inhibit the evolution of the BRS, carving out thin channels, which can be seen at $t = -2.59$, that separate the BRS into different islands.

In this case, the t_i^{LDT} values for the four vehicles are $-1.61, -3.16, -3.57$ and -2.47 respectively. In this method, vehicles use reduced control authority for trajectory planning towards a reduced-size effective target set. As a result, higher-priority vehicles tend to have earlier t_i^{LDT}'s compared to the other two methods, as evident from t_1^{LDT}. Because of this "sacrifice" made by the higher-priority vehicles during the trajectory planning phase, the t^{LDT}'s of lower-priority vehicles may be later compared to those in the other methods, as evident from t_4^{LDT}. Overall, it is unclear how t_i^{LDT} will change for a vehicle compared to the other methods, as the conservative trajectory planning leads to earlier t_i^{LDT}'s for higher-priority vehicles and later t_i^{LDT}'s for lower-priority vehicles.

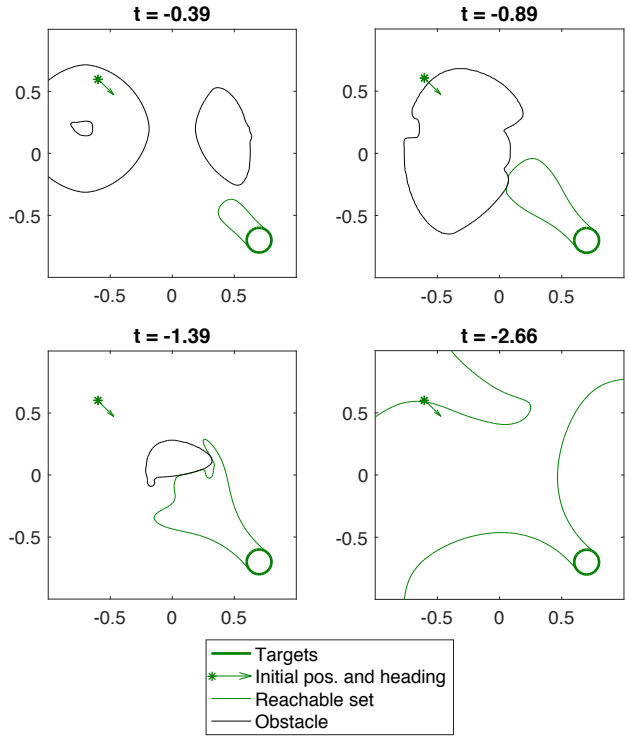

Fig. 9: Evolution of the BRS for Q_3 in the least restrictive control method. In this case, $t_3^{\text{LDT}} = -2.66$, significantly earlier than that in the centralized control method (-1.94), reflecting the impact of larger induced obstacles.

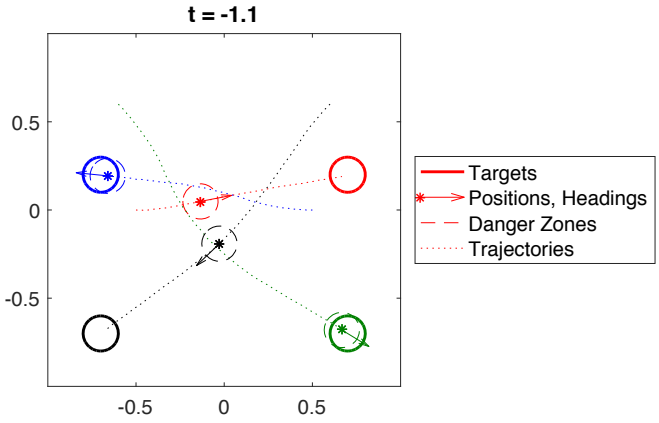

Fig. 10: Simulated trajectories for the robust trajectory tracking method.

VI. STP With An Intruder

In Section V, we made the basic STP algorithm more robust by taking into account disturbances and considering situations in which vehicles may not have complete information about the control strategy of the other vehicles. 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 original plan can lead to vehicles entering into another vehicle's danger zone.

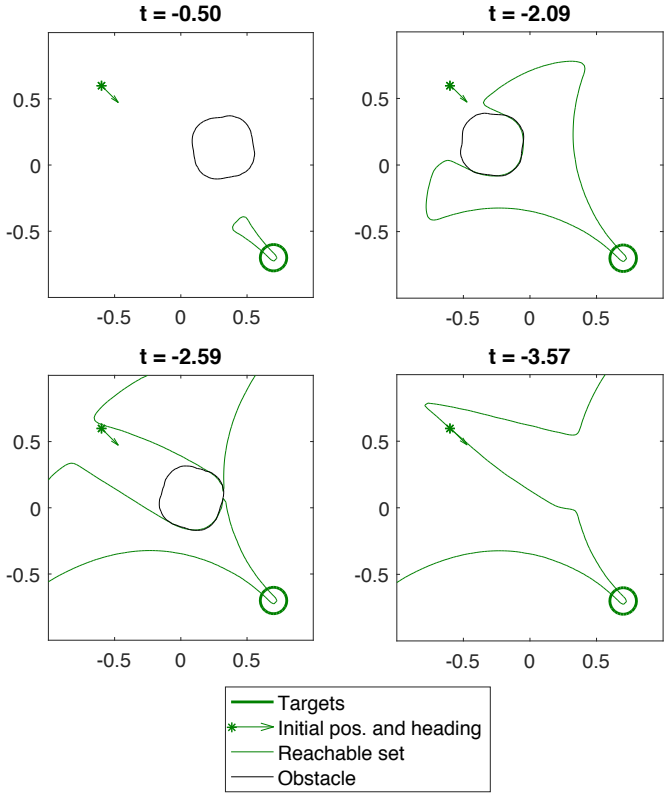

Fig. 11: Evolution of the BRS for Q_3 in the robust trajectory tracking method. As the BRS grows in time, the induced obstacles carve out a channel. Note that a smaller target set is used to compute the BRS to ensure that the vehicle reaches the target set by $t = 0$ for any allowed tracking error.

If vehicles do not plan with an additional safety margin that takes a potential intruder into account, a vehicle trying to avoid the intruder may effectively become an intruder itself, leading to a domino effect. In this section, we propose a method to allow vehicles to avoid an intruder while maintaining the STP structure.

A. Theory

In general, the effect of an intruder on the vehicles in structured flight can be entirely unpredictable, since the intruder in principle could be adversarial in nature, and the number of intruders could be arbitrary. Therefore, for our analysis to produce reasonable results, two assumptions about the intruders must be made.

Assumption 1: At most one intruder (denoted as Q_I) affects the STP vehicles at any given time. The intruder is removed from the system after affecting the STP vehicles after a duration of t^{IAT}. The removal of the intruder can be done, for example, by forcing it out of the altitude range of the STP vehicles.

Let the time at which intruder appears in the system be $\underline{t}$ and the time at which it disappears be $\bar{t}$. Assumption 1 implies that $\bar{t} \leq \underline{t} + t^{\text{IAT}}$. Thus, any vehicle Q_i would need to avoid the intruder Q_I for a maximum duration of t^{IAT}. This assumption can be valid in situations where intruders are rare, and that some fail-safe or enforcement mechanism exists to force the intruder out of the altitude level affecting the STP vehicles. Note that we do not make any assumptions about $\underline{t}$; however, we assume that once it 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. Even though assumption 2 often does not hold for practical systems, a conservative model of the intruder may be used in situations where the dynamics of the intruder are not known exactly. 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.

Based on the above assumptions, we aim to design a control policy that ensures for each STP vehicle separation with the intruder and with other STP vehicles, and successful transit to the destination. However, depending on the initial state of the intruder, its control policy, and the disturbances in the dynamics of a vehicle and the intruder, a vehicle may arrive at different states after avoiding the intruder. Therefore, a control policy that ensures a successful transit to the destination needs to account for all such possible states, which is a trajectory planning problem with multiple (infinite, to be precise) 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 or successful avoidance of the intruder if it does appear. (ii) After the intruder is removed from the system at $\bar{t}$, we solve a new STP problem (that is, we "re-plan") for vehicles which needed to avoid the intruder. In this case, the affected vehicles will re-plan as lowest-priority vehicles starting from the initial they happen to arrive at after avoiding the intruder. Suppose some vehicle Q_i starts avoiding the intruder Q_I at some time $t = \underline{t}$, and stops avoiding at $t = \bar{t}$. When $t < \underline{t}$, Q_i must plan its trajectory taking into account the possibility that it may need to avoid an intruder Q_I. Since Q_i may spend a duration of up to t^{IAT} performing avoidance, its induced obstacles $\mathcal{O}_k^i(t), k > i$ need to be computed in a way that reflects this possibility. The induced obstacles computation is discussed in Section VI-A1.

We must also ensure that while avoiding the intruder, Q_i does not collide with the total obstacle set $\mathcal{G}_i(t)$. This requires computing the augmented total obstacle $\tilde{\mathcal{G}}_i(t)$; the computation of $\tilde{\mathcal{G}}_i(t)$ and the controller that guarantees the avoidance of the augmented obstacles are discussed in Section VI-A2.

In Section VI-A3, we describe how Q_i can guarantee collision avoidance with the intruder. A pairwise collision avoidance problem such as this has been solved in isolation in [21].

Finally, when $t > \bar{t}$, Q_i has already successfully avoided the intruder, but depending on the state it happens to arrive at after avoiding the intruder, it may need to re-plan its trajectory to reach the target safely. The re-planning process is discussed in Section VI-A4.

1) Induced Obstacle Computation: The goal of this section is to compute, for each lower-priority vehicle Q_i, the time-varying obstacles induced by each higher-priority vehicle

$Q_j, j < i$, denoted by $\mathcal{O}_i^j(t)$. As before, the total obstacle set $\mathcal{G}_i(t)$ can then be obtained using (9). To compute the obstacle that Q_i needs to avoid 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 the STP vehicles would already be in the re-planning phase at time t and hence cannot be in conflict.

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 V to compute the "base" obstacles $\mathcal{B}_j(t)$; these are the obstacles that would have been induced by Q_j in the absence of an intruder. The base obstacles are respectively given by (25), (31) and (37) for the centralized control, least restrictive control and robust trajectory tracking methods.

The induced obstacles, $\mathcal{O}_i^j(t)$, are then given by the states that Q_j can reach while avoiding the intruder, starting from some state in $\mathcal{B}_j(\underline{t}), \underline{t} \in [t-t^{\text{IAT}}, t]$. These states can be obtained by computing an FRS from the base obstacles.

$$\begin{aligned}\mathcal{W}_j^{\mathcal{O}}(t-\tau, t) =& \{y : \exists u_j(\cdot) \in \mathbb{U}_j, \exists d_j(\cdot) \in \mathbb{D}_j, \\ & x_j(\cdot) \text{ satisfies (1)}, x_j(t-\tau) \in \mathcal{B}_j(t-\tau), \\ & x_j(t) = y\}.\end{aligned} \tag{43}$$

$\mathcal{W}_j^{\mathcal{O}}(t-\tau, t)$ represents the set of all possible states that Q_j can reach after a duration of τ starting from inside $\mathcal{B}_j(t-\tau)$. This FRS can be obtained by solving the HJ VI in (6) with the following Hamiltonian:

$$H_j^{\mathcal{O}}(x_j, \lambda) = \max_{u_j \in \mathcal{U}_j} \max_{d_j \in \mathcal{D}_j} \lambda \cdot f_j(x_j, u_j, d_j). \tag{44}$$

Since $\tau \in [0, t^{\text{IAT}}]$, the induced obstacles can be obtained as:

$$\begin{aligned}\mathcal{O}_i^j(t) &= \{x_i : \exists y \in \mathcal{P}_j(t), \|p_i - y\|_2 \leq R_c\} \\ \mathcal{P}_j(t) &= \{p_j : \exists h_j, (p_j, h_j) \in \bigcup_{\tau \in [0, t^{\text{IAT}}]} \mathcal{W}_j^{\mathcal{O}}(t-\tau, t)\}\end{aligned} \tag{45}$$

Note that by the definition of base obstacles, $\mathcal{B}_j(t+\tau_2) \subseteq \mathcal{W}_j^{\text{BO}}(t+\tau_1, t+\tau_2)\ \forall t, \tau_2 > \tau_1$, where $\mathcal{W}_j^{\text{BO}}(t+\tau_1, t+\tau_2)$ denotes the FRS of $\mathcal{B}_j(t+\tau_1)$ computed for a duration of $\tau_2 - \tau_1$. Therefore, we have that $\mathcal{W}_j^{\mathcal{O}}(t-\tau, t) \subseteq \mathcal{W}_j^{\mathcal{O}}(t - t^{\text{IAT}}, t^{\text{IAT}})\ \forall \tau \in [0, t^{\text{IAT}}]$. Thus, $\mathcal{P}_j(t)$ in (45) can be equivalently written as

$$\mathcal{P}_j(t) = \{p_j : \exists h_j, (p_j, h_j) \in \mathcal{W}_j^{\mathcal{O}}(t - t^{\text{IAT}}, t)\}. \tag{46}$$

2) Augmented Obstacle Computation: We next need to ensure that Q_i does not collide with the obstacles $\mathcal{G}_i(\cdot)$ computed in Section VI-A1 even when it is avoiding the intruder. In particular, we can compute a region around the obstacles $\mathcal{G}_i(\cdot)$ such that for all disturbances, Q_i can avoid colliding with obstacles for t^{IAT} seconds regardless of its avoidance control, if Q_i starts outside this region. Augmenting $\mathcal{G}_i(\cdot)$ with this region gives us the augmented obstacles, $\tilde{\mathcal{G}}_i(\cdot)$, that can then be used during the trajectory planning of Q_i to ensure collision avoidance with $\mathcal{G}_i(\cdot)$.

Suppose that the intruder appears in the system at some time time $\underline{t} = t - t^{\text{IAT}} + \tau, \tau \in [0, t^{\text{IAT}}]$. In this case, we need to ensure that Q_i does not collide with the obstacle $\mathcal{G}_i(t+\tau)$ at time $t+\tau$, regardless of its control $u_i(s)$ and disturbance $d_i(s)$ for the time interval $s \in [\underline{t}, t+\tau]$. It is, therefore, sufficient to avoid the τ-horizon BRS of $\mathcal{G}_i(t+\tau)$ at time t. This argument applies for all $\tau \in [0, t^{\text{IAT}}]$. Mathematically,

$$\tilde{\mathcal{G}}_i(t) = \bigcup_{\tau \in [0, t^{\text{IAT}}]} \mathcal{V}_i^{\mathcal{G}}(t, t+\tau) \tag{47}$$

where $\mathcal{V}_i^{\mathcal{G}}(t, t+\tau)$ represents BRS of $\mathcal{G}_i(t+\tau)$ computed backwards for τ seconds. Formally,

$$\begin{aligned}\mathcal{V}_i^{\mathcal{G}}(t, t+\tau) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, x_i(t) = y, \\ & \exists s \in [t, t+\tau], x_i(s) \in \mathcal{G}_i(s)\}.\end{aligned} \tag{48}$$

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

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

Remark 2: Note that if we use the robust trajectory tracking method to compute the base obstacles, we would need to augment the obstacles in (47) by the error bound of Q_i, Ω_i, as discussed in section V-A3.

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

$$\begin{aligned}\mathcal{V}_i^{\text{AO}}(t, t_i^{\text{STA}}) =& \{y : \exists u_i(\cdot) \in \mathbb{U}_i, \forall d_i(\cdot) \in \mathbb{D}_i, \\ & x_i(\cdot) \text{ satisfies (1)}, \forall s \in [t, t_i^{\text{STA}}], x_i(s) \notin \tilde{\mathcal{G}}_i(s), \\ & \exists s \in [t, t_i^{\text{STA}}], x_i(s) \in \mathcal{L}_i, x_i(t) = y\}.\end{aligned} \tag{50}$$

The Hamiltonian H_i^{AO} to compute BRS in (50) is given by:

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

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

$$u_i^{\text{AO}}(t, x_i) = \arg \min_{u_i \in \mathcal{U}_i} \max_{d_i \in \mathcal{D}_i} \lambda \cdot f_i(x_i, u_i, d_i) \tag{52}$$

Moreover, if Q_i starts within $\mathcal{V}_i^{\text{AO}}$, it is guaranteed to avoid collision for a duration of t^{IAT}, starting at any $\underline{t} < t_i^{\text{STA}}$, irrespective of the control and disturbance applied during this time period.

3) Optimal Avoidance Controller: First, we define relative dynamics of the intruder Q_I with state x_I with respect to Q_i with state x_i.

$$\begin{aligned}x_{I,i} &= x_I - x_i \\ \dot{x}_{I,i} &= f_r(x_{I,i}, u_i, u_I, d_i, d_I)\end{aligned} \tag{53}$$

Given the relative dynamics, we compute the set of states from which the joint states of Q_I and Q_i can enter danger zone $\mathcal{Z}_{iI}$ despite the best efforts of Q_i to avoid Q_I. This set of states is given by the BRS $\mathcal{V}^{\text{CA}}(t, t^{\text{IAT}}),\ t \in [0, t^{\text{IAT}}]$:

$$\begin{aligned}\mathcal{V}_i^{\text{CA}}(t, t^{\text{IAT}}) =& \{y : \forall u_i(\cdot) \in \mathbb{U}_i, \exists u_I(\cdot) \in \mathbb{U}_I, \exists d_i(\cdot) \in \mathbb{D}_i, \\ & \exists d_I(\cdot) \in \mathbb{D}_I, x_{I,i}(\cdot) \text{ satisfies (53)}, \\ & \exists s \in [t, t^{\text{IAT}}], x_{I,i}(s) \in \mathcal{L}_i^{\text{CA}}, x_{I,i}(t) = y\},\end{aligned} \tag{54}$$

where $\mathcal{L}_i^{\text{CA}} = \{x_{I,i} : \|p_{I,i}\|_2 \leq R_c\}$, and the Hamiltonian for computing this BRS is given by

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

Once the value function $V_i^{\text{CA}}(t,x_{I,i})$ corresponding to the BRS $\mathcal{V}_i^{\text{CA}}(t,t^{\text{IAT}})$ is computed, the optimal avoidance control u_i^{CA} can be obtained as:

$$u_i^{\text{CA}}(t,x_i,x_I)=\arg\max_{u_i\in\mathcal{U}_i}\Big(\min_{u_I\in\mathcal{U}_I,d_i\in\mathcal{D}_i,d_I\in\mathcal{D}_I}\lambda\cdot f_r(x_{I,i},u_i,u_I,d_i,d_I)\Big) \tag{55}$$

Under normal circumstances when the intruder Q_I is far away, we have $V_i^{\text{CA}}(0,x_{I,i})>0$; as Q_I gets closer to Q_i, $V_i^{\text{CA}}(0,x_{I,i})$ decreases. If Q_i applies the control u_i^{CA} when $V_i^{\text{CA}}(0,x_{I,i})=0$, then collision avoidance between Q_i and Q_I is guaranteed for a duration of t^{IAT} under the worst-case intruder control strategy.

In addition, obstacle augmentation (47) ensures that Q_i does not collide with $\mathcal{G}_i(\cdot)$ during the avoidance maneuver. The overall control policy for avoiding the intruder and collision with other vehicles is thus given by:

$$u_i^{\text{A}}(t)=\begin{cases}u_i^{\text{AO}}(t,x_i) & t\le\underline{t}\\ u_i^{\text{CA}}(t,x_i,x_I) & \underline{t}\le t\le\bar{t}\end{cases}$$

4) Replanning after intruder avoidance: After the intruder disappears, goal satisfaction controllers which ensure that the vehicles reach their destinations can be obtained by solving an STP problem as described in Section V, where the starting states of the vehicles are now given by the states they end up in, denoted $\tilde{x}_j^0$, after avoiding the intruder. Let the optimal control policy corresponding to this goal satisfaction 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 is given by:

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

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:

$$\begin{aligned}\mathcal{W}_i^{\text{RP}}(\bar{t},t)=&\{y\in\mathbb{R}^{n_i}:\exists u_i(\cdot)\in\mathbb{U}_i,\forall d_i(\cdot)\in\mathbb{D}_i,\\ &x_i(\cdot)\text{ satisfies (1)},x_i(\bar{t})=\tilde{x}_i^0,\\ &x_i(t)=y,\forall s\in[\bar{t},t],x_i(s)\notin\mathcal{G}_i^{\text{RP}}(s)\},\end{aligned} \tag{56}$$

where $\tilde{x}_i^0$ represents the state of Q_i at $t=\bar{t}$; $\mathcal{G}_i^{\text{RP}}(\cdot)$ takes into account the fact that Q_i needs to avoid higher-priority vehicles $Q_j, j<i$ and is defined in an way analogous to (9).

The FRS in (56) can be obtained by solving

$$\begin{aligned}&\max\Big\{D_tW^{\text{RP}}(t,x_i)+H_i^{\text{RP}}(t,x_i,\nabla W^{\text{RP}}(t,x_i)),\\ &\qquad-g^{\text{RP}}(t,x_i)-W^{\text{RP}}(t,x_i)\Big\}=0\\ &W^{\text{RP}}(\underline{t},x_i)=\max\{l^{\text{RP}}(x_i),-g^{\text{RP}}(\underline{t},x_i)\}\\ &H_i^{\text{RP}}(x_i,\lambda)=\max_{u_i\in\mathcal{U}_i}\min_{d_i\in\mathcal{D}_i}\lambda\cdot f_i(x_i,u_i,d_i)\end{aligned} \tag{57}$$

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

Remark 3: Note that we only need to re-plan the trajectories of the vehicles that are affected by the intruder. In particular, if $V^{\text{CA}}(0,x_{I,i}(t))>0$ during the entire duration $t\in[\underline{t},\bar{t}]$ for a vehicle, then the vehicle would need not to apply any avoidance control, and hence re-planning would not be required for this vehicle.

Remark 4: In general, an intruder can be present in the system for much longer than t^{IAT}, as long as it is not affecting the STP vehicles. $\underline{t}$ thus really corresponds to the time an intruder starts affecting a STP vehicle.

Remark 5: Note that even though we have presented the analysis for one intruder, the proposed method can handle multiple intruders as long as only one intruder is present *at any given time*.

We conclude this section with the overall STP algorithm that takes into account an intruder that may appear for a duration of t^{IAT}:

Algorithm 5: **Intruder Avoidance algorithm (offline planning)**: Given initial conditions x_i^0, vehicle dynamics (1), intruder dynamics in Assumption 2, target sets $\mathcal{L}_i$, and static obstacles $\mathcal{O}_i^{\text{static}}, i=1\ldots,N$, for each i,

1) determine the total obstacle set $\mathcal{G}_i(t)$, given in (9). In the case $i=1$, $\mathcal{G}_i(t)=\mathcal{O}_i^{\text{static}}\ \forall t$;
2) compute the augmented obstacle set $\tilde{\mathcal{G}}_i(t)$ given by (47), where $\mathcal{V}_i^{\mathcal{G}}(0,\tau)$ is given by (48);
3) given $\tilde{\mathcal{G}}_i(t)$, compute the BRS $\mathcal{V}_i^{\text{AO}}(t,t_i^{\text{STA}})$ defined in (50);
4) the optimal control to avoid the intruder can be obtained by computing $\mathcal{V}_i^{\text{CA}}(t,t^{\text{IAT}})$ in (54) and using (55);
5) the induced obstacles $\mathcal{O}_k^i(t)$ for each $k>i$ can be computed using (45).

Intruder Avoidance algorithm (online re-planning): For each vehicle i which performed avoidance in response to the intruder,

1) compute $\mathcal{W}_i^{\text{RP}}(\bar{t},t)$ using (56). The new t_i^{STA} for Q_i is given by $\arg\inf_t\{\mathcal{W}_j^{\text{RP}}(\bar{t},t)\cap\mathcal{L}_j\neq\emptyset\}$;
2) given t_i^{STA}, $\tilde{x}_i^0$, vehicle dynamics (1), target set $\mathcal{L}_i$, and static obstacles $\mathcal{O}_i^{\text{static}}, i=1\ldots,N$, use any of the three STP methods discussed in Section V for re-planning.

B. Numerical Simulations

To illustrate that our STP method is robust with respect to disturbances as well as a single intruder that is present for a duration of t^{IAT}, we use a five-vehicle example in which one of the five vehicles is an intruder. We assume that each vehicle has the dynamics given in (42). For this example, we chose the parameters $\underline{v}=0.1,\bar{v}=1,\bar{\omega}=1$, and disturbance bounds $d_r=0.1,\bar{d}_\theta=0.2$, which correspond to a 10% uncertainty in the dynamics.

The vehicles' initial states, scheduled times of arrival, and target sets are the same as those described in Section V-B, except that in this example, we have increased the target radius to $r = 0.15$. For illustrate purposes, we have chosen to use the robust trajectory tracking method described in Section V-A3 for the base obstacles' computation, and hence each vehicle tracks a nominal trajectory.

Fig. 12 shows the simulation at $t = \bar{t} = -2.39$, which corresponds to the time at which the intruder "disappears" from the domain. This time is chosen to maximally highlight the impact of the intruder. Here, the intruder is shown in black, and the STP vehicles are shown in the other different colors.

By the time $t = -2.39$, vehicle Q_2 (red) and vehicle Q_3 (green) have been avoiding the intruder for some time. This is evident from the amount of deviation between the actual positions of vehicles Q_2 and Q_3 (denoted by *) and their nominal positions (denoted by o) specified by the nominal trajectories they originally planned to track; these vehicles have abandoned nominal trajectory tracking in order to ensure safety with respect to the intruder. In contrast, Q_4 (magenta), which has not needed to avoid the intruder, is tracking its nominal trajectory very closely (but not exactly, due to the presence of disturbances).

The STP vehicles are rather far apart because a large margin is needed to ensure that they maintain separation even when multiple vehicles need to avoid the intruder. In this example in particular, the lowest-priority vehicle Q_4 needed to depart very early compared to Q_2 and Q_3 so that if an intruder were to arrive, Q_4 does not impede the ability of the other vehicles to perform avoidance. The early departure of Q_4 can be inferred from the fact that at $t = -2.39$, it is already nearly at its target.

For the same reason, the highest-priority vehicle Q_1 has not departed from its initial state yet, and thus is not shown at $t = -2.39$. Q_2 and Q_3 needed to depart very early compared to Q_1 to ensure sufficient margin for avoidance maneuvers.

Fig. 13 shows the nominal (black) and actual trajectories (red and green respectively) of vehicles Q_2 (top subplot) and Q_3 (bottom subplot). Specifically, the x and y positions over time are shown, and the black dotted vertical lines indicate the time interval in which the intruder is present. From Fig. 13, one can clearly see that before the intruder was present, both vehicles are able to track their nominal trajectories closely. When the intruder appears, the vehicles deviate from their nominal trajectories significantly. After the intruder disappears, both vehicles re-plan new trajectories, and at a later time, the resulting actual trajectories eventually arrive at the same location as the nominal trajectories.

In general, when all vehicles have different dynamics, the total computation time scales quadratically with the number of vehicles, because each vehicle must consider the BRSs (48) from the induced obstacle of higher-priority vehicles as discussed in Section VI-A1; the induced obstacles involve the FRS (43), which depends on the dynamics of the higher-priority vehicles. In our simulation, we assumed that all vehicles have the same dynamics, and thus the BRSs (48) are equivalent up to a coordinate transformation, resulting in a computation time that scales linearly with the number

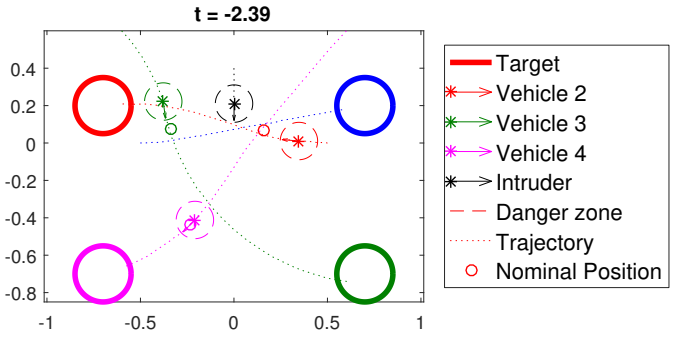

Fig. 12: The positions of the STP vehicles and the intruder at $t = -2.39$, the end of the intruder's appearance. The red and green vehicles Q_2, Q_3 have not been tracking their nominal trajectories for a while, and have been avoiding the intruder instead. Thus, their positions are far away from their nominal trajectories, indicated by the small colored circles. Q_4 has not needed to avoid the intruder, and tracks its nominal trajectory closely. The nominal trajectory of Q_4 allows it to stay far enough away from other vehicles so that all vehicles can remain safe in the presence of the intruder.

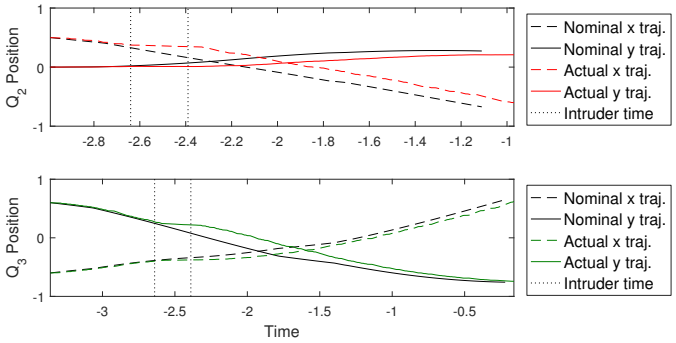

Fig. 13: The difference between the initially planned nominal trajectories and the actual trajectories for vehicles Q_2 (top subplot) and Q_3 (bottom subplot), which needed to perform avoidance with respect to the intruder during the time interval marked by the vertical black dotted lines. Before the intruder's presence, both vehicles track their nominal trajectories closely; however, both vehicles later deviate significantly from their nominal trajectories in order to avoid the intruder. After the intruder is gone, both vehicles replan their trajectories and arrive at their targets at a later time.

of vehicles. For the simulations shown, computations were done on a desktop computer with a Core i7 5820K processor and two GeForce GTX Titan X graphics processing units. The average computation time per vehicle is approximately 5 second using CUDA and GPU parallelization.

VII. Conclusions and Future Work

Guaranteed-safe multi-vehicle trajectory planning is a challenging problem, and previous analyses often either require strong assumptions on the motion of the vehicles or result in a large degree of conservatism. Optimal control and differential game techniques such as Hamilton-Jacobi (HJ) reachability are ideally suited for guaranteeing goal satisfaction and safety

under disturbances, but become intractable for even a small number of vehicles.

Our robust sequential trajectory planning (STP) method assigns a strict priority ordering to vehicles to offer a tractable and practical approach to the multi-vehicle trajectory planning problem. Under the proposed method, a portion of "space-time" is reserved for vehicles in the airspace in descending priority order to allow for dense vehicle configurations. Unlike previous priority-based methods, our approach accounts for disturbances and an adversarial intruder. STP reduces the scaling of HJ reachability's computational complexity from exponential to linear with respect to the number of vehicles, while maintaining hard guarantees on goal satisfaction and safety under disturbances. In the presence of a single intruder vehicle, STP still guarantees goal satisfaction and safety with a quadratically scaling computational complexity.

In the future, we plan to investigate ways of guaranteeing a maximum number of vehicles that need to re-plan, combine reachability analysis with other trajectory and path planning methods to improve computation speed, and to better understand the scenarios under which the STP scheme is the most useful by running large-scale simulations.

REFERENCES

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

[2] T. Prevot, J. Rios, P. Kopardekar, J. E. Robinson III, M. Johnson, and J. Jung, "UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations," in *16th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics, Jun. 2016, pp. 1–16.

[3] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, Jul. 1998.

[4] J. van den Berg, Ming Lin, and D. Manocha, "Reciprocal Velocity Obstacles for real-time multi-agent navigation," in *International Conference on Robotics and Automation*, May 2008, pp. 1928–1935.

[5] A. Wu and J. P. How, "Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles," *Autonomous Robots*, vol. 32, no. 3, pp. 227–242, Apr. 2012.

[6] R. Olfati-Saber and R. M. Murray, "DISTRIBUTED COOPERATIVE CONTROL OF MULTIPLE VEHICLE FORMATIONS USING STRUCTURAL POTENTIAL FUNCTIONS," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 495–500, 2002.

[7] Y. L. Chuang, Y. R. Huang, M. R. D'Orsogna, and A. L. Bertozzi, "Multi-vehicle flocking: Scalability of cooperative control algorithms using pairwise potentials," in *International Conference on Robotics and Automation*, Apr. 2007, pp. 2292–2299.

[8] Feng-Li Lian and R. Murray, "Real-time trajectory generation for the cooperative path planning of multi-vehicle systems," in *Conference on Decision and Control*, vol. 4, Dec. 2002, pp. 3766–3769.

[9] A. Ahmadzadeh, N. Motee, A. Jadbabaie, and G. Pappas, "Multi-vehicle path planning in dynamically changing environments," in *International Conference on Robotics and Automation*, May 2009, pp. 2449–2454.

[10] J. Bellingham, M. Tillerson, M. Alighanbari, and J. How, "Cooperative path planning for multiple UAVs in dynamic and uncertain environments," in *Conference on Decision and Control*, vol. 3, Dec. 2002, pp. 2816–2822.

[11] R. Beard and T. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *Conference on Decision and Control*, vol. 1, 2003, pp. 25–30.

[12] T. Schouwenaars and E. Feron, "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," in *Guidance, Navigation, and Control Conference and Exhibit*, Aug. 2004, pp. 2004–5141.

[13] D. M. Stipanovic, P. F. Hokayem, M. W. Spong, and D. D. Siljak, "Cooperative Avoidance Control for Multiagent Systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, p. 699, 2007.

[14] M. Massink and N. De Francesco, "Modelling free flight with collision avoidance," in *International Conference on Engineering of Complex Computer Systems*, 2001, pp. 270–279.

[15] M. Althoff and J. M. Dolan, "Set-based computation of vehicle behaviors for the online verification of autonomous vehicles," in *International Conference on Intelligent Transportation Systems*, Oct. 2011, pp. 1162–1167.

[16] Y. Lin and S. Saripalli, "Collision avoidance for UAVs using reachable sets," in *International Conference on Unmanned Aircraft Systems*, Jun. 2015, pp. 226–235.

[17] E. Lalish, K. A. Morgansen, and T. Tsukamaki, "Decentralized reactive collision avoidance for multiple unicycle-type vehicles," in *American Control Conference*, Jun. 2008, pp. 5055–5061.

[18] G. M. Hoffmann and C. J. Tomlin, "Decentralized cooperative collision avoidance for acceleration constrained vehicles," in *Conference on Decision and Control*, 2008, pp. 4357–4363.

[19] M. Chen, C.-Y. Shih, and C. J. Tomlin, "Multi-Vehicle Collision Avoidance via Hamilton-Jacobi Reachability and Mixed Integer Programming," in *Conference on Decision and Control*, 2016.

[20] E. N. Barron, "Differential games with maximum cost," *Nonlinear Analysis*, vol. 14, no. 11, pp. 971–989, Jun. 1990.

[21] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, Jul. 2005.

[22] O. Bokanowski, N. Forcadel, and H. Zidani, "Reachability and Minimal Times for State Constrained Nonlinear Problems without Any Controllability Assumption," *Journal on Control and Optimization*, vol. 48, no. 7, pp. 4292–4316, Jan. 2010.

[23] O. Bokanowski and H. Zidani, "MINIMAL TIME PROBLEMS WITH MOVING TARGETS AND OBSTACLES," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 2589–2593, Jan. 2011.

[24] K. Margellos and J. Lygeros, "HamiltonJacobi Formulation for ReachAvoid Differential Games," *Transactions on Automatic Control*, vol. 56, no. 8, pp. 1849–1861, Aug. 2011.

[25] J. F. Fisac, M. , C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *International Conference on Hybrid Systems Computation and Control*, 2015, pp. 11–20.

[26] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts." *National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996.

[27] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2006.

[28] I. M. Mitchell, "Application of Level Set Methods to Control and Reachability Problems in Continuous and Hybrid Systems," Ph.D. dissertation, Stanford University, 2002.

[29] "A toolbox of level set methods," Tech. Rep., 2007.

[30] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *Conference on Decision and Control*, 2008, pp. 3706–3712.

[31] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *International Conference on Robotics and Automation*, 2011, pp. 1451–1456.

[32] A. M. Bayen, I. M. Mitchell, M. K. Osihi, and C. J. Tomlin, "Aircraft Autolander Safety Analysis Through Optimal Control-Based Reach Set Computation," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 68–77, Jan. 2007.

[33] M. Erdmann and T. Lozano-Pérez, "On multiple moving objects," *Algorithmica*, vol. 2, no. 1-4, pp. 477–521, Nov. 1987.

[34] J. van den Berg and M. Overmars, "Prioritized motion planning for multiple robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 430–435.

[35] N. L. Earl A. Coddington, *Theory of ordinary differential equations*. R.E. Krieger, 1955.

[36] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," 2017.

[37] M. Chen, J. F. Fisac, S. Sastry, and C. J. Tomlin, "Safe sequential path planning of multi-vehicle systems via double-obstacle Hamilton-Jacobi-Isaacs variational inequality," in *European Control Conference*, Jul. 2015, pp. 3304–3309.

[38] S. Bansal, M. Chen, J. F. Fisac, and C. J. Tomlin, "Safe Sequential Path Planning of Multi-Vehicle Systems Under Presence of Disturbances and Imperfect Information," in *American Control Conference*, 2017.