Guaranteeing Safety and Liveness of Unmanned Aerial Vehicle Platoons on Air Highways

Mo Chen

(TO BE UPDATED!)
Recently, there has been immense interest in using unmanned aerial vehicles (UAVs) for civilian operations such as package delivery, firefighting, and fast disaster response. As a result, UAV traffic management systems are needed to support potentially thousands of UAVs flying simultaneously in the airspace, in order to ensure their liveness and safety requirements are met. Hamilton-Jacobi (HJ) reachability is a powerful framework for providing conditions under which these requirements can be met, and for synthesizing the optimal controller for meeting them. However, due to the curse of dimensionality, HJ reachability is only tractable for a small number of vehicles if their set of maneuvers is unrestricted. In this paper, we define a platoon to be a group of UAVs in a single-file formation. We model each vehicle as a hybrid system with modes corresponding to its role in the platoon, and specify the set of allowed maneuvers in each mode to make the analysis tractable. We propose several liveness controllers based on HJ reachability, and wrap a safety controller, also based on HJ reachability, around the liveness controllers. For a single altitude range, our approach guarantees safety for one safety breach; in the unlikely event of multiple safety breaches, safety can be guaranteed over multiple altitude ranges. We demonstrate the satisfaction of liveness and safety requirements through simulations of three common scenarios.

Nomenclature

(Nomenclature entries should have the units identified)

x	=	System state
$p = (p_x, p_y)$	=	Horizontal position
$v = (v_x, v_y)$	=	Horizontal velocity

I. Introduction

Unmanned aerial vehicle (UAV) systems have in the past been mainly used for military operations [1]. Recently, however, there has been an immense surge of interest in using UAVs for civil applications. Through projects such as Amazon Prime Air [2] and Google Project Wing [3], companies are looking to send UAVs into the airspace to not only deliver commerical packages, but also for important tasks such as aerial surveillance, emergency supply delivery, and . As a rough estimate, suppose in a city of 2 million people, each person requests a drone delivery every 2 months on average and each delivery requires a 30 minute trip for the drone. This would equate to thousands of vehicles simultaneously in the air. As a result, government agencies such as the Federal Aviation Administration (FAA) and National Aeronautics and Administration (NASA) of the United States are also investigating air traffic control for autonomous vehicles in order to prevent collisions among potentially numerous UAVs [4]. Applications of UAVs extend beyond package delivery; they can also be used, for example, to provide supplies or to firefight in areas that are difficult to reach but require prompt response [5].

Optimal control and game theory are powerful tools for providing liveness and safety guarantees to controlled dynamical systems under disturbances, and various formulations [6, 7, 8] have been successfully used to analyze problems involving a small number of vehicles [9, 10, 11]. These formulations are based on Hamilton-Jacobi (HJ) reachability, which computes the reachable set, defined as the set of states from which a system is guaranteed to have a control strategy to reach a target set of states. Reachability is a powerful tool because reachable sets can be used for synthesizing both controllers that steer the system towards a set of goal states (liveness controllers), and controllers that steer the system away from a set of unsafe states (safety controllers). Furthermore, the HJ formulations are flexible in terms of system dynamics, enabling the analysis of non-linear systems. The power and success of HJ reachability analysis in previous applications cannot be denied, especially since numerical tools are readily available to solve the associated HJ Partial Differential Equation (PDE) [12, 13, 14]. However, the computation is done on a grid, making the problem complexity scale exponentially with the number of states, and therefore with the number of vehicles. This makes the computation intractable for large numbers of vehicles.

A considerable body of work has been done on the platooning of vehicles [15]. For example, [16] investigated the feasibility of vehicle platooning in terms of tracking errors in the presence of disturbances, taking into account complex nonlinear dynamics of each vehicle. [17] explored several control techniques for performing various platoon maneuvers such as lane changes, merge procedures, and split procedures. In [18], the authors modeled vehicles in platoons as hybrid systems, synthesized safety controllers, and analyzed throughput. Finally, reachability analysis was used in [19] to analyze a platoon of two trucks in order to minimize drag by minimizing the following distance while maintaining collision avoidance safety guarantees.

Previous analysis of a large number of vehicles typically do not provide liveness and safety guarantees to the extent that HJ reachability does; however, HJ reachability typically cannot be used to tractably analyze a large number of vehicles. In this paper, we attempt to reconciliate this trade-off by assuming a single-file platoon, which provides structure that allows pairwise safety guarantees from HJ reachability to translate to safety guarantees for the whole platoon. We first propose a hybrid systems model of UAVs in platoons to capture this structure. Then, we show how HJ reachability can be used to synthesize *liveness controllers* that enable vehicles to reach a set of desired states, and wrap *safety controllers* around the liveness controllers in order to prevent dangerous configurations such as collisions. Finally, we show simulation results of quadrotors forming a platoon, a platoon responding to a malfunctioning member, and a platoon responding to an outside intruder to illustrate the behavior of vehicles in these scenarios and demonstrate the guarantees provided by HJ reachability.

II. Air Highways

Air highways are virtual highways in the airspace on which a number of UAV platoons may be present. UAVs seek to arrive at some desired destination starting from their origin by traveling along a sequence of air highways. Air highways are intended to be the common pathways for many UAV platoons, whose members may have different origins and destinations. By routing platoons of UAVs onto a few common pathways, the air space becomes more tractable to analyze and intuitive to monitor. The concept of platoons will be proposed in Section III; for now, we will focus on air highways.

Let an air highway be denoted by the function $\mathbb{H} : [0, 1] \rightarrow \mathbb{R}^2$. Such a highway lies in a horizontal plane of fixed altitude, with start and end points given by $\mathbb{H}(0) \in \mathbb{R}^2$ and $\mathbb{H}(1) \in \mathbb{R}^2$ respectively. For simplicity, we assume that the highway segment is a straight line segment, and the parameter s indicates the position in some fixed altitude as follows: $\mathbb{H}(s) = \mathbb{H}(0) + s(\mathbb{H}(1) - \mathbb{H}(0))$. To each highway, we assign a speed of travel $v_{\mathbb{H}}$ and specify the direction of travel to be the direction from $\mathbb{H}(0)$ to $\mathbb{H}(1)$, denoted using a unit vector $\hat{d} = \frac{\mathbb{H}(1)-\mathbb{H}(0)}{\|\mathbb{H}(1)-\mathbb{H}(0)\|_2}$.

Air highways must not only provide structure to make the analysis of a large number of vehicles tractable, but also allow vehicles reach their destinations while minimizing any relevant costs to the vehicles and to the surrounding regions. Thus, given a origin-destination pair (eg. two cities), air highways must connect the two points while potentially satisfying other criteria. We now define the air highway placement problem, and propose a simple and fast way to approximate its solution.

A. The Air Highway Placement Problem

Consider a map $c : \mathbb{R}^2 \to \mathbb{R}$ which defines the cost $c(p)$ incurred when a UAV flies over the position $p = (p_x, p_y) \in \mathbb{R}^2$. Given any p, a large value of $c(p)$ indicates that the position p is costly or undesirable for a UAV to fly over. Locations with a high cost could be In general, the cost map $c(\cdot)$ may be used to model cost of interference with commercial air spaces, cost of accidents, cost of noise pollution, etc., and can be designed by government regulation bodies.

Let p^o denote an origin point and p^d denote a destination point. Consider a sequence of highways $\mathbb{S}_N = \mathbb{H}_1, \mathbb{H}_2, \dots, \mathbb{H}_N$ that satisfies the following:

$$\begin{aligned} \mathbb{H}_1(0) &= p^o \\ \mathbb{H}_i(1) &= \mathbb{H}_{i+1}(0), i = 0, 1, \dots, N-1 \\ \mathbb{H}_N(1) &= p^d \end{aligned} \tag{1}$$

The interpretation of the above conditions is that the start point of first highway is the origin, the end point of a highway is the end point of the next highway, and the end point of last highway is the destination. The highways $\mathbb{H}_1, \dots, \mathbb{H}_N$ form a sequence of waypoints for a UAV starting at the origin p^o to reach its destination p^d.

Given only the origin point p^o and destination point p^d, there are an infinite number of choices for a sequence of highways that satisfy (1). However, if one takes into account the cost of flying over any position p using the cost map $c(\cdot)$, we arrive at the air highway placement problem:

$$\begin{aligned} &\min_{\mathbb{S}_N, N} \sum_{i=1}^{N} \int_0^1 c(\mathbb{H}_i(s))ds \\ &\text{subject to (1)} \end{aligned} \tag{2}$$

In other words, we consider air highways to be line segments of constant altitude over a region, and UAV platoons travel on these air highways to get from some origin to some destination. Any UAV flying on a highway over some position p incurs a cost of $c(p)$, so that the total cost of flying from the origin to the destination is given by (2). The air highway placement problem minimizes the cumulative cost of flying from some origin p^o to some destination p^d along the sequence of highways $\mathbb{S}_N$.

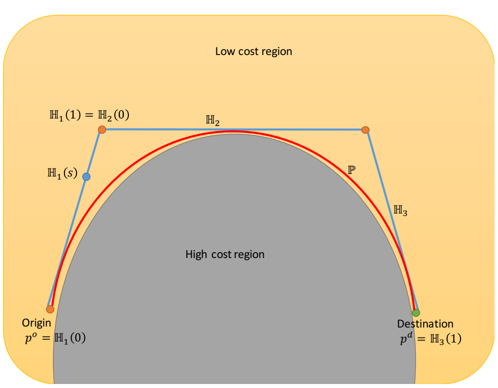


Figure 1. Highway illustration

B. The Eikonal Equation – Cost-Minimizing Path

Let $s_0, s_1 \in \mathbb{R}$, and let $\mathbb{P} : [s_0, s_1] \to \mathbb{R}^2$ be a path starting from an origin point $p^o = \mathbb{P}(s_0)$ and ending at a destination point $p^d = \mathbb{P}(s_1)$. Note that the sequence $\mathbb{S}_N$ in (2) is a piece-wise affine example of a path $\mathbb{P}(s), s \in [s_0, s_1]$; however, a path $\mathbb{P}$ that is not piece-wise affine cannot be written as a sequence of highways $\mathbb{S}_N$.

More concretely, suppose a UAV flies from an origin point p^o to a destination point p^d along some path $\mathbb{P}(s)$ parametrizes by the parameters s. Then, $\mathbb{P}(s_0) = p^o$ would denote the origin, and $\mathbb{P}(s_1) = p^d$ would denote the destination. All intermediate s values denote the intermediate positions of the path, i.e. $\mathbb{P}(s) = p(s) = (p_x(s), p_y(s))$.

Consider the cost map $c(p_x, p_y)$ which captures the cost incurred for UAVs flying over the position $p = (p_x, p_y)$. Along the entire path $\mathbb{P}(s)$, the cumulative cost $C(\mathbb{P})$ is incurred. Define C as follows:

$$C(\mathbb{P}) = \int_{s_0}^{s_1} c(\mathbb{P}(s))ds \tag{3}$$

The problem of finding the cost-minimizing path is finding the path such that the above cost is minimized. More generally, given an origin point p^o, we would like to compute the function V representing optimal cumulative cost for any destination point p:

$$\begin{aligned} V(p^d) &= \min_{\mathbb{P}(\cdot)} C(\mathbb{P}) \\ &= \min_{\mathbb{P}(\cdot)} \int_{s_0}^{s_1} c(\mathbb{P}(s))ds \end{aligned} \tag{4}$$

It is well known [] that the solution to the Eikonal equation (5) precisely computes the function $V(p^d)$ given the cost map c. Note that a single function characterizes the minimum cost from an origin p^o to any destination p^d. Once V is found, the optimal path $\mathbb{P}$ between p^o and p^d can be obtained via gradient descent.

$$\begin{aligned} c(p)|\nabla V(p)| &= 1 \\ V(p^o) &= 0 \end{aligned} \tag{5}$$

(5) can be efficiently computed numerically using the fast marching method [].

Note that (4) can be viewed as a relaxation of the air highway placement problem defined in (2). Unlike (2), the relaxation (4) can be quickly solved using currently available numerical tools. Thus, we first solve (4), and then post-process the solution to (4) to obtain an approximation to (2).

Given a single origin point p^o, the optimal cumulative cost function $V(p)$ can be computed. Suppose M different destination points $p_i^d, i = 1, \dots, M$ are chosen. Then, M different optimal paths $\mathbb{P}_i, i = 1, \dots, M$ are obtained from V.

C. From Paths to Waypoints

Each of the cost-minimizing paths $\mathbb{P}_i$ computed from the solution to the Eikonal equation consists of a continuous set of points. Each path $\mathbb{P}_i$ is an approximation to the sequence of highways $\mathbb{S}_{N_i}^i = \{\mathbb{H}_j^i\}_{i=1,j=1}^{i=M,j=N_i}$ defined in (2), but now indexes by the corresponding path.

For each path $\mathbb{P}_i$, we would like to sparsify the points on the path to obtain a collection of waypoints, $\mathcal{W}_{i,j}, j = 1, \dots, N_i + 1$, which are the end points of the highways:

$$\begin{aligned} \mathbb{H}_j^i(0) &= \mathcal{W}_{i,j}, \\ \mathbb{H}_j^i(1) &= \mathcal{W}_{i,j+1}, \\ j &= 1, \dots, N_i \end{aligned} \tag{6}$$

we sparsify this set of points to obtain a collection of waypoints, $\mathcal{W}$, which are also endpoints of air highways, we first add the destination point to the collection and note the path's heading. Next, we add to the collection of waypoints the first point on the path at which the heading changes by some threshold θ_C. This process is repeated along the entire path. Finally, we add the origin points to the collection. We create a collection of these points for all the cost-minimizing paths.

If there is a large change in heading within a small section of the cost-minimizing path, then the collections $\mathcal{W}_{ij}$ may contain many points which are close together. In addition, there may be multiple paths that are very close to each other (in fact, this behavior is desirable), also cluttering the airspace with too many waypoints. We propose to sparsify the waypoints by clustering the points in $\mathcal{W}_{ij}$. Each cluster contains points that are within a certain distance to the closest point in the cluster, and all points in each cluster are replaced with a single point located at the centroid of the cluster.

D. Results

To illustrate our air highway placement proposal, we used the San Francisco Bay Area as an example, and classified each point on the map into three different region. Each region has an associated cost, reflecting the desirability of flying a vehicle over an area in the region. In general, these costs can be arbitrary and determined by government regulation agencies. For illustration purposes, we assumed the following categories and costs:

- Region around airports: $c_{\text{airports}} = b$,
- Cities: $c_{\text{cities}} = 1$,
- Water: $c_{\text{water}} = b^{-2}$,
- Other: $c_{\text{other}} = b^{-1}$.

This assumption assigns costs in descending order to the categories "region around airports", "cities", "water", and "other". Flying a UAV in each category is more costly by a factor of b compared to the next most important category. The factor b is a tuning parameter that we adjusted to vary the relative importance of the different categories, and we used $b = 4$ in the figures below.

1. *Cost-Minimizing Paths*

Fig. 2 shows the San Francisco Bay Area (geographic) map, cost map, cost-minimizing paths, and contours of the value function. The region enclosed by the black boundary represent "regoin around airports", which have the highest cost. The dark blue, yellow, and light blue regions represent the "cities", the "water", and the "other"' categories, respectively. We assumed that the origin corresponds to the city "Concord", and chose a number of other major cities as destinations.

A couple of important observations can be made here. First, the cost-minimizing paths to the various destinations in general overlap, and only split up when they are very close to entering their destination cities. This is intuitively desirable because the number of air highways is kept low. In addition, the cost of flying in the airspace according to the cost map is minimized. Secondly, the spacing of the contours, which correspond to level curves of the value function, have a spacing corresponding to the cost map. This provides insight into the placement of air highways to destinations that were not shown in this example.

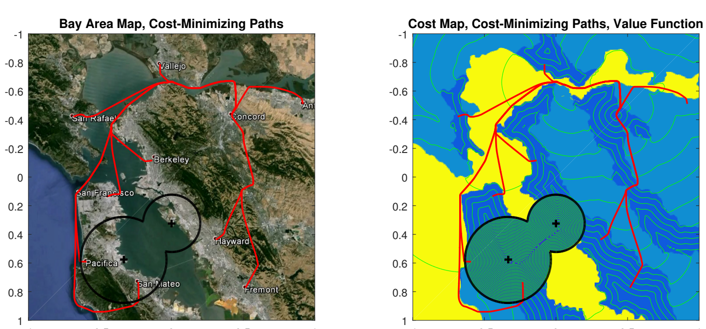


Figure 2. Cost-minimizing paths computed by the Fast Marching Method based on the assumed cost map of the San Francisco Bay Area

Fig. 3 shows the result of converting the cost-minimizing paths to a small number of waypoints. The left subfigure shows

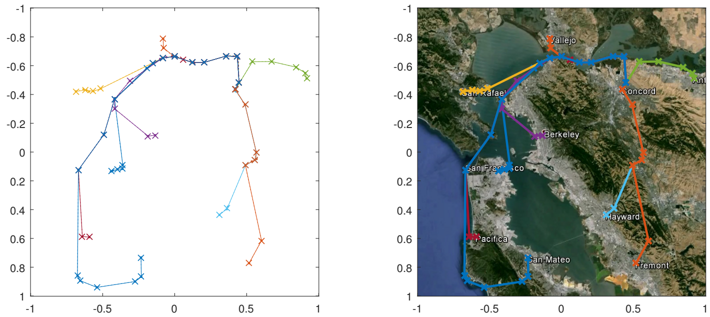


Figure 3. Results of conversion from cost-minimizing paths to highway way points.

III. Unmanned Aerial Vehicle Platooning

A. Problem Formulation

1. *Vehicle Dynamics*

Consider a UAV whose dynamics are given by

$$\dot{x} = f(x, u) \tag{7}$$

where x represents the state, and u represents the control action. The techniques we present in this paper do not depend on the dynamics of the vehicles. However, for concreteness, we consider assume that the UAVs are quadrotors that fly at a constant altitude under non-faulty circumstances. For the quadrotor, we use a simple model in which the x and y dynamics are double integrators:

$$\begin{aligned}\dot{p}_x &= v_x \\ \dot{p}_y &= v_y \\ \dot{v}_x &= u_x \\ \dot{v}_y &= u_y \\ |u_x|, |u_y| &\leq u_{\max}\end{aligned} \tag{8}$$

where the state $x = (p_x, v_x, p_y, v_y) \in \mathbb{R}^4$ represents the quadrotor's position in the x-direction, its velocity in the x-direction, and its position and velocity in the y-direction, respectively. The control input $u = (u_x, u_y) \in \mathbb{R}^2$ consists of the acceleration in the x- and y- directions. For convenience, we will denote the position and velocity $p = (p_x, p_y), v = (v_x, v_y)$, respectively. We will consider a group of N quadrotors $Q_i, i = 1 \ldots, N$.

In general, the problem of collision avoidance among N vehicles cannot be tractably solved using traditional dynamic programming approaches because the computation complexity of these approaches scales exponentially with the number of vehicles. Thus, in our present work, we will consider the situation where UAVs travel on air highways in platoons, defined in the following sections. The structure imposed by air highways and the platoon enables us to analyze the liveness and safety of the vehicles in a tractable manner.

2. *Vehicles as Hybrid Systems*

We model each vehicle as a hybrid system [18, 20] consisting of the modes "Free", "Leader", "Follower", and "Faulty". Within each mode, a set of maneuvers is available to allow the vehicle to change modes if desired. The modes and maneuvers are as follows:

- Free:

  A Free vehicle is not in a platoon or on a highway, and its possible maneuvers or mode transitions are

  - remain a Free vehicle by staying away from highways,
  - become a Leader by entering a highway to create a new platoon, and
  - become a Follower by joining a platoon that is currently on a highway.

- Leader: A Leader vehicle is the vehicle at the front of a platoon (which could consist of only the vehicle itself). The available maneuvers and mode transitions are
  - remain a Leader by traveling along the highway at a pre-specified speed,
  - become a Follower by merging the current platoon with a platoon in front, and
  - become a Free vehicle by leaving the highway.
- Follower:

  A Follower vehicle is a vehicle that is following a platoon leader. The Available maneuvers and mode transitions are
  - remain a Follower by staying a distance of b behind the vehicle in front in the current platoon,
  - become a Leader by splitting from the current platoon, and
  - become a Free vehicle by leaving the highway.
- Faulty:

  If a vehicle from any of the other modes malfunction, it transitions into the Faulty mode and descends after a duration of t_{internal}.

The available maneuvers and associated mode transitions are summarized in Figure 4.

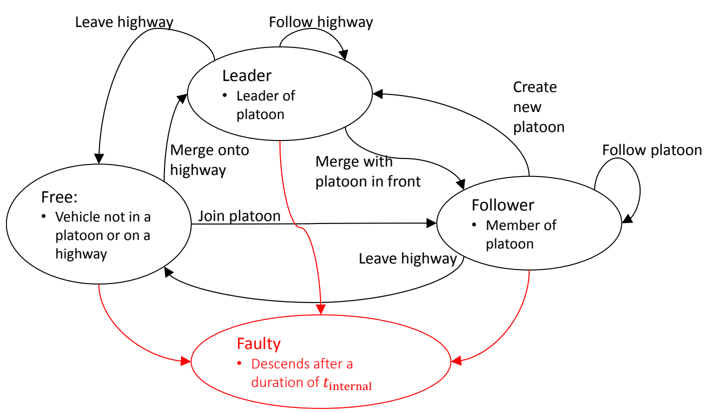


Figure 4. Hybrid modes for vehicles in platoons.

Suppose that there are N vehicles in total. We consider a platoon of vehicles to be a group of $M \leq N$ vehicles, denoted $Q_{P_1}, \dots, Q_{P_M}, \{P_j\}_{j=1}^M \subseteq \{i\}_{i=1}^N$, in a single-file formation. We will assume that the vehicles in a platoon travel along an air highway. The vehicles maintain a separation distance of b with its neighbors inside the platoon. In order to allow for close proximity of the vehicles and the ability to resolve multiple simultaneous safety breaches, we assume that in the event of a malfunction, a vehicle will be able to exit the altitude range of the highway within a duration of $t_{\text{internal}} = 1.5$. Such a requirement may be implemented practically as an emergency landing procedure to which the vehicles revert when a malfunction is detected.

3. *Objectives*

Given the above modeling assumptions, our goal is to provide control strategies to guarantee the success and safety of all the mode transitions. The theoretical tool used to provide the liveness and safety guarantees is reachability. The reachable sets we compute will allow each vehicle to perform complex actions such as

SHOULD EVASION BE A MODE?

- merge onto a highway to form a platoon,
- join a new or different platoon,
- leave a platoon to create a new one,
- reacting to a malfunctioning or intruder vehicles.

We also propose more basic controllers to perform other simpler actions such as

- following the highway at constant altitude at a specified speed,
- maintaining a constant relative position and velocity with the leader of a platoon.

In general, the control strategies of each vehicle have a liveness component, which specifies a set of states towards which the vehicle aims to reach, and a safety component, which specifies a set of states that it must avoid. Together, the liveness and safety controllers guarantee the success and safety of a vehicle in the airspace making any desired mode transition. In this paper, these guarantees are provided using reachability analysis, and allow the multi-UAV system to perform joint maneuvers essential to maintaining structure in the airspace.

B. Hamilton-Jacobi Reachability

1. *General Framework*

Consider a differential game between two players described by the system

$$\dot{x} = f(x, u_1, u_2), \text{for almost every } t \in [-T, 0] \tag{9}$$

where $x \in \mathbb{R}^n$ is the system state, $u_1 \in \mathcal{U}_1$ is the control of Player 1, and $u_2 \in \mathcal{U}_2$ is the control of Player 2. We assume $f : \mathbb{R}^n \times \mathcal{U}_1 \times \mathcal{U}_2 \to \mathbb{R}^n$ is uniformly continuous, bounded, and Lipschitz continuous in x for fixed u_1, u_2, and the control functions $u_1(\cdot) \in \mathbb{U}_1, u_2(\cdot) \in \mathbb{U}_2$ are drawn from the set of measurable functions[a]. Player 2 is allowed to use nonanticipative strategies [21, 22] γ, defined by

$$\begin{aligned}\gamma \in \Gamma := \{&\mathcal{N} : \mathbb{U}_1 \to \mathbb{U}_2 \mid u_1(r) = \hat{u}_1(r) \\ &\text{for almost every } r \in [t, s] \Rightarrow \mathcal{N}[u_1](r) \\ &= \mathcal{N}[\hat{u}_1](r) \text{ for almost every } r \in [t, s]\}\end{aligned} \tag{10}$$

In our differential game, the goal of Player 2 is to drive the system into some target set $\mathcal{L}$, and the goal of Player 1 is to drive the system away from it. The set $\mathcal{L}$ is represented as the zero sublevel set of a bounded, Lipschitz continuous function $l : \mathbb{R}^n \to \mathbb{R}$. We call $l(\cdot)$ the *implicit surface function* representing the set $\mathcal{L} : \mathcal{L} = \{x \in \mathbb{R}^n \mid l(x) \leq 0\}$.

Given the dynamics (9) and the target set $\mathcal{L}$, we would like to compute the backwards reachable set, $\mathcal{V}(t)$:

$$\begin{aligned}\mathcal{V}(t) := \{&x \in \mathbb{R}^n \mid \exists \gamma \in \Gamma \text{ such that} \forall u_1(\cdot) \in \mathbb{U}_1, \\ &\exists s \in [t, 0], \xi_f(s; t, x, u_1(\cdot), \gamma[u_1](\cdot)) \in \mathcal{L}\}\end{aligned} \tag{11}$$

where ξ_f is the trajectory of the system satisfying initial conditions $\xi_f(t; x, t, u_1(\cdot), u_2(\cdot)) = x$ and the following differential equation almost everywhere on $[-t, 0]$

$$\begin{aligned}&\frac{d}{ds}\xi_f(s; x, t, u_1(\cdot), u_2(\cdot)) \\ &\quad = f(\xi_f(s; x, t, u_1(\cdot), u_2(\cdot)), u_1(s), u_2(s))\end{aligned} \tag{12}$$

Many methods involving solving HJ PDEs [7] and HJ variational inequalities (VI) [6, 8, 9] have been developed for computing the reachable set. These HJ PDEs and HJ VIs can be solved using well-established numerical methods. For this paper, we use the formulation in [7], which has shown that the backwards reachable set $\mathcal{V}(t)$ can be obtained as the zero sublevel set of the viscosity solution [23] $V(t, x)$ of the following terminal value Hamilton-Jacobi-Isaacs (HJI) PDE:

[a] 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.

$$\begin{aligned}&D_t V(t,x)+\\&\quad \min\{0, \max_{u_1\in\mathcal{U}_1}\min_{u_2\in\mathcal{U}_2} D_x V(t,x)\cdot f(x,u_1,u_2)\} = 0,\\&\quad V(0,x) = l(x)\end{aligned} \tag{13}$$

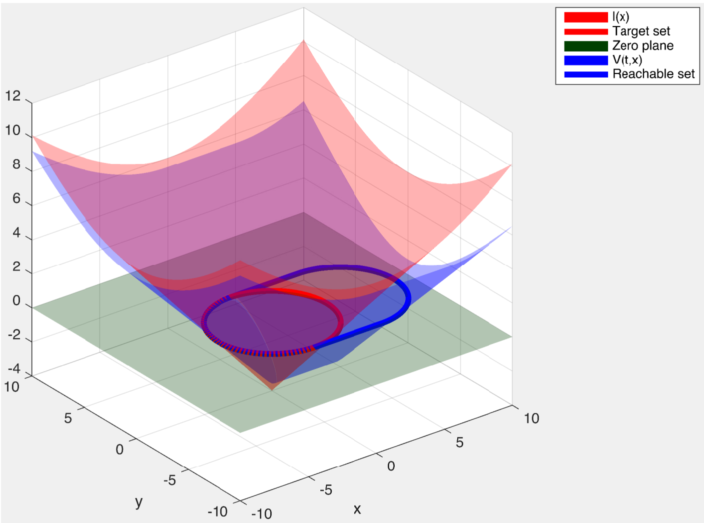


Figure 5. Illustration of a target set, reachable set, and their implicit surface functions.

from which we obtain $\mathcal{V}(t) = \{x \in \mathbb{R}^n \mid V(t,x) \leq 0\}$. From the solution $V(t,x)$, we can also obtain the optimal controls for both players via the following:

$$\begin{aligned}u_1^*(t,x) &= \arg\max_{u_1\in\mathcal{U}_1}\min_{u_2\in\mathcal{U}_2} D_x V(t,x)\cdot f(x,u_1,u_2)\\u_2^*(t,x) &= \arg\min_{u_2\in\mathcal{U}_2} D_x V(t,x)\cdot f(x,u_1^*,u_2)\end{aligned} \tag{14}$$

In the special case where there is only one player, we obtain an optimal control problem for a system with dynamics

$$\dot{x} = f(x,u), t\in[-T,0], u\in\mathcal{U}. \tag{15}$$

The reachable set in this case would be given by the Hamilton-Jacobi-Bellman (HJB) PDE

$$\begin{aligned}D_t V(t,x) + \min\{0, \min_{u\in\mathcal{U}} D_x V(t,x)\cdot f(x,u)\} = 0\\V(0,x) = l(x)\end{aligned} \tag{16}$$

where the optimal control is given by

$$u^*(t,x) = \arg\min_{u\in\mathcal{U}} D_x V(t,x)\cdot f(x,u) \tag{17}$$

For our application, we will use a several decoupled system models and utilize the decoupled HJ formulation in [24], which enables real time 4D reachable set computations and tractable 6D reachable set computations.

2. *Relative Dynamics and Augmented Relative Dynamics*

Besides Equation (9), we will also consider the relative dynamics between two quadrotors Q_i, Q_j. These dynamics can be obtained by defining the relative variables

$$\begin{aligned}p_{x,r} &= p_{x,i} - p_{x,j}\\p_{y,r} &= p_{y,i} - p_{y,j}\\v_{x,r} &= v_{x,i} - v_{x,j}\\v_{y,r} &= v_{y,i} - v_{y,j}\end{aligned} \tag{18}$$

We treat Q_i as Player 1, the evader who wishes to avoid collision, and we treat Q_j as Player 2, the pursuer, or disturbance, that wishes to cause a collision. In terms of the relative variables given in (18), we have

$$\begin{aligned}\dot{p}_{x,r} &= v_{x,r}\\\dot{p}_{y,r} &= v_{y,r}\\\dot{v}_{x,r} &= u_{x,i} - u_{x,j}\\\dot{v}_{y,r} &= u_{y,i} - u_{y,j}\end{aligned} \tag{19}$$

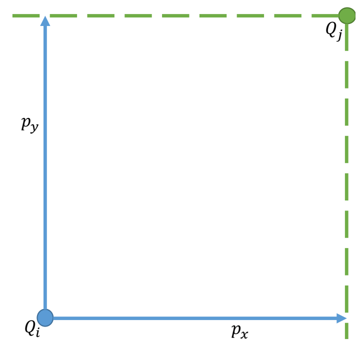


Figure 6. Relative coordinates

We also augment (18) with the velocity of Q_i, given in (20), to impose a velocity limit on the quadrotor.

$$\begin{aligned}\dot{p}_{x,r} &= v_{x,r}\\\dot{p}_{y,r} &= v_{y,r}\\\dot{v}_{x,r} &= u_{x,i} - u_{x,j}\\\dot{v}_{y,r} &= u_{y,i} - u_{y,j}\\\dot{v}_{x,i} &= u_{x,i}\\\dot{v}_{y,i} &= u_{y,i}\end{aligned} \tag{20}$$

C. Reachability-Based Controllers

Reachability analysis is useful for constructing controllers in a large variety of situations. In order to construct different controllers, an appropriate target set needs to be defined depending on the goal of the controller. If one defines the target set to be a set of desired states, the reachable set, once computed, would represent the states from which a system needs to first arrive at in order to reach the desired states. On the other hand, if the target set represents a set of undesirable states, then the reachable set would indicate the region of the state space that the system needs to avoid. In addition, the system dynamics with which the reachable set is computed provides additional flexibility when using reachability to construct controllers.

Using a number of different target sets and dynamics, we now propose different reachability-based controllers used for vehicle mode transitions in our platooning concept.

1. *Getting to a Target State*

The controller used by a vehicle to a target state is important in a couple of situations in the platooning context. First, a vehicle in the "Free" mode can use the controller to merge onto a highway, forming a platoon and changing modes to a "Leader" vehicle. Second, a vehicle in either the "Leader" mode or the "Follower" mode can use this controller to change to a different highway, changing modes to a "Leader" vehicle.

In both of the above cases, we use the dynamics of a single vehicle specified in 9. The target state would be a position $(\bar{p}_x, \bar{p}_y)$ representing the desired merging point on the highway, along with a velocity $(\bar{v}_x, \bar{v}_y)$ that corresponds to the speed and direction of travel specified by the highway. For the reachability computation, we define the target set to be a small range of states around the target state $x_H = (\bar{p}_x, \bar{p}_y, \bar{v}_x, \bar{v}_y)$:

$$\mathcal{L}_H = \{x : |p_x - \bar{p}_x| \leq r_{p_x}, |v_x - \bar{v}_x| \leq r_{v_x}, \\ |p_y - \bar{p}_y| \leq r_{p_y}, |v_y - \bar{v}_y| \leq r_{vv_y}\}. \tag{21}$$

Here, we represent the target set $\mathcal{L}_H$ as the zero sublevel set of the function $l_H(x)$, which specifies the terminal condition of the HJB PDE that we need to solve. Once the HJB PDE is solved, we obtain the reachable set from the subzero level set of the solution $V_H(t,x)$: $\mathcal{V}_H(t)$. More concretely, $\mathcal{V}_H(T) = \{x : V_H(-T,x) \leq 0\}$ is the set of states from which the system can be driven to the target $\mathcal{L}_H$ within a duration of T.

Depending on the time horizon T, the size of the reachable set $\mathcal{V}_H(T)$ varies. In general, a vehicle may not initially be inside the reachable set $\mathcal{V}_H(T)$, yet it needs to be in order to get to its desired target state. Determining a control strategy to reach $\mathcal{V}_H(T)$ is itself a reachability problem (with $\mathcal{V}_H(T)$ as the target set), and it would seem like this reachability problem needs to be solved in order for us to use the results from our first reachability problem. However, practically, one could choose T to be large enough to cover a sufficiently large area to include any practically conceivable initial state. From our simulations, a suitable algorithm for getting to a desired target state is as follows:

1. Move towards $\bar{x}_H$ in a straight line, with some velocity, until $V_H(-T,x) \leq 0$. In practice, this step consistently drives the system into the reachable set.

2. Apply the optimal control extracted from $V_H(-T,x)$ according to (17) until $\mathcal{L}_H$ is reached.

2. *Getting to a State Relative to Another Vehicle*

In the platooning context, being able to go to a state relative to another moving vehicle is important for the purpose of forming and joining platoons. For example, a "Free" vehicle may join an existing platoon that is on a highway and change modes to become a "Follower". Also, a "Leader" or "Follower" may join another platoon and afterwards go into the "Follower" mode.

To construct a controller for getting to a state relative to another vehicle, we use the relative dynamics of two vehicles, given in 19. In general, the target state is specified to be some position $(\bar{p}_{x,r}, \bar{p}_{y,r}$ and velocity $(\bar{v}_{x,r}, \bar{v}_{y,r})$ relative to a reference vehicle. In the case of a vehicle joining a platoon that maintains a single file, the reference vehicle would be the platoon leader, the desired relative position would be a certain distance behind the leader, depending on how many other vehicles are already in the platoon, and the desired relative velocity would be $(0,0)$ so that the formation can be kept.

For the reachability problem, we define the target set to be a small range of states around the target state $x_P = (\bar{p}_{x,r}, \bar{p}_{y,r}, \bar{v}_{x,r}, \bar{v}_{y,r}$:

$$\mathcal{L}_P = \{x : |p_{x,r} - \bar{p}_{x,r}| \leq r_{p_x}, |v_{x,r} - \bar{v}_{x,r}| \leq r_{v_x}, \\ |p_{y,r} - \bar{p}_{y,r}| \leq r_{p_y}, |v_{y,r} - \bar{v}_{y,r}| \leq r_{v_y}\} \tag{22}$$

The target set $\mathcal{L}_P$ is represented by the zero sublevel set of the implicit surface function $l_P(x)$, which specifies the terminal condition of the HJI PDE (13). The zero sublevel set of the solution to (13), $V_P(-T,x)$, gives us the set of relative states from which a quadrotor can reach the target in the relative coordinates within a duration of T. In the reachable set computation, we assume that the reference vehicle moves along the highway at constant speed, so that $u_j(t) = 0$. The following is a suitable algorithm for a vehicle joining a platoon to follow the platoon leader:

1. Move towards $\bar{x}_P$ in a straight line, with some velocity, until $V_P(-T,x) \leq 0$.

2. Apply the optimal control extracted from $V_P(-T,x)$ according to (14) until $\mathcal{L}_P$ is reached.

3. *Avoiding Collision*

A vehicle can use a liveness controller described in the previous sections when it is not in any danger of collision with other vehicles. If the vehicle could potentially be involved in a collision within the next short period of time, it must switch to a safety controller. The safety controller is available in every mode, and executing the safety controller to perform an avoidance maneuver does not change a vehicle's mode.

In the context of our platooning concept, we define an unsafe configuration as follows: a vehicle is either within a minimum separation distance d to a reference vehicle in both the x and y directions, or is traveling with a speed above the speed limit $v_{\max}$ in either of the x and y directions. To take this specification into account, we use the augmented relative dynamics given by (20) for the reachability problem, and define the target set as follows:

$$\mathcal{L}_S = \{x : |p_{x,r}|, |p_{y,r}| \leq d \vee |v_{x,i}| \geq v_{\max} \vee |v_{y,i}| \geq v_{\max}\} \tag{23}$$

We can now define the implicit surface function $l_S(x)$ corresponding to $\mathcal{L}_S$, and solve the HJI PDE (13) using $l_S(x)$ as the terminal condition. As before, the zero sublevel set of the solution $V_S(t,x)$ specifies the reachable set $\mathcal{V}_S(t)$, which characterizes the states in the augmented relative coordinates, as defined in (20), from which Q_i *cannot* avoid $\mathcal{L}_S$ for a time period of t, if Q_j uses the worst case control. To avoid collisions, Q_i must apply the safety controller according to (14) on the boundary of the reachable set in order to avoid going into the reachable set. The following algorithm wraps our safety controller around liveness controllers:

1. For a specified time horizon t, evaluate$V_S(-t, x_i - x_j)$ for all $j \in \mathcal{Q}(i)$.

   $\mathcal{Q}(i)$ is the set of quadrotors with which quadrotor i checks safety against. We discuss $\mathcal{Q}(i)$ in Section **??**.

2. Use the safety or liveness controller depending on the values $V_S(-t, x_i - x_j), j \in \mathcal{Q}(i)$:

   If $\exists j \in \mathcal{Q}(i), V_S(-t, x_i - x_j) \leq 0$, then Q_i, Q_j are in potential conflict, and Q_i must use a safety controller; otherwise Q_i uses a liveness controller.

THE FOLLOWING SHOULD BE IN ANOTHER SECTION

4. *Other Quadrotor Maneuvers*

Reachability was used in Sections **??** and **??** for the relatively complex maneuvers. For the simpler maneuvers of traveling along a highway and following a platoon, we resort to simpler controllers described below.

5. *Traveling along a highway*

We use a model-predictive controller (MPC) for traveling along a highway; this controller allows the leader to travel along a highway at a pre-specified speed. Here, the goal is for a leader quadrotor to track a constant-altitude path, defined as a curve $\bar{p}(s)$ parametrized by $s \in [0,1]$ in $p = (p_x, p_y)$ space (position space), while maintaining a velocity $\bar{v}(s)$ that corresponds to constant speed in the direction of the highway. Assuming that the initial position on the highway, $s_0 = s(t_0)$ is specified, such a controller can be obtained from the following optimization problem over the time horizon $[t_0, t_1]$:

$$\begin{aligned} \text{minimize} \quad & \int_{t_0}^{t_1} \big\{\|p(t) - \bar{p}(s(t))\|_2 + \\ & \qquad \|v(t) - \bar{v}(s(t))\|_2 + 1 - s\big\} dt \\ \text{subject to} \quad & \dot{x} = f(x,u) \text{ where } f \text{ is given in (9)} \\ & |u_x|, |u_y| \leq u_{\max}, |v_x|, |v_y| \leq v_{\max} \\ & s(t_0) = s_0, \dot{s} \geq 0 \end{aligned} \tag{24}$$

If we discretize time, and assume that $\bar{p}(\cdot)$ is linear, then the above optimization is convex, and can be quickly solved.

6. *Following a Platoon*

Follower vehicles use a feedback control law tracking a nominal position and velocity in the platoon, with an additional feed-forward term given by the leader's acceleration input; here, for simplicity, we assume perfect communication between the leader and the follower vehicles. This following law enables smooth vehicle trajectories in the relative platoon frame, while allowing the platoon as a whole to perform agile maneuvers by transmitting the leader's acceleration command $u_{P_1}(t)$ to all vehicles.

The i-th member of the platoon, Q_{P_i}, is expected to track a relative position in the platoon $r^i = (r^i_x, r^i_y)$ with respect to the leader's position p_{P_1}, and the leader's velocity v_{P_1} at all times. The resulting control law has the form:

$$u^i(t) = k_p\left[p_{P_1}(t)+r^i(t)-p^i(t)\right]+k_v\left[v_{P_1}(t)-v^i(t)\right]+u_{P_1}(t) \quad (25)$$

for some $k_p, k_v > 0$. The leader can modify the nominal position of vehicles in the platoon, for example to command the formation to turn. In particular, a simple rule for determining $r^i(t)$ in a single-file platoon is given for Q_{P_i} as:

$$r^i(t) = -(i-1)b\frac{v_{P_1}}{\|v_{P_1}\|_2} \quad (26)$$

where b is the spacing between vehicles along the platoon. and $\frac{v_{P_1}}{\|v_{P_1}\|_2}$ is the platoon leader's direction of travel.

D. Inter-Platoon Controllers

E. Safety Analysis

Under normal operations in a single platoon, each follower quadrotor $Q_i, i = P_2, \ldots, P_{N-1}$ in a platoon checks whether it is in the safety reachable set with respect to $Q_{P_{i-1}}$ and $Q_{P_{i+1}}$. So $\mathcal{Q}(i) = \{P_{i+1}, P_{i-1}\}$ for $i = P_2, \ldots, P_{N-1}$. Assuming there are no nearby quadrotors outside of the platoon, the platoon leader Q_{P_1} checks safety against Q_{P_2}, and the platoon trailer Q_{P_N} checks safety against $Q_{P_{N-1}}$. So $\mathcal{Q}(P_1) = \{P_2\}, \mathcal{Q}(P_N) = \{P_{N-1}\}$. When all quadrotors are using liveness controllers to perform their allowed maneuvers, no pair of quadrotors should be in an unsafe configuration if the liveness controllers are well-designed. However, occasionally a quadrotor Q_k may behave unexpectedly due to faults or malfunctions, in which case it may come into an unsafe configuration with another quadrotor.

With our choice of $\mathcal{Q}(i)$ and the assumption that the platoon is in a single-file formation, some quadrotor Q_i would get into an unsafe configuration with Q_k, where Q_k is likely to be the quadrotor in front or behind of Q_i. In this case, a "safety breach" occurs. Our synthesis of the safety controller guarantees that between every pair of quadrotors Q_i, Q_k, as long as $V_S(-t, x_i - x_k) > 0$, $\exists u_i$ to keep Q_i from colliding with Q_k for a desired time horizon t, despite the worst case (an adversarial) control from Q_k. Therefore, as long as the number of "safety breaches" is at most one, Q_i can simply use the optimal control to avoid Q_k and avoid collision for the time horizon of t. Since quadrotors in platoons are able to exit the current altitude range within a duration of t_{internal}, if we choose $t = t_{\text{internal}}$, the safety breach would always end before any collision can occur.

Within a duration of t_{internal}, there is a small chance that additional safety breaches may occur. However, as long as the total number of safety breaches does not exceed the number of affected quadrotors, collision avoidance of all the quadrotors can be guaranteed for the duration t_{internal}. However, as our simulation results show, putting quadrotors in single-file platoons makes the likelihood of multiple safety breaches low during a quadrotor malfunction and during the presence of one intruder vehicle.

In the event that multiple safety breaches occur for some of the quadrotors due to a malfunctioning quadrotor within the platoon or an intruding quadrotor outside of the platoon, those quadrotors with more than one safety breach still have the option of exiting the highway altitude range in order to avoid collisions. Every extra altitude range reduces the number of simultaneous safety breaches by 1, so K simultaneous safety breaches can be resolved using $K-1$ different altitude ranges.

Given that quadrotors within a platoon are safe with respect to each other, each platoon can be treated as a single vehicle, and perform collision avoidance with other platoons. By treating each platoon as a single unit, we reduce the number of individual quadrotors that need to check for safety against each other, reducing overall computation burden.

F. Numerical Simulations

In this section, we consider several situations that quadrotors in a platoon on an air highway may commonly encounter, and show via simulations the behaviors that emerge from the controllers we defined in Sections **??** and **??**.

1. *Forming a Platoon*

We first consider the scenario in which some quadrotors are trying to merge onto an initially unoccupied highway. In order to do this, each quadrotor first checks for safety with respect to the other quadrotors, and uses the safety controller if necessary, according to Section **??**. Otherwise, the quadrotor uses the liveness controller described in Section **??**.

For the simulation example, the highway is specified by the line $p_y = 0.5p_x$, the point of entry on the highway is chosen to be $(\bar{p}_x, \bar{p}_y) = (4, 2)$, and the velocity on the highway is chosen to be $(\bar{v}_x, \bar{v}_y) = \frac{\bar{v}}{\sqrt{0.5^2+1^2}}(0.5, 1)$. The velocity simply states that the quadrotors must travel at a speed $\bar{v} = 3$ along the direction of the highway. This forms the target state $\bar{x}_H = (\bar{p}_x, \bar{v}_x, \bar{p}_y, \bar{v}_y)$, from which we define the target set $\mathcal{L}_H$ as in Section **??**.

The first quadrotor that completes merging onto the empty highway creates a platoon and becomes its leader, while subsequent quadrotors form a platoon behind the leader in a pre-specified order according to the process described in Section **??**. Here, we choose $(\bar{p}_{x,r}, \bar{p}_{y,r})$ to be a distance b behind the last quadrotor in the platoon, and $(\bar{v}_{x,r}, \bar{v}_{y,r}) = (0, 0)$. This gives us the target set $\mathcal{L}_P$ that we need.

Figures **??** and **??** show the simulation results. Since the liveness reachable sets are in 4D and the safety reachable sets are in 6D, we compute and plot their 2D slices based on the quadrotors' velocities and relative velocities.

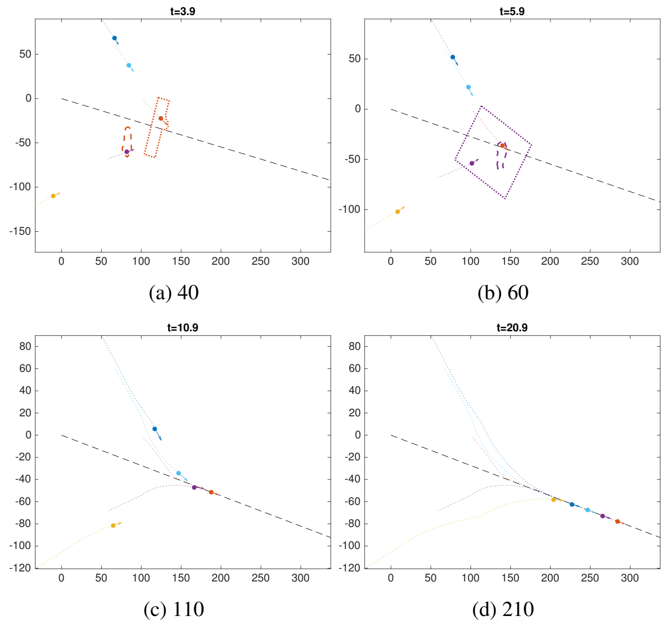
Figure **??** illustrates the use of liveness and safety reachable sets using just two quadrotors to reduce visual clutter. The first quadrotor Q_1 (red disk) first travels in a straight line towards the highway merging point $\bar{x}$ (red circle) at $t = 1.5$, because it is not yet in the liveness reachable set for merging onto the highway (red dotted boundary). When it is within the liveness reachable set boundary at $t = 2.8$, it is "locked-in" to the target state $\bar{x}_H$, and follows the optimal control in (17) to $\bar{x}_H$. During the entire time, the Q_1 checks whether it may collide with Q_2 within a time horizon of t_{external}; we chose $t_{\text{external}} = 3 > t_{\text{internal}}$. However, since Q_1 never goes into the boundary of the safety reachable set (red dashed boundary), it uses the liveness controller the entire time.

After Q_1 has reached $\bar{x}_H$, it forms a platoon, becomes the platoon leader, and continues to travel along the highway. Q_2 (blue disk), at $t = 7$, begins joining the platoon behind Q_1, by moving towards the target $\bar{x}_P$ relative to the position of Q_1. Note that $\bar{x}_P$ moves with Q_1 as $\bar{x}_P$ is defined in terms of the relative states of the two quadrotors. When Q_2 moves inside the liveness reachable set boundary for joining the platoon (blue dotted boundary), it is "locked-in" to the target relative state $\bar{x}_P$, and begins following the optimal control in (14) towards the target as long as it stays out of the safety reachable set (blue dashed boundary).

Figure **??** shows the behavior of all 5 quadrotors which eventually form a platoon and travel along the highway together. The liveness controllers allow the quadrotors to optimally and smoothly enter the highway and join platoons, while the safety controllers prevent collisions from occurring.

2. *Intruder Vehicle*

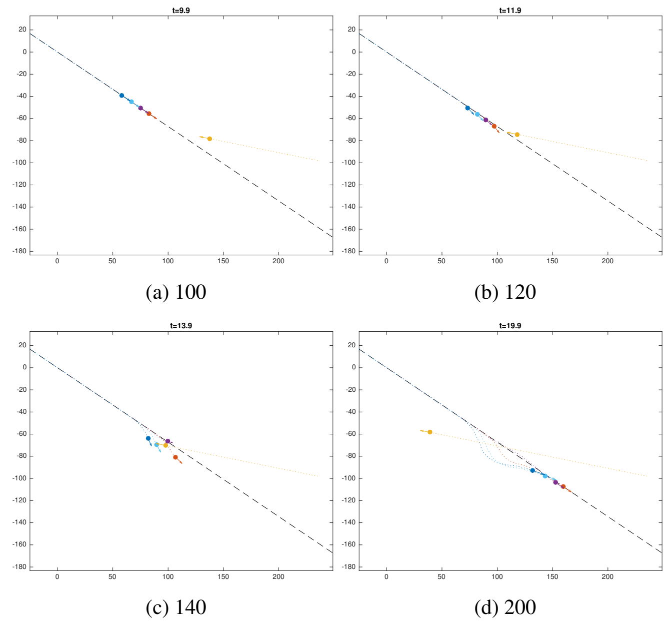
We now consider the scenario in which a platoon of quadrotors encounters an intruder vehicle. To avoid collision, each quadrotor checks for safety with respect to the intruder and any quadrotor in front and



(a) 40 (b) 60

(c) 110 (d) 210

Figure 7. form platoons



(a) 100 (b) 120

(c) 140 (d) 200

Figure 8. intruder

behind in the platoon. If necessary, the quadrotor uses the safety controller, otherwise it uses the appropriate liveness controller, depending on whether it is a leader or follower.

Figure **??** shows the simulation result. At $t = 0$, a platoon of 4 quadrotors, $Q_{P_i}, i = 1, \dots, 4$ with $P_i = i$, travels along the highway defined by the line $p_y = p_x$. An intruder vehicle Q_0 (red disk) starts from position $(p_x, p_y) = (40, 30)$ and heads toward bottom-left of the grid. As before, platoon members incur a safety breach if $V(-t_{\text{internal}}, x_{P_i} - x_{P_j}) \leq 0$. However, with respect to Q_0, platoon members incur a safety breach if $V(-t_{\text{external}}, x_{P_i} - x_0) \leq 0$ with $t_{\text{external}} > t_{\text{internal}}$.

The platoon leader Q_{P_1}'s (black disk) safety is unaffected by the intruder, thus it simply follows its original path on the highway using the liveness controller described in Section 5. Followers Q_{P_2} (blue disk), Q_{P_3} (green disk) and Q_{P_4} (pink disk), on the other hand, must use the safety controller in order to avoid collision with the intruder ($t = 3.3, 6.2$). This causes their paths to deviate off the highway. Once each quadrotor is safe relative to the intruder, they merge back onto the highway, join the original platoon and continue traveling along it ($t = 12.4$).

Figure **??** illustrates the use of safety reachable sets in this scenario using only Q_{P_2} as an example. The safety reachable sets of Q_{P_2} with respect to the intruder Q_0 (red dashed line), Q_{P_1} (black dashed line) and Q_{P_3} (green dashed line) are shown. With respect to Q_0, Q_{P_2}'s safety is considered to be breached if $V_S(-t_{\text{external}}, x_{P_2} - x_0) \leq 0$. To avoid possible collision with the intruder, Q_{P_2} must remain outside the safety reachable set with respect to the intruder. The same applies to collision avoidance with Q_{P_1} and Q_{P_3}.

Initially, Q_{P_2} ($P_2 = 2$) is a follower and is outside all 3 safety reachable sets. Hence it is allowed to use the liveness controller to follow the platoon as described in Section 6. At time $t = 0.6$, Q_2 comes to the boundary of the safety set with respect to the intruder and therefore must apply the safety control law to avoid potential future collision. Thus it splits the original platoon and becomes the leader of a new platoon consisting of itself, Q_3 and Q_4. Q_2 keeps using the safety controller until it is safe with respect to the intruder again at $t = 3$, since if it tried to merge back to the highway before this time, it would enter the safety set with respect to the intruder and lose the t_{external} safety guarantee. After $t = 3$, Q_2 is safe to use the liveness controller again to merge back onto the highway and join the original platoon as a follower. Note that during the entire time, Q_2 maintains safety against the intruder, Q_1 and Q_3 by always staying outside of all three safety reachable sets.

3. *Changing highways*

IV. Conclusions

We considered single-file platoons of UAVs modeled by hybrid systems traveling along air highways. Using HJ reachability, we proposed liveness controllers for merging onto highways and merging into existing platoons, and wrapped a safety controller around liveness controllers to ensure no collision between the UAVs can occur. Under the assumption that faulty vehicles can descend after a pre-specified duration, our safety controller guarantees that no collisions will occur in a single altitude level as long as at most one safety breach occurs for each vehicle in the platoon. Additional safety breaches can be handled by multiple altitude ranges in the airspace. Our simulations show that by putting vehicles into single-file platoons, the likelihood of having multiple safety breaches is low, and conflicts involving a single malfunctioning UAV or intruder can be resolved in a single altitude level.

Immediate future work includes exploring different vehicle models, investigating algorithms for resolving multiple safety breaches within the same altitude, and more broadly related problems such as airspace structure and air highway placement.

Appendix

Acknowledgments

This work is supported in part by NSF under CPS:ActionWebs (CNS-0931843) and CPS:FORCES (CNS1239166), by NASA under grants NNX12AR18A and UCSCMCA-14-022 (UARC), by ONR under grants N00014-12-1-0609, N000141310341 (Embedded Humans MURI), and MIT_5710002646 (SMARTS MURI), and by AFOSR under grants UPenn-FA9550-10-1-0567 (CHASE MURI) and the SURE project.

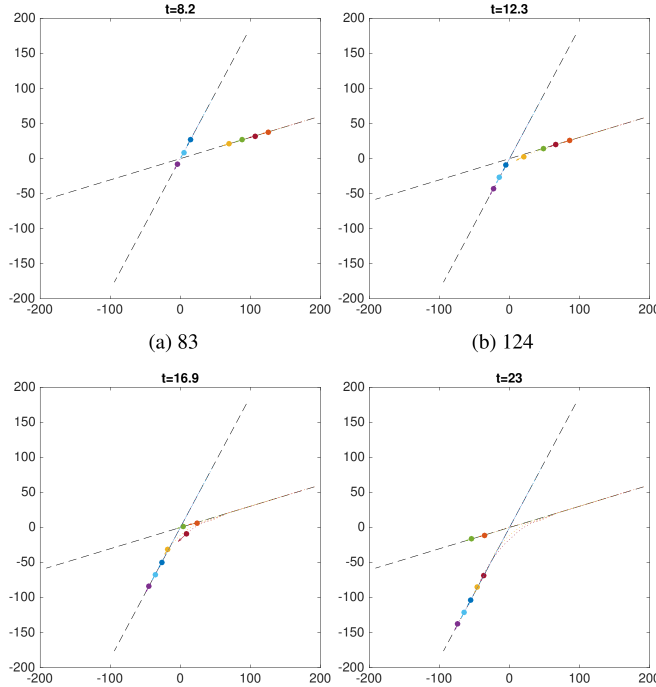
References

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

[2] Amazon.com, Inc., “Amazon Prime Air,” 2014.

[3] Stewart, J., “Google tests drone deliveries in Project Wing trials,” 2014.

[4] Jointed Planning and Development Office (JPDO), “Unmanned Aircraft Systems (UAS) Comprehensive Plan – A Report on the Nation’s UAS Path Forward,” Tech. rep., Federal Aviation Administration, 2013.



(a) 83 (b) 124

(c) 170 (d) 231

Figure 9. change highways

[5] Debusk, W. M., "Unmanned Aerial Vehicle Systems for Disaster Relief: Tornado Alley," *Infotech@Aerospace Conferences*, 2010.

[6] Bokanowski, O., Forcadel, N., and Zidani, H., "Reachability and minimal times for state constrained nonlinear problems without any controllability assumption," *SIAM Journal on Control and . . .* , 2010, pp. 1–24.

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

[8] Barron, E. and Ishii, H., "The Bellman equation for minimizing the maximum cost," *Nonlinear Analysis: Theory, Methods & Applications*, 1989.

[9] Fisac, J. F., Chen, M., Tomlin, C. J., and Sastry, S. S., "Reach-Avoid Problems with Time-Varying Dynamics, Targets and Constraints." 2015.

[10] Chen, M., Zhou, Z., and Tomlin, C., "Multiplayer Reach-Avoid Games via Low Dimensional Solutions and Maximum Matching," *Proceedings of the American Control Conference*, 2014.

[11] Ding, J., Sprinkle, J., Sastry, S. S., and Tomlin, C. J., "Reachability Calculations for Automated Aerial Refueling," *IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

[12] Mitchell, I., *A Toolbox of Level Set Methods*, 2009, http://people.cs.ubc.ca/~mitchell/ToolboxLS/index.html.

[13] Osher, S. and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, 2002, ISBN: 978-0-387-95482-0.

[14] Sethian, J. A., "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, Vol. 93, No. 4, 1996, pp. 1591–1595.

[15] Kavathekar, P. and Chen, Y., "Vehicle Platooning: A Brief Survey and Categorization," Vol. 3, 2011, pp. 829–845.

[16] McMahon, D., Hedrick, J., and Shladover, S., "Vehicle Modelling and Control for Automated Highway Systems," *American Control Conference, 1990*, May 1990, pp. 297–303.

[17] Hedrick, J., Zhang, G., Narendran, V., Chang, K., for Advanced Transit, P., (Calif.), H., and University of California, B. I. o. T. S., *Transitional Platoon Maneuvers in an Automated Highway System*, California PATH Program, Institute of Transportation Studies, University of California at Berkeley, 1992.

[18] Lygeros, J., Godbole, D., and Sastry, S., "Verified hybrid controllers for automated vehicles," *Automatic Control, IEEE Transactions on*, Vol. 43, No. 4, Apr 1998, pp. 522–539.

[19] Alam, A., Gattami, A., Johansson, K. H., and Tomlin, C. J., "Establishing Safety for Heavy Duty Vehicle Platooning: A Game Theoretical Approach," *18th IFAC World Congress*, Milan, Italy, August 2011.

[20] Lygeros, J., Sastry, S., and Tomlin, C., *Hybrid Systems: Foundations, advanced topics and applications*, Springer Verlag, 2012.

[21] Evans, L. C. and Souganidis, P. E., "Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations," *Indiana University Mathematics Journal*, Vol. 33, No. 5, 1984, pp. 773–797.

[22] Varaiya, P., "On the existence of solutions to a differential game," *SIAM Journal on Control*, Vol. 5, No. 1, 1967, pp. 153–162.

[23] Crandall, M. G., Evans, L. C., and Lions, P. L., "Some Properties of Viscosity Solutions of Hamilton-Jacobi Equations," *Transactions of the American Mathematical Society*, Vol. 282, No. 2, April 1984, pp. 487.

[24] Chen, M. and Tomlin, C. J., "Exact and Efficient Hamilton-Jacobi Reachability for Decoupled Systems," *Submitted to 54th IEEE Conference on Decision and Control*, 2015.