Applications of Hybrid Reachability Analysis to Robotic Aerial Vehicles

Jeremy H. Gillula*
Stanford University
Stanford, CA 94305, USA
jgillula@cs.stanford.edu

Gabriel M. Hoffmann*
Stanford University
Stanford, CA 94305, USA
gabe.hoffmann@gmail.com

Haomiao Huang*
Stanford University
Stanford, CA 94305, USA
haomiao@stanford.edu

Michael P. Vitus*
Stanford University
Stanford, CA 94305, USA
vitus@stanford.edu

Claire J. Tomlin
UC Berkeley
Berkeley, CA 94720, USA
tomlin@eecs.berkeley.edu

October 4, 2010

Abstract

The control of complex nonlinear systems can be aided by modeling each system as a collection of simplified hybrid modes, with each mode representing a particular operating regime defined by the system dynamics or by a region of the state space in which the system operates. Guarantees on the safety and performance of such hybrid systems can still be challenging to generate, however. Reachability analysis using a dynamic game formulation with Hamilton-Jacobi methods provides a useful way to generate these types of guarantees, and the technique is flexible enough to analyze a wide variety of systems. This paper presents two applications of reachable sets, both focused on guaranteeing the safety and performance of robotic aerial vehicles. In the first example, reachable sets are used to design and implement a backflip maneuver for a quadrotor helicopter. In the second, reachability analysis is used to design a decentralized collision avoidance algorithm for multiple quadrotors. The theory for both examples is explained, and successful experimental results are presented from flight tests on the STARMAC quadrotor helicopter platform.

1 Introduction

As robotic systems increase in complexity, their design and analysis have become increasingly challenging, making provable guarantees on safety and performance more difficult to provide. One successful approach is to break down the behavior of a complex nonlinear system into a collection of discrete states or *modes*, with different continuous dynamics for each mode. Such a hybrid decomposition can simplify analysis of the overall system behavior, while planning and control are also simplified by the ability to generate plans at the level of the discrete modes (see Figure 1). This approach has proven successful in a variety of applications, including manipulator motion planning (Lozano-Perez et al. 1984; Burridge et al. 1999), specifications for mobile robot behaviors (Kress-Gazit et al. 2008), and aircraft trajectory planning where complex trajectories were designed by building up sequences of discrete maneuvers (Frazzoli et al. 2005).

One approach to analyzing the safety of such hybrid systems is to pose the control problem as a dynamic game between a control input that is managed by the system designer as either human or

*These authors contributed equally to this work.

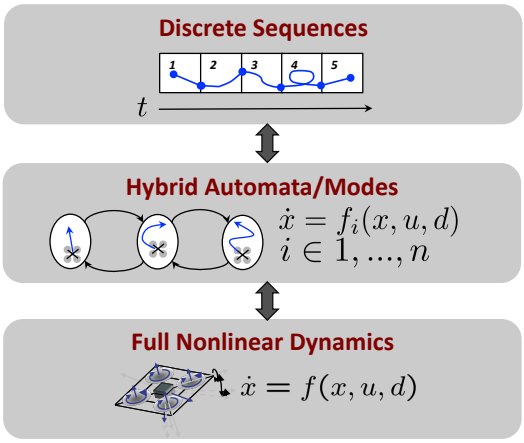


Figure 1: Hybrid hierarchical representation of maneuvers for a quadrotor helicopter.

autonomous input, and a disturbance input that models the effects of the environment or other systems that cannot be controlled by the designer. Then, under the assumption of worst case behavior of the disturbance and best case behavior of the control, one can effectively "carve out" regions of the state space as unsafe by using reachability analysis to determine the region of the state space that the system can reach in each mode. This approach motivates the formulation of the problem as a dynamic game between control and disturbance.

Solving the reachability problem for such a dynamic game may be done by solving an appropriate Hamilton-Jacobi equation (Lygeros et al. 1999; Tomlin et al. 2000), and effective computational techniques for doing this have been developed (Bardi and Capuzzo-Dolcetta 1997; Bardi et al. 1999; Cardaliaguet et al. 1999; Kurzhanski and Varaiya 2002; Mitchell et al. 2005). Performance objectives such as analyzing the attainability of certain system states or behaviors may be similarly posed as a reachability problem. The resulting backward reachable sets define those states the system can start in such that the system is guaranteed to either keep out of, or arrive into, a user-defined region of the state space within some time horizon (Mitchell et al. 2005). Although the formal guarantees provided by reachability analysis are only valid to within the limits of the model used, in many cases the model uncertainty can be incorporated into the analysis as a disturbance. Throughout this paper when references are made to provable guarantees, it is implied that the guarantees hold only under the modeling assumptions used.

Reachability analysis of this form has been used to derive guaranteed safe switching regions for a collision avoidance controller for manned aircraft (Tomlin et al. 1998; Mitchell et al. 2005), where an evading aircraft could be guaranteed to stay a given distance away from a pursuer. The approach has also been generalized to autoland systems and ultra-close formation flight (Oishi et al. 2008; Ding et al. 2008), demonstrating that in addition to being a rigorous way to provide provable guarantees about system safety, the methods are flexible enough to be adapted to many different system models and applications.

In this work, reachability analysis for hybrid systems is used to derive provably safe control laws for two challenges involving quadrotor helicopter unmanned aerial vehicles (UAVs). In the first, it is used to verify the safety and performance criteria for an aerobatic UAV maneuver. In the second, reachable sets are used to address the task of collision avoidance between multiple UAVs. While the methodology presented can be computationally expensive, we show that useful, intuitive information can be obtained by applying this method to simplified models of the quadrotor systems. In general, the methodology can be used to rapidly rule out large parts of the state space as unreachable, and allow the designer to focus on the operation of the vehicle in potentially problematic areas.

To design provably safe aerobatic maneuvers, a novel method is presented that uses hybrid dynamics and reachability tools. It is applied to a quadrotor helicopter performing a backflip maneuver with three modes: impulse (initializing rotation), drift (motors off while rotating and free-falling), and recovery

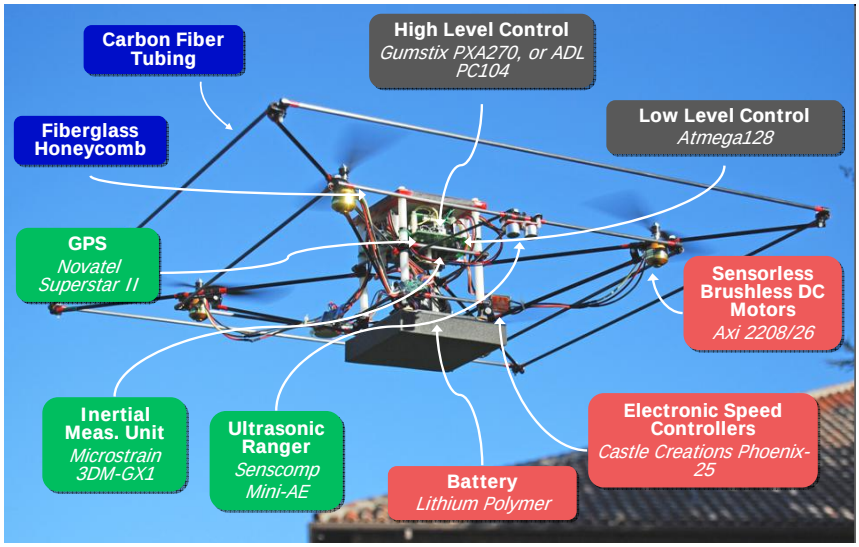


Figure 2: One of the STARMAC quadrotor vehicles.

(return to controlled hover). Provably safe switching conditions on altitude, attitude, and their rates are generated using the solution of the Hamilton-Jacobi equation in the dynamic game formulation of reachable sets, to guarantee that the vehicle will successfully pass through all three modes to arrive at a specified, safe, final condition. The initial concept behind this work was presented in (Gillula et al. 2009) and experimental results were highlighted in (Gillula et al. 2010).

In the second application, reachable set analysis is used to design collision avoidance algorithms between multiple vehicles. The algorithms use a decentralized cooperative switching control strategy, and are minimally invasive in the sense that they only affect control inputs when required to avoid collision. As a result, the overall trajectories may be suboptimal in performance when a collision avoidance action is taken; however, when no collision is imminent, the presented control system does not interfere. Two control laws are presented, one for two vehicles ($n_v = 2$), and one for more than two vehicles ($n_v > 2$), with a computational complexity of $O(n_v)$. The run time for 200 interacting simulated vehicles was 9.2 ms, comparing very favorably with previous methods (Hoffmann and Tomlin 2010). Preliminary results for this work were presented in (Hoffmann and Tomlin 2008), and have been augmented here with additional flight tests.

Both examples were implemented on the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC), a flexible robotic aerial vehicle platform currently consisting of six autonomous quadrotor vehicles (see Figure 2) each with their own onboard sensing, computation, and control. The quadrotors are highly capable research vehicles and have been used for a number of experiments (Hoffmann et al. 2007). Separate flight tests for the two examples were performed, and their results are presented.

The organization of this paper is as follows. Section 2 discusses the background for the Hamilton-Jacobi formulation of reachability. The first example (maneuver sequencing) is presented in Section 3. A brief review of past maneuver sequencing techniques is presented in Section 3.1, the use of reachability for maneuver sequencing and design is presented in Section 3.2, and flight test results are presented in Section 3.3. The second application example (decentralized collision avoidance) is presented in Section 4.

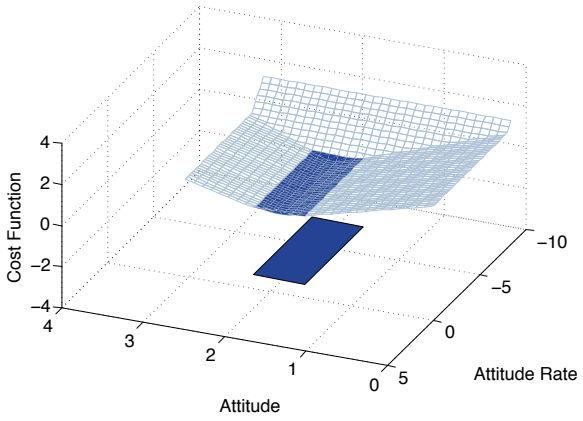


Figure 3: The zero sub-level set of an appropriate cost function is used to define capture or unsafe regions in the state space. The cost function is the light colored region and the desired set is the dark colored region. This example creates a rectangular set of bounded attitude and attitude rate using a union of hyperplanes as the cost function.

A brief review of past decentralized collision avoidance literature is presented in Section 4.1, the use of reachability for collision avoidance is presented in Section 4.2, and results from flight tests are presented in Section 4.3. Finally, conclusions and future work are described in Section 5.

2 Overview of Backward Reachable Sets and Hamilton-Jacobi Reachability

The concept of backward reachability is used in this work as a means of guaranteeing safety, either by generating unsafe sets to avoid during flight, or by generating guaranteed safe sets for mode transitions. A backward reachable set $Pre_\tau(K)$ is defined as the set of all states from which the system can arrive in some set K within a time horizon of length τ, under some appropriate assumptions about the control and disturbance. Two conditions are considered here: safety, in which the goal of the control is to stay out of an undesired set in the state space that the disturbance is trying to force the system into, and attainability, in which the goal is to reach a desired set while the disturbance tries to keep the system out of that set.

The reachable sets in this work are calculated using the technique described in Mitchell et al. with some modifications (Mitchell et al. 2005). For any given mode i, the system evolves under the dynamics

$$\dot{x} = f_i(t, x, u, d) \tag{1}$$

where t is time, x is the system state, u is the control input, and d is the disturbance input. The inputs, u and d, are assumed to be constrained such that their values lie in some sets U and D, respectively. In this method, the reachability problem is posed as a differential game between the control and disturbance inputs; the disturbance chooses the worst case inputs to either drive the system into the undesired set or away from the desired set and the control does the opposite. An overview of the differential game formulation using unsafe sets is described below, but the principles are the same for attainability. This overview is adapted from Tomlin (Tomlin 2009).

2.1 Unsafe Sets for Safety

For safety, the goal is to prevent the system from entering an undesired set, and the disturbance is assumed to be attempting to drive the system into this unsafe area of the state space. The unsafe set

relative to the undesired set K for some time horizon τ is denoted $Pre_\tau(K)$, and is defined as the region of the state space where, for any control inputs $u(t) \in U$, there exists some sequence of disturbances $d(t) \in D$ such that $x(t) \in K$ for some $t \in [-\tau, 0]$. That is, if $x(-\tau) \in Pre_\tau(K)$, then no matter what the control does the disturbance can drive the system into K in time less than or equal to τ. Conversely, if the initial state $x(0)$ is outside of $Pre_\tau(K)$, then there exists a control $u(t)$ that keeps the system out of K for up to time τ.

2.1.1 Dynamic Programming Solution

In the game formulation, the boundary of the initial set K is defined as the zero level set of an appropriately selected cost function $l(x)$ that is negative inside K and positive outside (for an example see Figure 3). To drive the system into the undesired set, the disturbance is modeled as attempting to minimize $l(x)$, while the control attempts to maximize this cost. To be conservative, the disturbance is allowed to select its input after the control input is known. Let the final time be $t = 0$, then the game formulation can be captured by the following value function:

$$J(x,t) = \max_{u \in U} \min_{d \in D} l(x(0)) \tag{2}$$

which includes only a terminal cost $l(x(0))$ that is negative if x lies within the region K, zero on the boundary, and positive otherwise.

Using the tools of dynamic programming, the optimal value of $J(x,t)$ can be calculated by introducing a Hamiltonian. The optimal Hamiltonian H^* will be

$$H^*(x,p) = \max_{u \in U} \min_{d \in D} p^T f_i(t,x,u,d) \tag{3}$$

where p is the Hamiltonian costate (the Lagrange multipliers of the state equation of the optimal control problem), and must satisfy the relationship

$$\begin{aligned} p &= \frac{\partial J}{\partial x}(x,t) && (4)\\ \dot{p} &= -\frac{\partial H}{\partial x}^T && (5)\\ \frac{\partial J}{\partial t}(x,t) &= -H^*(x, \frac{\partial J}{\partial x}(x,t)) && (6)\\ J(x,0) &= l(x) && (7) \end{aligned}$$

where the partial differential equation is evolved "backward" in time to calculate a set of initial states.

Solving this partial differential equation will generate the desired set of unsafe states that enter K in time $t \in [-\tau, 0]$. Suppose the equation for $J(x,0)$ is evolved backward in time to determine a subset of $x(-\tau)$ that satisfies $J(x(-\tau), -\tau) < 0$. Then this set contains all the initial conditions for all possible inputs u such that $l(x(0)) < 0$. In other words, it contains all the initial conditions that can be driven into the unsafe set K at time $t = 0$. This procedure almost produces the desired unsafe set, but it fails to capture initial states that result in trajectories that pass through K but have exited the set at time $t = 0$. To include these, the Hamilton-Jacobi equation can be modified in the following manner:

$$\begin{aligned} \frac{\partial J}{\partial t}(x,t) &= -\min\{0, H^*(x, \frac{\partial J}{\partial x}(x,t))\} && (8)\\ J(x,0) &= l(x). && (9) \end{aligned}$$

In the examples discussed in this paper, the resulting cost function $J(x,t)$ is solved numerically using the Level Set Toolbox developed at the University of British Columbia (Mitchell 2009), or analytically by using a coordinate transform to reduce dimensionality. The Level Set Toolbox uses a cartesian grid to represent the reachable set over the state space, and consequently computational complexity grows exponentially in the number of dimensions. As a result, the toolbox can be used to easily analyze systems of up to three dimensions, while systems of up to six dimensions are tractable but require long computation times. While the exact computation of reachable sets is typically done offline (such as with the aerobatic maneuver example described in Section 3) over-approximating the reachable sets can allow for real-time computation (as is done for the collision avoidance example in Section 4).

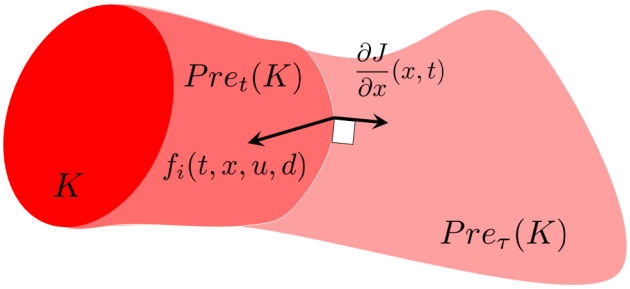


Figure 4: The growth of a reachable set over time is captured by the angle the dynamics make with the outward-pointing normal. Angles greater than $\pi/2$ illustrate that the system must flow into the set, indicating that the set grows locally.

2.1.2 Graphical Interpretation

For insight into how the reachable set evolves over time, it is helpful to interpret the evolution of the Hamilton-Jacobi partial differential equation graphically. Consider some set at time t where

$$Pre_t(K) = \{x : J(x,t) < 0\}. \tag{10}$$

We want to look at how the boundaries of this set evolve over time. Consider a point $x \in \partial Pre_t(K)$ on the boundary where

$$H^*(x, \frac{\partial J}{\partial x}(x,t)) < 0. \tag{11}$$

This implies that

$$\max_{u\in U} \min_{d\in D} H(x, \frac{\partial J}{\partial x}(x,t)) < 0 \tag{12}$$

$$\Leftrightarrow \max_{u\in U} \min_{d\in D} \frac{\partial J}{\partial x}(x,t) f_i(t,x,u,d) < 0 \tag{13}$$

$$\tag{14}$$

which means that for any input $u \in U$, there exists a disturbance $d \in D$ such that

$$\frac{\partial J}{\partial x}(x,t) f_i(t,x,u,d) < 0. \tag{15}$$

Note that $\frac{\partial J}{\partial x}(x,t)$ is normal to the boundary of $Pre_t(K)$ pointing outward, and that $\frac{\partial J}{\partial x}(x,t) f_i(t,x,u,d)$ is the inner product of the vector $f_i(t,x,u,d)$ with this normal (see Figure 4). Therefore, the fact that this inner product is less than zero implies that the angle between the normal vector and the dynamics $f_i(t,x,u,d)$ is greater than $\pi/2$. Consequently, $f_i(t,x,u,d)$ points inside $Pre_t(K)$. The points on the boundary where this is true are thus points where, no matter what the input is, the disturbance d can be chosen such that the dynamics $f_i(t,x,u,d)$ point inside the set $Pre_t(K)$, i.e. the disturbance can instantaneously drive the state into the set.

By looking at the Hamilton Jacobi Equation, this corresponds to points on the boundary where,

$$\frac{\partial J}{\partial t}(x,t) = -\min\{0, H^*(x, \frac{\partial J}{\partial x}(x,t))\} \tag{16}$$

$$= -H^*(x, \frac{\partial J}{\partial x}(x,t)) \tag{17}$$

$$> 0. \tag{18}$$

Similarly, for the points on the boundary that do not grow outward (that is, the inner product points inward or is parallel to the boundary),

$$H^*(x, \frac{\partial J}{\partial x}(x,t)) \geq 0 \tag{19}$$

and thus the Hamilton-Jacobi equation simplifies to,

$$\frac{\partial J}{\partial t}(x,t) = -\min\{0, H^*(x, \frac{\partial J}{\partial x}(x,t))\} \quad (20)$$

$$= 0 \quad (21)$$

Note that the min with zero effectively prevents points that were in the set from exiting and seeming safe. Thus by starting with an initial unsafe set K and numerically evolving the boundary of that set backward in time, a backward reachable set at any given time previous to $t = 0$ can be calculated.

2.2 Capture Sets for Attainability

Symmetrically, for attainability the desired goal of the controller is to drive the system into some desired set D by minimizing the cost function, while the disturbance is assumed to be attempting to drive the system away, thereby maximizing the cost function. Thus a capture set $Pre_\tau(D)$ can be defined for the system which guarantees that there exists a control input that drives the system into D within time horizon τ no matter what the disturbance does. This reverses the role of control and disturbance, where the control is now trying to minimize the final cost and the disturbance is trying to maximize it. Again for robustness the disturbance is allowed knowledge of the control input, with the modified value function being

$$J(x,t) = \min_{u\in U} \max_{d\in D} l(x(0)) \quad (22)$$

and the optimal Hamiltonian thus being

$$H^*(x,p) = \min_{u\in U} \max_{d\in D} p^T f_i(t,x,p,u,d). \quad (23)$$

2.3 Reachable Sets Without Optimal Control

In many cases (such as with the aerobatic maneuver example described in Section 3) it can be convenient to use a particular predetermined controller. This can easily be incorporated into the reachable set analysis by defining a controller

$$u = C_i(t,x) \quad (24)$$

specified for the currently active mode. This controller can then be subsumed into a modified form of the system dynamics, such that

$$\dot{x} = f_i(t,x,C_i(t,x),d) = \hat{f}_i(t,x,d). \quad (25)$$

The optimal Hamiltonians for safety and reachability can be modified by removing the max and min, respectively, over u, and leaving only the optimization over d.

Once the dynamics and initial set (and defining cost function) are selected, the appropriate Hamiltonian of the system can be formulated and the backward reachable sets can be calculated using the Level Set Toolbox (Mitchell 2009) as described in Sections 2.1.1 and 2.1.2. This toolbox was used to compute the reachable sets for the aerobatics shown later in this paper.

In the next two sections, we explain in detail how the techniques described here were used for two very different aerial vehicle applications: aerobatic maneuver design and decentralized collision avoidance.

3 Reachable Sets for Aerobatic Maneuver Design and Execution

As the demands on UAVs grow in complexity, they require increasingly sophisticated control systems to take advantage of their full range of capabilities. This section addresses one such challenge, designing safe aerobatic maneuvers. The full nonlinear dynamics are simplified into a hybrid model and reachability analysis is used to design and implement a backflip maneuver for a quadrotor helicopter.

3.1 Related Work

Many approaches to the control of highly maneuverable aircraft have used statistical learning techniques, for example by copying an expert pilot's example trajectory either through machine learning or via manual creation of approximate trajectories (Coates et al. 2009; Gavrilets et al. 2002; Abbeel et al. 2007), or through iterative schemes which update the control laws at each step using information from experimental runs (Lupashin et al. 2010; Purwin and DAndrea 2009). These methods have been able to push the envelope of what is possible with autonomous control, but since they lack performance guarantees about their stability and robustness, their use must be limited to situations where safety is critical.

As mentioned in Section 1, an alternate approach that allows more rigorous formal analysis is hybrid decomposition. In this method, the behavior of the system is approximated as a discrete set of simpler modes representing the dynamics in specific regimes or portions of the state space. An important consideration in the design and control of systems with switched dynamics is the safety of transitions between modes. For example, in the case of aircraft maneuver sequences it is necessary to ensure that an aircraft completing one maneuver is able to begin the next maneuver without being in an unsafe or infeasible configuration.

This has been accomplished in a variety of ways in the past. Previous helicopter maneuvering work used "trim states" such as steady flight or hover that the vehicle was required to return to after a maneuver before beginning another (Frazzoli et al. 2005). In robotic manipulation, sequences of specially derived Lyapunov functions have been used to guarantee that a defined sequence could be followed (Burridge et al. 1999) as well as analytically calculating regions where a given motion was guaranteed to place a part in a desired configuration (Lozano-Perez et al. 1984). There has also been extensive work in the Hybrid Systems literature on the construction of switching regions for mode switching (Zefran and Burdick 1998; Lazar and Jokic 2009; Egerstedt et al. 1999). Specifically, partitions or manifolds in the state space have been found that are regions of attraction for particular modes or controllers.

Much of the existing work has focused on switching under nominal conditions or sensing uncertainty and has not explicitly considered external disturbances. Additionally, in most cases the particular method to ensure continuity between modes has been specific to the application at hand. These methods have also not considered separate safety requirements such as obstacle avoidance. In this Section, a general method using reachable sets is developed that can be used to design aerobatic maneuvers for a quadrotor helicopter. This method accounts for external disturbances as well as separate safety requirements, and is flexible enough to be adapted to many nonlinear systems. In particular, the Hamilton-Jacobi differential game formulation of reachable sets (Mitchell et al. 2005) is used to construct maneuvers that safely transition through a sequence of modes to perform a backflip maneuver while arriving at a target state and avoiding unsafe states en route. While the actual simulation and experimental trajectories are similar to those achieved in the literature using other methods, the reachable sets provide both formal guarantees of safety and attainability, as well as information about how much the maneuvers can deviate from those given and still maintain those guarantees.

3.2 Maneuver Sequencing Using Reachable Sets

Capture and avoid sets (as described in Section 2) can be used to construct safe sequences of maneuvers. Starting with the final (desired target) set D_n, the dynamics for the final (n^{th}) maneuver can be used along with the attainability formulation described in Section 2.2 to generate a backwards reachable set set $Pre_\tau(D_n)$ for that maneuver (i.e. all points in the state space guaranteed to reach D_n within time horizon τ). This forms a capture set for the n^{th} maneuver. Then the desired set D_{n-1} for the previous ($n - 1^{st}$) maneuver can be selected to be within the capture set $Pre_\tau(D_n)$ of the final maneuver. Thus an initial condition within the capture set $Pre_\tau(D_{n-1})$ of the $n - 1^{st}$ maneuver is guaranteed to arrive within the capture set $Pre_\tau(D_n)$ of the n^{th} maneuver, allowing a safe switch into the n^{th} maneuver and eventual safe arrival at the final target set D_n (see Figure 5). This process can be repeated for any number of desired maneuvers to identify a start region for the entire sequence.[1]

[1]It should be noted that this method is not guaranteed to always produce a solution (though if a solution is produced, it is guaranteed to be feasible). For example, if the disturbances are so great that the reachable sets for two subsequent maneuvers do not overlap, then there will be no guaranteed feasible way to transition between them. Since the reachable sets are generated before initiating the maneuver, however, this does not put the system in any danger (unless the maneuver is initiated anyway, despite the lack of achievability guarantees). In fact, when used properly this lack of a solution could enable the designer to spot problems in the maneuver design or the model of the system ahead of time, and correct them if possible.

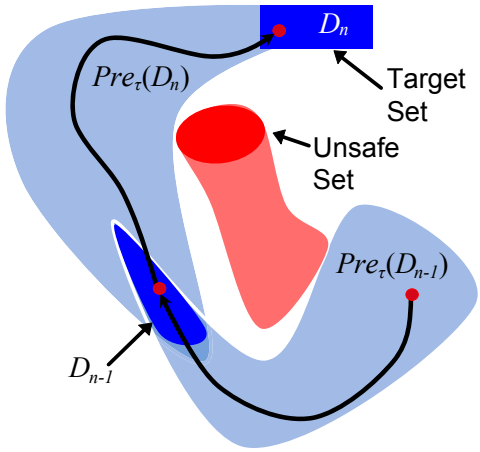


Figure 5: Capture and avoid sets for sequencing two modes/maneuvers.

Avoid sets for safety can be generated in a similar fashion. Starting from an initial unsafe set, the boundary of the full avoid set can be progressively propagated backward using the sequence of mode dynamics. The need to simultaneously consider safety and attainability can be encoded either by choosing capture sets that avoid the unsafe regions of particular reach sets, or by using set-intersection operations to generate reach-avoid sets that will reach target sets while avoiding the unsafe sets.

3.2.1 Problem Formulation

To demonstrate the validity of this approach for maneuver sequencing, this method was used to develop a backflip maneuver for the STARMAC quadrotor helicopter. As the backflip is a planar maneuver the quadrotor's dynamics were modeled in a 2D plane, since the out-of-plane dynamics can be stabilized without affecting the maneuver (as supported by analysis and testing). The planar dynamics are given by:

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \phi \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -\frac{1}{m}C_D^v\dot{x} \\ \dot{y} \\ -\frac{1}{m}\left(mg + C_D^v\dot{y}\right) \\ \dot{\phi} \\ -\frac{1}{Iyy}C_D^\phi\dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ D_x \\ 0 \\ D_y \\ 0 \\ D_\phi \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{1}{m}\sin\phi & -\frac{1}{m}\sin\phi \\ 0 & 0 \\ \frac{1}{m}\cos\phi & \frac{1}{m}\cos\phi \\ 0 & 0 \\ -\frac{l}{Iyy} & \frac{l}{Iyy} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \tag{26}$$

where the state variables x, y, and ϕ represent the vehicle's lateral, vertical, and rotational motion, respectively; D_x, D_y and D_ϕ are disturbances; and constant system parameters are given by m for the vehicle's mass, g for gravity, C_D^v for translational drag[2], C_D^ϕ for rotational drag, and I_{yy} for the moment of inertia.

It should be noted that in situations not captured by the 2D model (e.g. if an unmodeled disturbance overpowered the vehicle's out-of-plane stabilization) the guarantees described in this section would obviously no longer hold. The same statement is true for any guarantee regarding stability and robustness: if

[2]It should be noted that for simplicity the vehicle's drag was modeled as linear with respect to velocity, an assumption that was later shown via experiment to be sufficiently accurate.

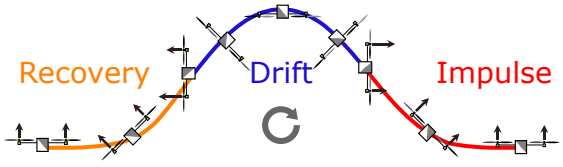


Figure 6: The backflip maneuver, broken into three modes. The vehicle travels from right to left, spinning clockwise as it does so. The size of each arrow indicates the relative thrust from each rotor.

the model used to generate the guarantee does not closely match the actual system, then the guarantee may be invalid. However, a great deal of prior field testing has shown these modeling assumptions to be reasonable under the conditions in which the quadrotors typically operate.

For ease of analysis and visualization, it is useful to decompose high dimensional systems into multiple lower dimensional systems (assuming the system's dynamics decouple). Six-dimensional problems are difficult to visualize and time-intensive to compute, slowing the design process, so the system's states were divided into three sets for independent analysis. The rotational dynamics were analyzed to ensure the attainability of the backflip, the vertical dynamics were analyzed to ensure safety (i.e. that the vehicle remained above some minimum altitude), and the horizontal dynamics were ignored as they were not relevant to successfully completing the maneuver.

3.2.2 Backflip Attainability

For the purpose of guaranteeing attainability, the backflip was divided into three modes as shown in Figure 6: impulse, in which the rotation of the vehicle is initialized; drift, where the vehicle freely rotates and falls under gravity; and recovery, which brings the vehicle to a controlled hover condition.[3]

Each mode was designed using the method described in the beginning of Section 3.2. The target sets D_3, D_2, and D_1 in the $(\phi, \dot{\phi})$ space for the recovery, drift, and impulse modes respectively, were $(0 \pm 5^\circ, 0 \pm 10^\circ/\text{sec})$ – essentially a stable hover configuration, $(110 \pm 20^\circ, -180 \pm 185^\circ/\text{sec})$, and $(310 \pm 10^\circ, -287 \pm 58^\circ/\text{sec})$, as shown in Figure 7. As described in Section 2.3, a fixed controller (in this case a standard PD controller on ϕ) was used to drive the vehicle during the recovery mode, and a D-DD controller was used during the impulse mode.

Finally, $Pre_{\tau_3}(D_3)$, $Pre_{\tau_2}(D_2)$, $Pre_{\tau_1}(D_1)$, were calculated as described in Section 2.2 using worst case disturbances. In this case, the set of allowable disturbances is assumed to be rectangular, with

$$D_{i,min} \leq D_i \leq D_{i,max} \tag{27}$$

along each axis $i = x, y, z$. The magnitude of these disturbances has a significant impact on the resulting reachable sets; if the potential worst case disturbances are too large, then a solution may not exist that will allow the vehicle to reach the target set. The symmetry of the quadrotors used in these experiments means that the aerodynamic center is very close to the center of mass. Therefore the wind conditions the quadrotors typically experience do not impart a significant rotational moment, which was confirmed observationally in test flights. Thus for the recovery and impulse modes, the primary disturbance source was noise in the thrust produced by the motors, which was determined from previous flight data to be less than 5% of the total nominal thrust (Hoffmann et al. 2007). This disturbance was reduced for the drift mode due to the lack of motor noise as the motors were turned off.

Actuator saturation was modeled by constraining the thrusts to lie within the range

$$0 \leq T_i \leq T_{max} \tag{28}$$

for motors $i = 1, 2$ where T_{max} is the maximum thrust produced by the motors. For the quadrotors and controllers discussed in this paper, the typical operating regime is around 40% of maximum thrust

[3]This division was driven largely by the fact that unlike a standard helicopter, a quadrotor's blades have a fixed pitch, which means that a quadrotor is only capable of generating thrust in one direction. As a result, whenever the vehicle is inverted any thrust generated by its rotors must propel it downward. Thus, to successfully complete a backflip maneuver on the STARMAC vehicle with a slow rotational rate (e.g. around 400°/sec), it was necessary to turn off the motors while the vehicle was inverted to prevent the vehicle from propelling itself into the ground.

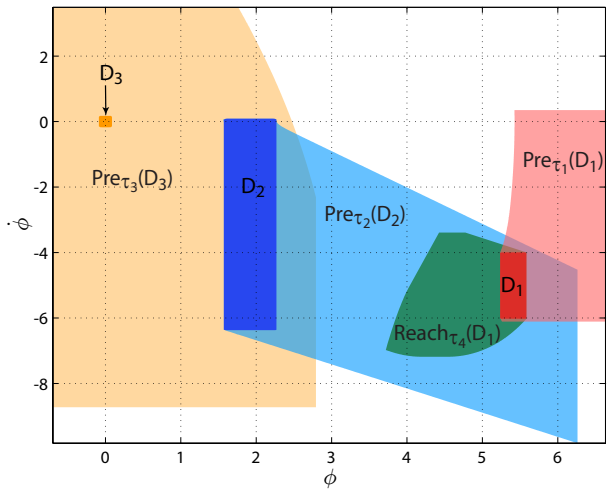


Figure 7: Composite capture sets of the backflip maneuver are plotted in the ϕ (radians) vs $\dot{\phi}$ (radians/second) plane. The backflip maneuver starts in the region labeled $Pre_{\tau_1}(D_1)$ and ends in the region labeled D_3.

with control inputs of at most 20% magnitude, thus actuator saturation, while accounted for in the calculations, did not have an impact on the flight experiments except for motor turn-off in the drift phase.

It was originally assumed that the motors would turn off instantaneously when the vehicle entered the drift mode; some initial experiments proved that assumption incorrect. As a result an additional mode was added for the purpose of analysis. In this mode, the motor turn off was modeled as a linear decay in the vehicle's angular acceleration, i.e.:

$$u_{\text{turnoff}} = \min\{\alpha t + \ddot{\phi}, 0\}/I_{yy} \tag{29}$$

where the parameter α was found using linear regression. These dynamics were then propagated forward from the target set of the impulse mode, D_1; the resulting level set (labeled $Reach_{\tau_4}(D_1)$ in Figure 7) contains all possible states the vehicle could be in while the motors were turning off. Thus, as long as this set was contained in the drift set, $Pre_{\tau_2}(D_2)$, attainability of the backflip was once again guaranteed.

3.2.3 Backflip Safety

To ensure the vehicle would perform the backflip safely, a similar procedure to that described for attainability was used. First, a final unsafe set K_3 was chosen to represent all configurations the vehicle would need to avoid during the recovery mode: the union of the region $y \leq 0$ (representing the ground) and the region $\dot{y} \leq -1.73y$ (representing the vehicle falling in such a way that it will hit the ground in a little over half a second). Because the vehicle's rotational and vertical dynamics are coupled during powered thrust, however, it was first necessary to find a way to decouple them so that safety could be analyzed solely in the vertical state space. This decoupling was accomplished by taking advantage of the fact that the recovery mode was designed to use a fixed control law. As a result, a nominal trajectory with some bounds on deviation could be generated that could then be incorporated into the system dynamics. This allowed the backward reachable set to be computed as usual by propagating it backward for a fixed time τ_r, based on the maximum time that the rotational part of the recovery mode could take. The resulting level set $K_2 = Pre_{\tau_r}(K_3)$ indicates all the configurations from which it would be unsafe for the vehicle to enter the recovery mode.

In the drift mode, the rotational and vertical dynamics decouple, and so the unsafe set for the drift mode, $K_1 = Pre_{\tau_d}(K_2)$ was generated by propagating backward the unsafe set for the recovery mode

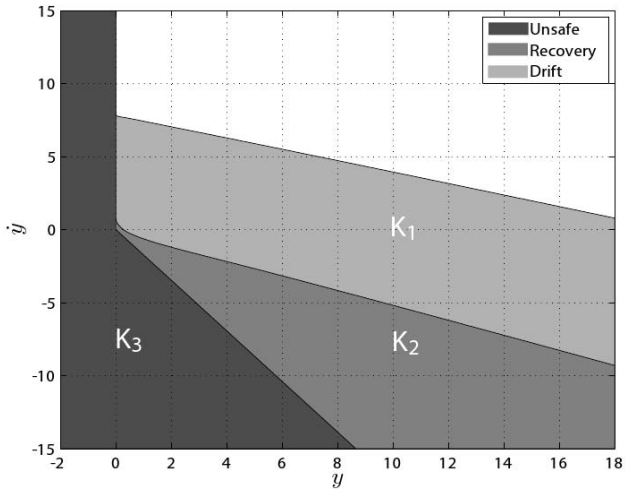


Figure 8: Unsafe vertical sets of the backflip maneuver are plotted in the y (meters) vs $\dot{y}$ (meters/second) plane. As long as the vehicle begins a given mode outside that mode's unsafe set, safety is guaranteed.

using the vertical dynamics. Once again, this was done for a fixed time τ_d, based on the maximum length of the maneuver as calculated from the rotational dynamics. The resulting level set represents all the configurations in which it would be unsafe for the vehicle to enter the drift mode.

For the impulse mode, it was assumed that there would be no loss in altitude because the impulse mode was designed so that the vehicle's thrust would always be upward during this mode. The resulting unsafe sets are pictured in Figure 8; as long as the vehicle began each mode outside the unsafe set for that mode, the overall safety of the system was guaranteed. To ensure that the vehicle began the entire maneuver outside of these unsafe sets, an additional preliminary climb mode was added before the impulse mode, in which the vehicle would accelerate upward until it reached a safe altitude and velocity. The addition of this climb mode guaranteed that as long as the disturbances stayed within the bounds used when calculating the capture and avoid sets, the vehicle would always be able to switch to the next mode when it reached the target set of the current mode.

3.3 Results

A mosaic of one of the demonstrations of the backflip maneuver is shown in Figure 9. Figure 9(a) depicts the quadrotor after the initial climb mode which is the start of the impulse mode, and Figure 9(b) is at the end of the impulse mode and at the beginning of the drift mode. Figures 9(b)-(f) display the entire drift portion of the maneuver and Figure 9(e) shows the quadrotor inverted. Finally, Figures 9(f)-(j) display the recovery mode of the backflip maneuver which successively returns the quadrotor to a safe condition of $\phi = 0^\circ$ and $\dot{\phi} = 0^\circ$/sec. Figure 10 shows the trajectory of the video corresponding to the experimental trial displayed in Figure 9; the labeled points correspond to the frames in the mosaic. Video of the backflip maneuver can be viewed at `http://hybrid.eecs.berkeley.edu/aerobatics.html`.

3.3.1 Attainability: Attitude Results

Figure 11 shows the $(\phi, \dot{\phi})$ trajectory through the designed capture sets for the backflip maneuver. Figure 11(a) shows the simulated trajectory and Figure 11(b) displays three experimental validations. As the figure illustrates, the trajectories are contained within the capture sets for each maneuver. The transition between the impulse and drift modes is denoted by a black diamond, and the transition between the drift and recovery modes is indicated by a black square. The switch between the maneuvers are contained within each of their goal regions, D_1 and D_2, respectively. When the quadrotor switches into the drift mode, it takes approximately 0.2 seconds for the motors to spin down which explains why the quadrotor is still accelerating at the beginning of the drift maneuver. Table 1 displays the time spent in each mode for each experimental validation. Figure 12 displays the pitch of the quadrotor throughout

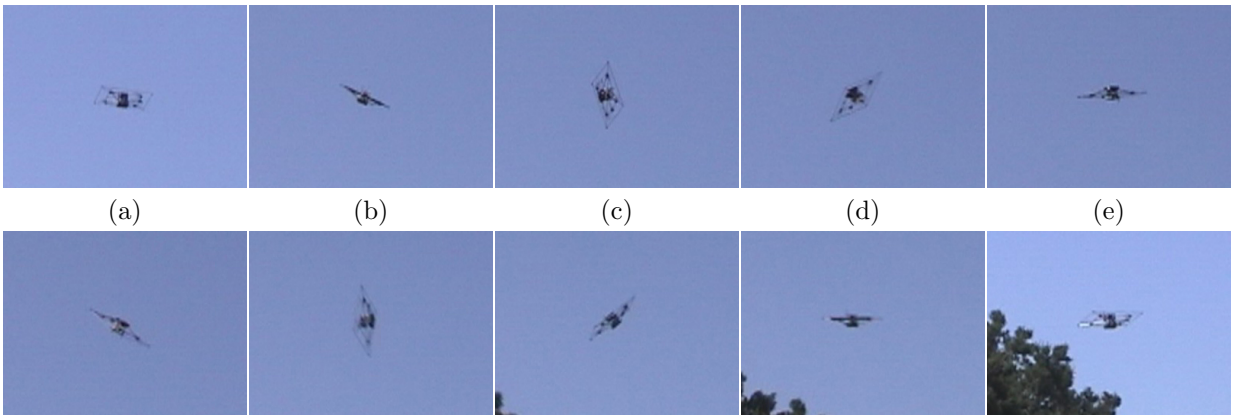


Figure 9: A mosaic of the successful demonstration of the backflip maneuver. (a): The quadrotor has finished the climb portion of the backflip and is starting the impulse mode. (b): The quadrotor has finished the impulse stage and is entering the drift portion. (b)-(f): The drift stage of the backflip. (f): The drift mode is concluding and the recovery mode has started. (f)-(j): The recovery mode is safely returning the quadrotor to its hovering position.

	Impulse (sec.)	Drift (sec.)	Recovery (sec.)
Trial 1	0.55	0.55	2.56
Trial 2	0.50	0.55	5.54
Trial 3	0.55	0.61	4.70

Table 1: The amount of time spent in each mode for each experimental trial of the backflip maneuver.

the maneuver, which is within $\pm 5^\circ$ for almost the entire maneuver. This validates the assumption that the backflip maneuver can be modeled in the 2D $(\phi, \dot{\phi})$ plane.

3.3.2 Safety: Altitude Results

Figure 13 displays the unsafe vertical reachable sets and the switching points for the three experimental validations of the maneuver. The upper (blue) points correspond to entering into the drift mode and the lower (orange) points correspond to entering into the recovery mode. As the figure illustrates, all the points are outside of their respective unsafe set and therefore the vehicle can safely perform the maneuver without hitting the ground.

Finally, it should be noted that while the results of three trials are presented here, several additional trials were also conducted with varying levels of success. However, these failures of the unsuccessful trials were due solely to factors outside the scope of the reachable set analysis. For example, some trials failed because of human error, in the form of bugs in the code running on the vehicle. Others were unsuccessful due to hardware malfunctions (e.g. a broken sonic ranger, or saturation of the IMU's accelerometers). As every roboticist knows, these sorts of failures are typical when making the transition from theory to "real" engineering. However, they also serve a useful purpose: they underscore the importance of understanding the limitations that arise when applying guarantees generated by provably safe techniques to real-world robots.

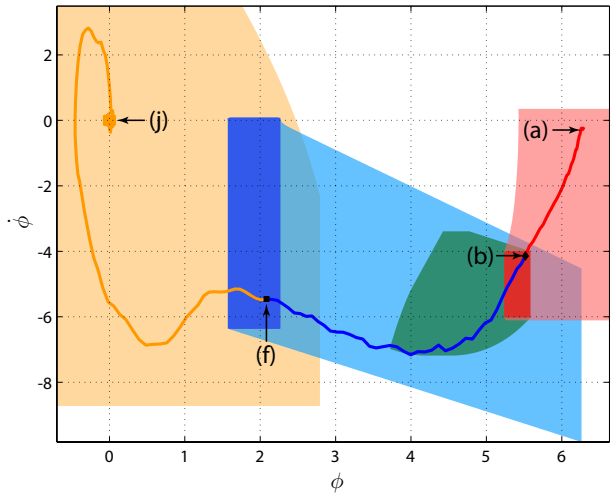


Figure 10: The trajectory of the vehicle corresponding to the mosaic shown in Figure 9. The labeled points correspond to the frames in the mosaic.

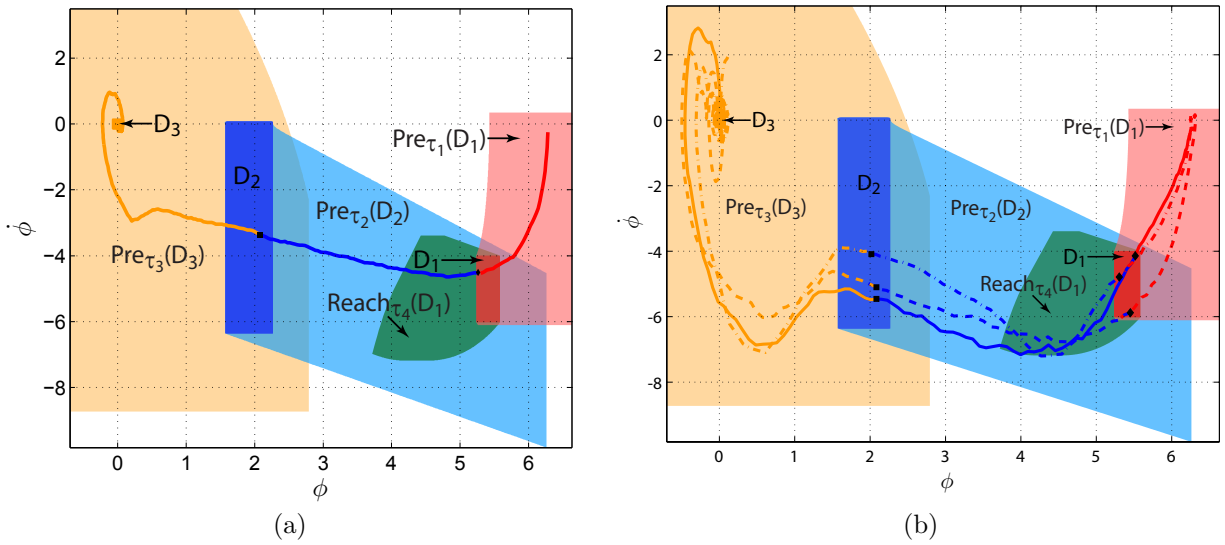


Figure 11: Simulation and experimental validation of the designed backflip maneuver overlaid on the composite reach sets. The transitions from the impulse to drift mode are shown as black diamonds which are contained in region D_1, and the transitions from the drift to the recovery mode are indicated by the black squares that are confined to region D_2. (a) The simulated trajectory of the vehicle. (b) Three experimental validations (solid, dash and dash-dot lines) of the backflip maneuver.

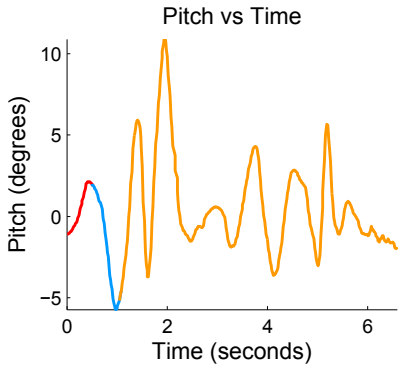


Figure 12: Experimental data from a single run of the backflip maneuver showing the out-of-plane pitch of the quadrotor throughout the maneuver.

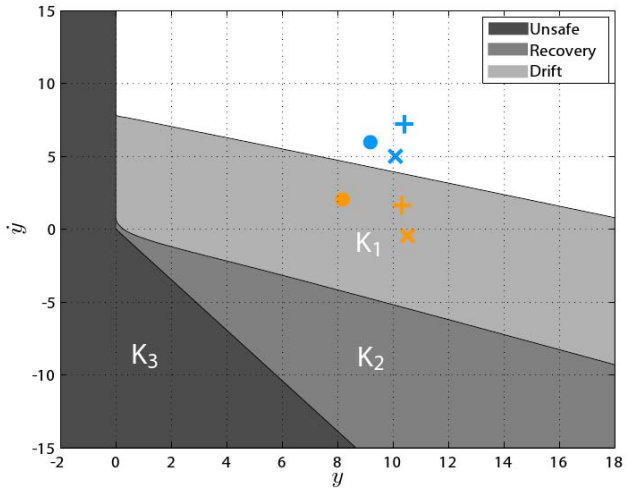


Figure 13: Three experimental validations (×, +, and •) of the backflip maneuver. The light blue symbols (in the white region) correspond to when the vehicle entered into the drift mode and the orange symbols (in the medium-gray region) correspond to when the vehicle entered into the recovery mode. Since all points were outside of their respective reachable set, it was safe for the vehicle to execute the maneuver.

Figure 14: A mid-air collision between two out of three remote-piloted quadrotor helicopters operating in close proximity. The proposed algorithm would assume control when required to prevent imminent collisions.

4 Reachable Sets for Collision Avoidance

Another area where safety must be ensured is in the deployment of multi-agent systems. A motivating application is mobile sensor networks, which can deliver exciting new capabilities in surveillance, reconnaissance, and scientific discovery through their ability to move sensors to vantage points rich in information (Ogren et al. 2004; Grocholsky et al. 2003; Hoffmann et al. 2006). However, as the number of vehicles increases, safety becomes challenging, even for human pilots, as shown anecdotally in Figure 14. Automated collision avoidance algorithms are one way to ensure safety in these types of systems.

Reachable sets are a powerful tool for collision avoidance. In this case, instead of avoiding some region defined in the workspace of a single vehicle, collision avoidance constraints between any given pair of aircraft can be defined in the relative coordinates between the two. For example a constraint requiring some minimum separation between a pair of aircraft can be naturally posed as avoiding some region around the origin in the relative coordinates (Mitchell et al. 2005). There is also a desire to create a minimally constraining controller: that is, the collision avoidance control should only become active when the vehicles are in danger of colliding, but otherwise allow the vehicles to move as their mission requirements dictate. The use of reachable sets is ideally suited to this kind of switching behavior, which is complementary to the ideas of maneuver switching and sequencing discussed in the previous section. This section discusses the use of reachable sets in creating a decentralized multi-vehicle collision avoidance algorithm for a large number of helicopter-like vehicles.

4.1 Related Work

Research has been performed in the literature on flocking and general multi-agent systems, using rule- and optimization-based approaches. One example of a rule-based approach is potential methods. These have been used for an interesting sensor network control application, though without consideration for control input constraints (Ogren et al. 2004). Multi-agent systems with second order dynamics have been studied with a proposed decentralized control law using a detection shell and "gyroscopic" forces (Chang et al. 2003). However, control constraints are not considered, and it is assumed that only one vehicle is in the detection shell at any time. First order dynamics have been used to formulate collision avoidance laws for aircraft, maintaining provable spacing between vehicles (Dimarogonas and Kyriakopoulos 2005). Virtual attractive-repulsive potentials have been used for cooperative control with second order dynamics, though input constraints are not considered (Nguyen et al. 2005). Switching rules have been proposed for decentralized control, but collision avoidance was not considered (Shucker et al. 2007).

Much related work has solved distributed and centralized optimization problems, though tradeoffs are made between computational efficiency for real-time execution and guarantees of collision avoidance. An iterative distributed multi-agent optimization was formulated for general dynamics, (Inalhan et al. 2002) however this exterior point method generates ϵ-feasible solutions, and iterations between vehicles are time consuming as the size of the network increases. A numerical method to ensure aircraft collision avoidance was found using computational geometry, with guarantees when the solution is centralized (Hu

et al. 2003). One approach for distributed collision avoidance is to formulate a mixed integer linear program (MILP) using constraints on speed and acceleration (Schouwenaars et al. 2004). The vehicles are ordered centrally and the optimizations are distributed, though run sequentially. By using loiter patterns, collision avoidance is guaranteed. Developments have been made to simplify the method, though a computationally intensive MILP step is still required (Kuwata and How 2007). A centralized nonlinear program was formulated for aircraft collision avoidance, though the computational expense scales poorly with the number of vehicles (Raghunathan et al. 2004). The multi-agent formation control problem has been decentralized by formulating the dual problem, with provable computational savings, though collision avoidance is not considered (Raffard et al. 2004). Decentralized nonlinear model predictive control was formulated for rotorcraft control, though collision avoidance is only enforced through potential functions, with no guarantees (Shim et al. 2003). Again, while the actual trajectories generated by the reachable set methodology presented here are similar to those generated by other optimization-based techniques in the literature, the reachable set methodology provides both formal guarantees of safety and robustness.

4.2 Collision Avoidance Using Reachable Sets

To address some of the limitations of prior approaches, two new control laws are formulated using reachability for collision avoidance: one for $n_v = 2$ vehicles, and one for $n_v > 2$ vehicles. In both, the vehicles compute analytical avoid set boundaries with respect to every other vehicle. When any vehicles are on the boundary of their avoid sets, collision avoidance action is taken. In the two vehicle scheme, a minimum sized boundary is computed using optimal control that is proven safe analytically, and computed with trigonometric functions. The optimal action is computed numerically. In the $n_v > 2$ vehicle scheme, pairwise avoid sets and actions can be computed by linear algebra with computational complexity $O(n_v)$. Safety is proven analytically for three vehicles, and validated for $n_v \geq 2$ in simulation and analytical analysis of scenarios.

4.2.1 Problem Formulation

Consider a set of n_v vehicles where the state of the i^{th} vehicle is

$$\mathbf{x}_i = \begin{bmatrix} x_i & y_i & \dot{x}_i & \dot{y}_i \end{bmatrix}^T.$$

Note that this formulation neglects potential spatial trajectories in $\mathbb{R}^3$ for aerial, underwater, and space vehicles. The methods developed here can be extended to such scenarios. Also note that for convenience of notation, y refers to a second direction in the horizontal plane, not vertical as in the previous section.

The vehicles are modeled for this application to have undamped second order dynamics with acceleration control inputs $\mathbf{u}_i = \begin{bmatrix} \theta_i & a_i \end{bmatrix}^T$, where the acceleration direction is $\theta_i \in [0, 2\pi)$ and the magnitude is $a_i \in [0, a_{max}]$. This model was found in experiments to adequately approximate the dynamics of quadrotor helicopters (Hoffmann et al. 2008), and is similar to that of many rotorcraft.

To analyze vehicle collision avoidance, define the relative state of vehicle j with respect to i as $\mathbf{x}_{i,j} = \mathbf{x}_j - \mathbf{x}_i$. Define the distance between i and j to be $d_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}$. The collision avoidance requirement is that

$$d_{i,j} \geq d_{min} \ \forall \ \{i, j | i \in [1, n_v],\ j \in [1, n_v],\ j \neq i\} \tag{30}$$

where d_{min} is the minimum allowed distance between vehicle centers. Define the speed of i to be $v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2}$ and the relative speed to be $v_{i,j} = \sqrt{\dot{x}_{i,j}^2 + \dot{y}_{i,j}^2}$. The equations of motion of the relative dynamics for any pair of vehicles are

$$f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) = \frac{\partial}{\partial t} \begin{bmatrix} x_{i,j} \\ y_{i,j} \\ \dot{x}_{i,j} \\ \dot{y}_{i,j} \end{bmatrix} = \begin{bmatrix} \dot{x}_{i,j} \\ \dot{y}_{i,j} \\ -a_i \cos\theta_i + a_j \cos\theta_j \\ -a_i \sin\theta_i + a_j \sin\theta_j \end{bmatrix} \tag{31}$$

where $\mathbf{u}_{i,j} = \begin{bmatrix} \mathbf{u}_i^T & \mathbf{u}_j^T \end{bmatrix}^T$.

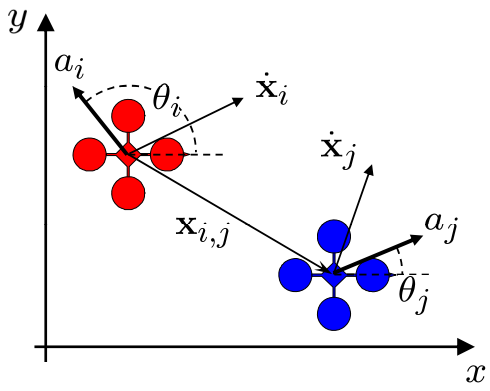


Figure 15: Relative states of two acceleration constrained quadrotor helicopters. The control inputs for aircraft i and j are accelerations $a_i \in [0, a_{max}]$ and $a_j \in [0, a_{max}]$ in directions θ_i and θ_j, respectively.

4.2.2 Optimal Control Approach

The switched control approach is an adaptation of the game formulation posed in Section 2.1, a pursuit-evasion game for non-cooperative control (Tomlin et al. 1998). The goal for safe operation is to prevent the pairwise relative states from entering keepout set K, given by Equation (30). The set $Pre_\tau(K) \subset \mathbb{R}^4$ can be computed from which the control strategy causes a vehicle to enter K in at most τ time. The problem is formulated as a two-person, zero-sum dynamical game, where the "losing" states are calculated for the vehicles. The value function (2) is the cost of a trajectory that starts at $t \leq 0$, evolves according to the dynamics and control inputs, and ends at the final state $\mathbf{x}_{i,j}(0)$. In this case, since the quadrotors are able to come to a halt, an infinite time backwards reachable set $Pre_\infty(K)$ can be calculated, and will be subsequently referred to as simply $Pre(K)$ for simplicity.

An analytic solution is found in Section 4.2.3 for optimal control inputs that are required in a region of minimum size. For more than two vehicles, computing the reachable set becomes computationally expensive. Consequently, a suboptimal control law is presented in Section 4.2.4, inspired by the optimal control law, that has low computational overhead and is trivially decentralized.

4.2.3 Two Vehicle Collision Avoidance

This section presents the optimal switching control law for two vehicle collision avoidance with acceleration constraints. First, the optimal control inputs are derived. Then, a backwards reachable unsafe set $Pre(K)$ is found. Its boundary, $\partial Pre(K)$, is the surface along which vehicles transition from nominal control to collision avoidance.

Optimal Control Input: The final time for the game is the time of closest approach, at $t = 0$. The cost function is,

$$J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t) = l(\mathbf{x}_{i,j}(0)) = d_{i,j}^2 - d_{min}^2. \tag{32}$$

The objective function, the Hamiltonian, is the rate of change of $J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$, given by

$$\frac{\partial J}{\partial t} = \left(\frac{\partial J}{\partial \mathbf{x}_{i,j}}\right)^T \frac{\partial \mathbf{x}_{i,j}}{\partial t} = \mathbf{p}^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) \tag{33}$$

where the costate $\mathbf{p} = \nabla_{\mathbf{x}_{i,j}} J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$, with elements $p_k : k \in [1, 4]$. Expanding it yields

$$\begin{aligned} H(\mathbf{x}_{i,j}, \mathbf{p}) &= \mathbf{p}^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) \\ &= p_1 \dot{x}_{i,j} + p_2 \dot{y}_{i,j} + p_3 \left(-a_1 \cos\theta_i + a_2 \cos\theta_j\right) \\ &\quad + p_4 \left(-a_1 \sin\theta_i + a_2 \sin\theta_j\right). \end{aligned}$$

The optimization problem for evasion-evasion (analogous to pursuit-evasion) is

$$\begin{aligned} &H^*(\mathbf{x}_{i,j}, \mathbf{p}) = \\ &\max_{\mathbf{u_1}} \max_{\mathbf{u_2}}((p_1\dot{x}_{i,j} + p_2\dot{y}_{i,j}) - a_1\left(p_3\cos\theta_i + p_4\sin\theta_i\right) \\ &\qquad\qquad +a_2\left(p_3\cos\theta_j + p_4\cos\theta_j\right)). \end{aligned} \tag{34}$$

To solve for each player separately, first note that at the extrema of the objective, $a_1 = a_2 = a_{max}$. Then, the derivative is taken with respect to the remaining control inputs. Consider the perspective of vehicle j,

$$\frac{\partial}{\partial\theta_j}\left(p_3\cos\theta_j + p_4\sin\theta_j\right) = -p_3\sin\theta_j + p_4\cos\theta_j. \tag{35}$$

The extrema, then, are the solutions to

$$-p_3\sin\theta_j^* + p_4\cos\theta_j^* = 0. \tag{36}$$

Thus,

$$\theta^* = \arctan\left(\frac{p_4}{p_3}\right) + n\pi. \tag{37}$$

From the calculus of variations,

$$\dot{\mathbf{p}} = -\frac{\partial}{\partial\mathbf{x}_{i,j}}H^*(\mathbf{x}_{i,j}, \mathbf{p}) = \begin{bmatrix} 0 & 0 & -p_1 & -p_2 \end{bmatrix}^T \tag{38}$$

and

$$\mathbf{p}(0) = \frac{\partial}{\partial\mathbf{x}_{i,j}}J(\mathbf{x}_{i,j}(0)) = \begin{bmatrix} 2x_{i,j}(0) & 2y_{i,j}(0) & 0 & 0 \end{bmatrix}^T. \tag{39}$$

Then, integrating to find the optimal control inputs yields

$$\mathbf{p}(t) = \begin{bmatrix} 2x_{i,j}(0) \\ 2y_{i,j}(0) \\ -\int_t^0 p_1 dt \\ -\int_t^0 p_2 dt \end{bmatrix} = \begin{bmatrix} 2x_{i,j}(0) \\ 2y_{i,j}(0) \\ -tx_{i,j}(0) \\ -ty_{i,j}(0) \end{bmatrix}. \tag{40}$$

Substituting (40) into (37) yields the optimal input,

$$\theta_j^* = \arctan\left(\frac{y_{i,j}(0)}{x_{i,j}(0)}\right) + n\pi. \tag{41}$$

Substituting (41) into (34) it is found that $n = 1$ maximizes the optimization. Thus, it is optimal to accelerate away from the point of closest approach on ∂K.

Avoid Set: The boundary of the avoid set is the locus of points along which collision avoidance control is required. The collision avoidance control input is constant acceleration perpendicular to ∂K at the point of closest approach, so the path in relative coordinates is a parabola, as depicted in Figure 16.

For analysis, a change of coordinates is performed. The change of coordinates rotates the relative coordinate frame such that in the rotated frame, $\dot{x}_{i,j} = 0$ and $\dot{y}_{i,j} < 0$. Define the rotation angle of the avoid region, with respect to the relative coordinate frame, to be[4]

$$\phi_{i,j} = \text{atan2}\left(\dot{y}_{i,j}, \dot{x}_{i,j}\right) - \frac{\pi}{2} \tag{42}$$

where the $\frac{\pi}{2}$ offset orients the avoid set along the $y_{i,j}$ axis for analysis; an arbitrary choice. Analysis of the avoid set for any $\phi_{i,j}$ can be performed in the coordinates shown in Figure 16, removing the need to consider $\dot{x}_{i,j}$ in this frame.

For the remainder of this section, the relative coordinate frame is rotated by $-\phi_{i,j}$. To avoid over complicating the notation, the same variable names are used for the rotated frame; to return results to the original coordinate frame, they must be rotated by $\phi_{i,j}$, as depicted in Figure 18.

[4]As in ANSI-C, $\text{atan2}\left(y, x\right)$ is similar to $\arctan\left(\frac{y}{x}\right)$, but by using the signs of x and y, it returns the angle in the domain $(-\pi, \pi]$ rather than $(-\frac{\pi}{2}, \frac{\pi}{2}]$.

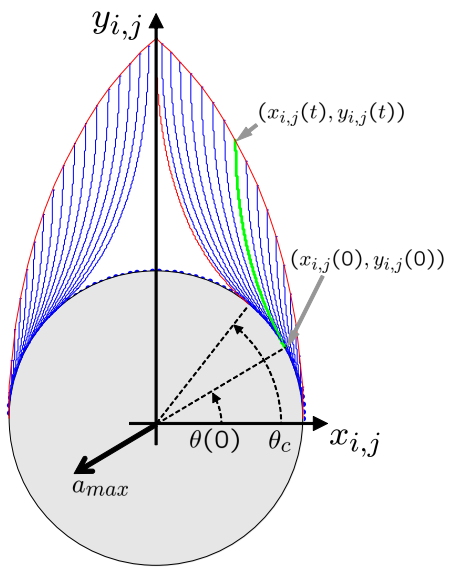


Figure 16: The keepout set K (gray) and the avoid set $Pre(K)$ (white) for two-vehicle collision avoidance, in the rotated coordinate frame. The parabolas are the trajectories that a vehicle follows from any point on the boundary. The avoid set is time varying and never crossed.

The boundary $\partial Pre_(K)$ of the avoid set can be found in the rotated frame, as shown in Figure 16, such that use of the optimal control input results in a closest approach of d_{min}. The conditions of the rotated frame can be used: $\dot{x}_{i,j} = 0$, $\dot{y}_{i,j} = -v_{i,j}$. The parabola is rotated by $\theta(0) - \pi/2$, and has a second derivative in that direction of $2a_{max}$, resulting from collision avoidance action by both vehicles. Thus, the usable part of $\partial Pre(K)$ is defined by,

$$x_{i,j}(t) = d_{min}\cos\theta(0) - \tfrac{v_{i,j}(t)^2}{2a_{max}}\sin^2\theta(0)\cos\theta(0) \quad (43)$$

$$y_{i,j}(t) = \left(d + \tfrac{v_{i,j}(t)^2}{a_{max}}\right)\sin\theta(0) - \tfrac{v_{i,j}^2}{2a_{max}}\sin^3\theta(0). \quad (44)$$

To numerically test if vehicle j has crossed the boundary, the region can be approximated by a polygon with vertices generated using discrete values of $\theta(0)$ from zero to the critical angle, θ_c, and then mirroring about the $y_{i,j}$-axis. The angle θ_c is the one past which no points of closest approach occur, given the use of optimal control. It can be shown that $\theta_c < \frac{\pi}{2}$ if $2d_{min}a_{max} < v_{i,j}^2$. To find θ_c, (43) is solved with $x_{i,j} = 0$,

$$\theta_c = \begin{cases} \arcsin\sqrt{\frac{2d_{min}a_{max}}{v_{i,j}(t)^2}} & \text{if } 2d_{min}a_{max} < v_{i,j}^2 \\ \pi/2 & \text{otherwise} \end{cases} \quad (45)$$

Then, ($x_{i,j}$, $y_{i,j}$) can be tested to see if it is in the polygon.[5] If it is, then collision avoidance action must be taken.

To find the optimal control, the point of closest approach must be found. This can be found by solving (44) for $\theta(0)$ using the current value of $y_{i,j}$. Note that by substituting $\zeta = \sin\theta(0)$, this is a cubic function, with roots easily found numerically; the only physical root is in the domain $[-1, 1]$.

Note that synchronous control is not required, though to strictly avoid entering K, $Pre_(K)$ must be grown to account for all possible control inputs. When implemented in discrete time, the control law must test if the vehicles are at $\partial Pre_(K)$, which requires crossing the boundary of the avoid set by at most $v_{i,j}\Delta t$, where Δt is the discrete time step. To guarantee strict avoidance, d_{min} must be increased by this quantity.

[5]Standard algorithms can test if ($x_{i,j}$, $y_{i,j}$) is in the polygon, though here it is more efficient to use a cross product test with the points on the boundary horizontally closest to $x_{i,j}$.

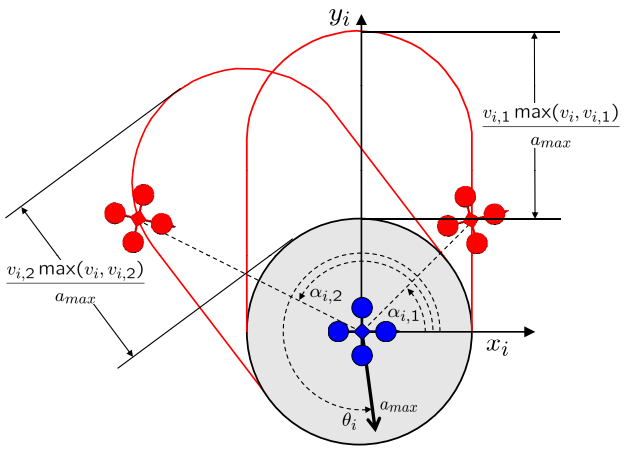


Figure 17: Vehicle v accelerates at a_{max} in direction θ_v to avoid vehicles 1 and 2, which are at the edge of their respective avoid sets. The orientation of the avoid set is that of the relative velocity vectors.

The optimal collision avoidance action is to accelerate away from $\theta(0)$, or in the original relative coordinate frame, $\theta(0) + \phi_{i,j}$. This control law guarantees that no collisions will occur between two vehicles, and is required for a minimum portion of the physical area. However to address many interacting vehicles further development is required, and is considered next.

4.2.4 Multi-Vehicle Collision Avoidance

Optimal Control Input: Following the same methods used to derive (41) more vehicle interactions can be included, yielding the optimal control law for many vehicles,

$$\theta_i^* = \arctan\left(\frac{y_{i,1}(0) + y_{i,2}(0) + \dots}{x_{i,1}(0) + x_{i,2}(0) + \dots}\right) + \pi. \tag{46}$$

However it is computationally expensive to find $\mathbf{x}_{i,j}(0)$ for $n_v > 2$, so this control law will not be used used here. Rather, it is used to inspire a proposed alternate control strategy,

$$\theta_i = \arctan\left(\frac{y_{i,1}(t) + y_{i,2}(t) + \dots}{x_{i,1}(t) + x_{i,2}(t) + \dots}\right) + \pi \tag{47}$$

where the vehicles considered are those at the boundaries of their respective avoid sets. Note that these sets, defined as $A_{i,j}$, will no longer be the exact infinite-time backwards reachable set $Pre(K)$ for each pair. Again, use $a_i = a_{max}$. The control law is equivalent to accelerating away from the centroid of all vehicles that must be actively avoided. Additional logic is required as described below.

This control law is suboptimal in the sense that its use may be required in regions for which another collision avoidance control law would not require action. However, it yields enormous computational savings, and will be shown to yield collision avoidance using a reasonably small avoid set. Now the avoid sets must be found.

Avoid Set: The separate avoid sets for each vehicle are shown in Figure 17. They are inspired by the avoid set for two vehicle collision avoidance, and are again aligned with the relative velocity direction. The length of the proposed region is

$$L_{i,j} = \frac{v_{i,j}\max\left(v_i, v_{i,j}\right)}{a_{max}} \tag{48}$$

in order to maintain the minimum separation distance. This avoid set is designed such that the control law provably works for two and three vehicles. It is validated analytically and numerically for $n_v > 3$ in selected configurations assumed to be most challenging to the algorithm.

Table 2: Collision Avoidance Computation Time

Algorithm	# Vehicles	Time (ms)
Two-Veh.	2	0.23
Many-Veh.	2	0.16
Many-Veh.	8	0.41
Many-Veh.	32	1.6
Many-Veh.	64	3.6
Many-Veh.	100	4.6
Many-Veh.	200	9.2

The analytical verification of the dynamics of up to three vehicles on their respective switching boundaries $\partial A_{i,j}$ shows that they do not cross into $A_{i,j}$, and because $K \subset A_{i,j}$, this demonstrates that no vehicle can enter K. The nonlinear dynamics proofs were detailed in Hoffmann and Tomlin (Hoffmann and Tomlin 2008). For more than three vehicles, two configurations were analyzed with the potential to cause failure: a large circle of vehicles converging toward one point within the circle, and a large line of vehicles converging toward one point on the line. The algorithm was proven analytically to guarantee that separation is maintained in both cases, as detailed in (Hoffmann and Tomlin 2008). Next, simulation and experimental results are presented.

4.3 Results

The proposed control laws were tested in simulation and flight experiments. All simulated vehicles are quadrotor helicopters with second order dynamics and ℓ^2-norm constraints on acceleration. Simulations were run in Matlab using one core of a Core 2 Duo 2.16 GHz processor. The number of vehicles demonstrated would be computationally overwhelming for other methods known to the authors. The timing results for the simulations are shown in Table 2. These control laws are applied in simulations of 2 to 200 vehicles, and demonstrated in flight experiments. Run time for two simulated vehicles was 0.2 ms. Run time for 200 interacting simulated vehicles was 9.2 ms.

Flight experiments used STARMAC quadrotor helicopters tracking attitude commands from human pilots (Hoffmann et al. 2008). These experiments validated the results from simulations. Note that the collision avoidance algorithm only prevents violations of keepout sets. If the nominal control law repeats similar inputs that lead to the avoidance action, repeated switching between avoidance and nominal control actions may ensue. However, in both simulations and flight experiments, this switching did not lead to deadlock. Deadlock would require high symmetry to numerical precision and noise-free sensing and actuation—an improbable condition in real systems, as observed in numerous flight experiments.

4.3.1 Two Vehicles

The two vehicle collision avoidance algorithm was used for two trajectory tracking vehicles flown toward each other with a variety of crossing paths, speeds, and acceleration constraints. The circular set K and avoid set $Pre(K)$ are shaded in Figure 18. As the control action is taken, the relative velocity changes. This causes the avoid sets to morph, keeping the vehicles on each other's boundaries. The vehicles never enter the avoid set; the avoidance action is required while the vehicles are on each other's boundaries. At the end of the avoidance maneuver, they resume line tracking.

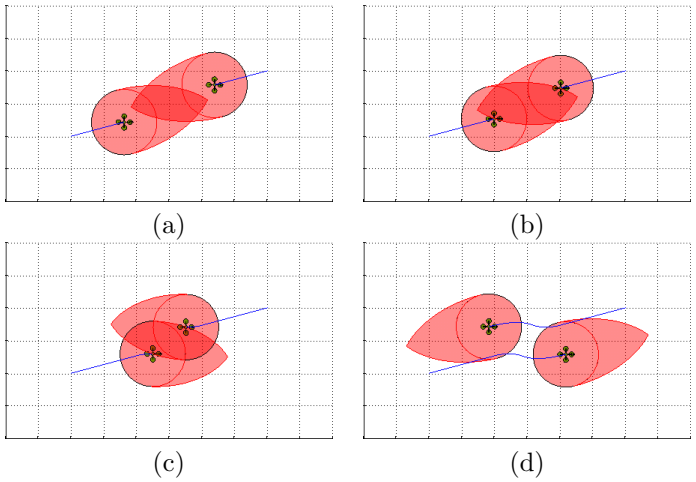


Figure 18: Simulation of two quadrotor helicopters approaching one another while tracking trajectories (a). When the vehicles touched each other's avoid set (b), the sets extending from the circular unsafe sets, they applied collision avoidance control inputs (c). After the conflict was resolved, they resumed their previous trajectories (d).

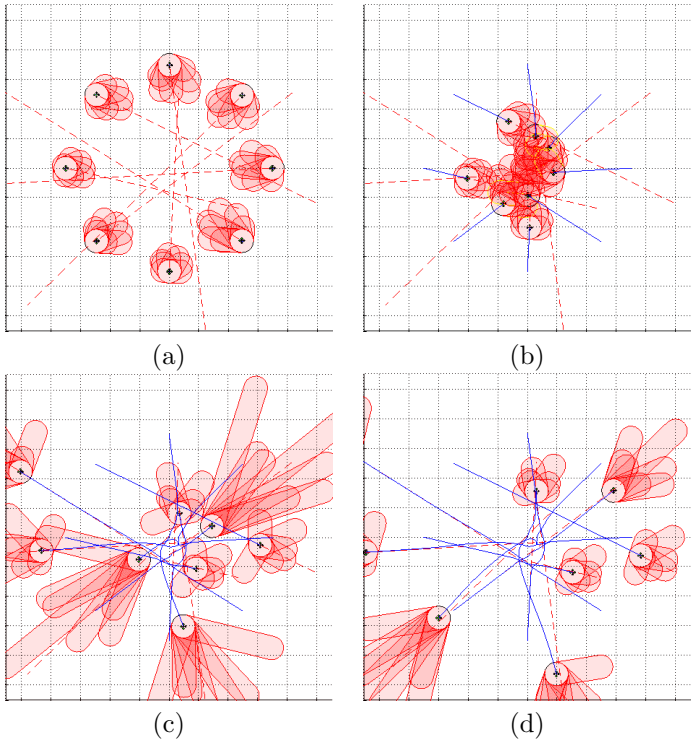


Figure 19: Simulation of 8 vehicles tracking randomly generated trajectories, using the many-vehicle collision avoidance law. The circles are keepout sets K, and shaded long regions are avoid sets $A_{i,j}$. Only a few regions touch the vehicles to which they correspond, leading to collision avoidance action, as highlighted in (b).

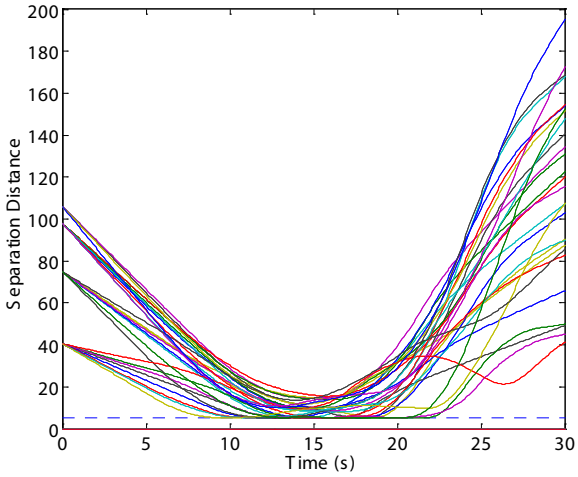
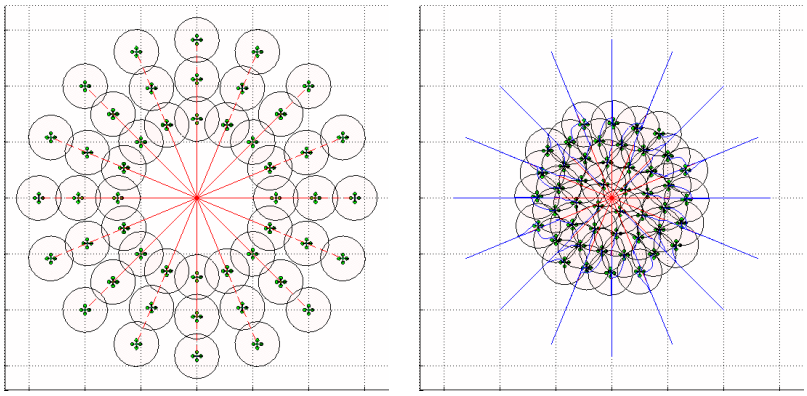


Figure 20: Separation distances between each vehicle for the eight vehicle simulation in Figure 19, and a line showing the minimum allowed distance, $d_{min} = 2$. Separation was maintained throughout the simulation.



(a) (b)

Figure 21: Simulation of 3 rings of 16 vehicles converging toward the same point, with the outermost rings moving fastest, at $t = 0$ (a) and at $t = 10$ (b). Many avoid sets were active for each vehicle due to the range of speeds of the rings.

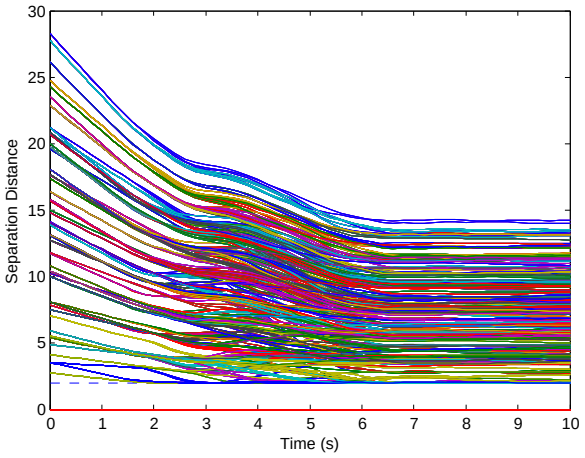


Figure 22: Separation distances between each vehicle for the 48 vehicle simulation in Figure 21, and a line at the minimum allowed distance, $d_{min} = 2$. Separation was maintained throughout the simulation.

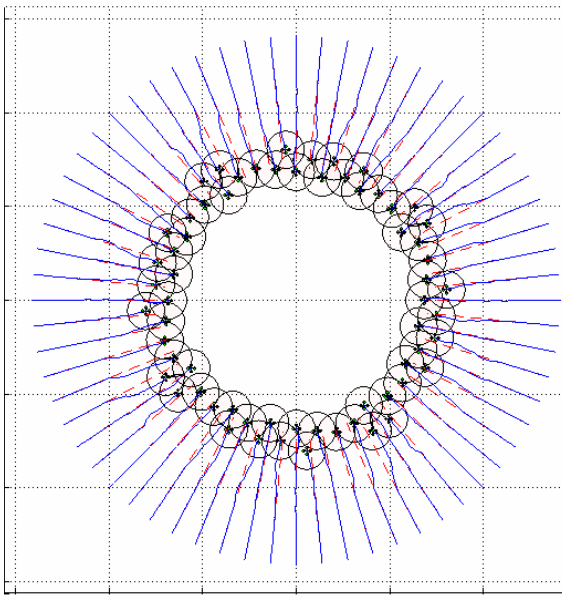


Figure 23: Simulation of a ring of 64 vehicles converging toward a center. The many-vehicle collision avoidance algorithm safely prevented the vehicles from being wedged together.

4.3.2 Many Vehicles

Sets of 2 to 200 vehicles were simulated doing trajectory tracking using many-vehicle collision avoidance on a variety of collision-courses. One scenario is in Figure 19, with the separation distance between each vehicle in Figure 20. The vehicles navigate past one another as the trajectory tracking control is allowed to resume. Timing from these simulations, in Table 2, shows that run time at each vehicle was approximately $n_v \times 0.05$ ms.

Many challenging scenarios were simulated to validate performance in unreasonably complicated situations. One such case shown in Figure 21 has three rings of 16 vehicles converging, with the outermost rings traveling faster. The $n_v - 1$ avoid sets at each vehicle are omitted from the plot for clarity. Collision avoidance must prevent lines of vehicles from piling up and rings of vehicles being wedged together. Spacing is maintained as shown in Figure 20. Due to numerical precision, the vehicles are pushed off of their trajectories and approach their destination, the origin.

Another challenging case is a large circle of vehicles converging, as shown in Figure 23, where the time history of relative distances again verified that separation was maintained. The vehicles eventually slide past one another and follow their trajectories. In both the 48 and 64 vehicle scenarios, the near-perfect symmetry and lack of noise led to the vehicles coming to a near stop temporarily, though the right-of-way was resolved eventually due to numerical noise from machine precision. When switching rapidly between avoidance and nominal control inputs the system maintained separation, as was the case for all simulations.

4.3.3 Flight Experiments

The many-vehicle algorithm was implemented in the onboard computers of the STARMAC testbed. The software on each quadrotor broadcasts vehicle states to all other quadrotors at 10 Hz. The algorithm uses $d_{min} = 2$ m, $a_{max} = 1.7\frac{m}{s^2}$, and runs on the 600 MHz PXA270 processor on board the aircraft.

The vehicles nominally used either attitude reference commands from a human pilot, or a waypoint tracking controller. The human pilots issued malicious commands to attempt to instigate collisions, which the collision avoidance algorithm successfully prevented, as shown in Figures 24, 25, and 26. Experiments were flown using both 2- and 4-vehicle fleets, demonstrating the expected performance in both cases. The effect of time discretization was not included in d_{min} for the flight software implementation due to the fast update rate relative to vehicle speeds. In all but one instance, the minimum separation distance was maintained. One incident occurred where it was violated by 0.1 m when $v_{1,2} = 5\frac{m}{s}$. This is less than the $v_{1,2}\Delta t$ safety margin required for time discretization, verifying the proposed approach.

5 Conclusion

The combination of hybrid decomposition and reachable set theory is a powerful tool for analysis, design and verification of complex control problems. By using these analytic tools, provable guarantees on safety and performance can be made for complicated platforms such as the STARMAC quadrotor helicopter. In the aerobatic maneuver work, reachable set analysis allowed the design of a sequence of modes that could be guaranteed to safely transition from one to the next, arriving at a desired final state while avoiding an undesired region of the state space. The flight experiments on the STARMAC platform demonstrated this provably safe backflip maneuver and showed the validity of using reachability for maneuver design.

Similarly, the collision avoidance results showed the flexibility and utility of reachable sets as applied to multi-vehicle systems. Although an analytic proof of safety is not yet available for the case where $n_v > 3$, the system has been shown to be effective and safe in a number of simulation and flight experiments. The same concepts used in the design of the backflip and collision avoidance system can be extended to creating provably safe sequences of maneuvers for aerial vehicles or generalized to building safe mode sequences for other complex systems. Provable guarantees are valuable in the design of any system, and hybrid reachability holds great promise for further development of such guarantees for many complex systems.

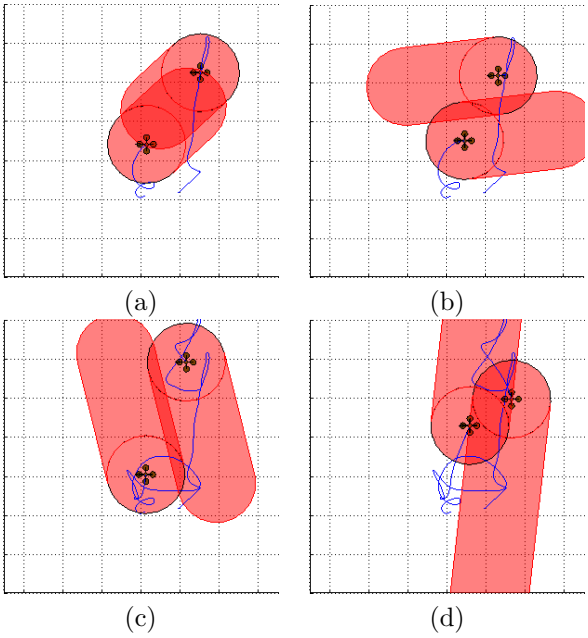


Figure 24: Automatic collision avoidance flight experiment using $d_{min} = 2$ m with human control inputs attempting to cause collisions. A conflict was detected at $t = 25$ s (a) and recovered from by $t = 26$ s (b). The aircraft approached at $t = 35$ s (c), resulting in the conflict at $t = 36$ s (d).

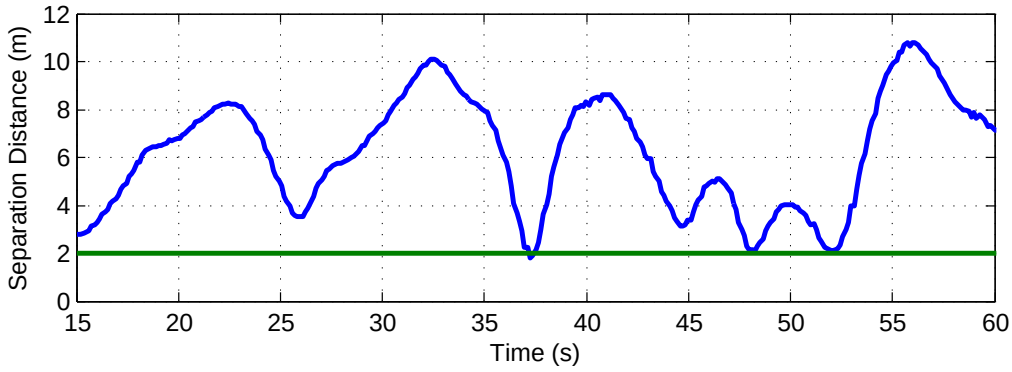


Figure 25: Separation according to GPS data for the flight experiment shown in Figure 24. Even without extending d_{min} to account for time discretization, at $v_{1,2} = 5\frac{m}{s}$, d_{min} was violated by 0.1 m, less than $v_{1,2}\Delta t$.

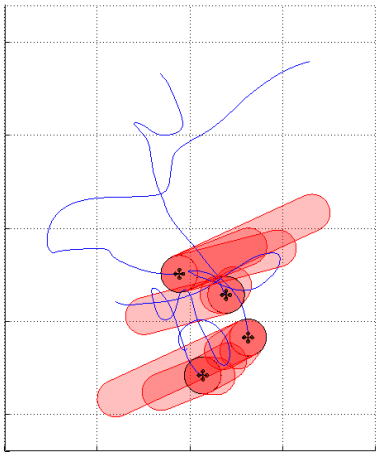


Figure 26: Automatic collision avoidance flight with four quadrotor helicopters. Note that pairwise regions with opposite orientations remained unviolated, regardless of the pilots' attempts. The vehicles were able to fly within 2 meters of each other, center to center, without concern on the part of the pilots.

Acknowledgements

The authors would like to thank Steven Waslander, Vijay Pradeep and Tony Mercer for their assistance in performing the flight tests presented in this work.

Funding

This research was supported in part by the "CoMotion: Computational Techniques for Collaborative Motion" project administered by the ONR under MURI grant #N00014-02-1-0720, in part by the "Integrating Collision Avoidance and Tactical Air Traffic Control Tools" project administered by NASA under grant #NNA06CN22A, and in part by the "Frameworks and Tools for High Confidence Design of Adaptive, Distributed Embedded Control Systems" project administered by the AFOSR under MURI grant #FA9550-06-0312. Additionally, the authors would like to thank the NSF and the ASEE for their support via graduate fellowships.

References

Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. In *Advances in Neural Information Processing Systems 19*, pages 1–8. MIT Press.

Bardi, M. and Capuzzo-Dolcetta, I. (1997). *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman equations*. Birkhäuser, Boston.

Bardi, M., Falcone, M., and Soravia, P. (1999). Numerical methods for pursuit-evasion games and viscosity solutions. In Bardi, M., Parthasarathy, T., and Raghavan, T. E. S., editors, *Stochastic and Differential Games: Theory and Numerical Methods*, volume 4 of *Annals of Int. Society of Dynamic Games*, pages 105–175. Birkhäuser, Boston.

Burridge, R., Rizzi, A., and Koditschek, D. (1999). Sequential composition of dynamically dexterous robot behaviors. *Int. J. of Robotics Research*, 18(6):534–555.

Cardaliaguet, P., Quincampoix, M., and Saint-Pierre, P. (1999). Set-valued numerical analysis for optimal control and differential games. In Bardi, M., Parthasarathy, T., and Raghavan, T. E. S., editors, *Stochastic and Differential Games: Theory and Numerical Methods*, volume 4 of *Annals of Int. Society of Dynamic Games*, pages 177–248. Birkhäuser, Boston.

Chang, D. E., Shadden, S. C., Marsden, J. E., and Olfati-Saber, R. (2003). Collision avoidance for multiple agent systems. In *Proc. 42nd IEEE Conf. Decision and Control*, pages 539–543, Maui, HI.

Coates, A., Abbeel, P., and Ng, A. Y. (2009). Apprenticeship learning for helicopter control. *Commun. ACM*, 52(7):97–105.

Dimarogonas, D. V. and Kyriakopoulos, K. J. (2005). Decentralized stabilization and collision avoidance of multiple air vehicles with limited sensing capabilities. In *Proc. 2005 American Control Conf.*, pages 4667–4672, Portland, OR.

Ding, J., Sprinkle, J., Sastry, S. S., and Tomlin, C. J. (2008). Reachability calculations for automated aerial refueling. In *Proc. 47th IEEE Conf. Decision and Control*, pages 3706–3712, Cancun, Mexico.

Egerstedt, M., Koo, T. J., Hoffmann, F., and Sastry, S. (1999). Path planning and flight controller scheduling for an autonomous helicopter. In *Proc. 2nd International Conf. on Hybrid Systems: Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*, pages 91–102, Berg en Dal, The Netherlands. Springer-Verlag.

Frazzoli, E., Dahleh, M. A., and Feron, E. (2005). Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Transactions on Robotics*, 21(6):1077–1091.

Gavrilets, V., Martinos, I., Mettler, B., and Feron, E. (2002). Flight test and simulation results for an autonomous aerobatic helicopter. In *Proc. 21st Digital Avionics Systems Conf.*, pages 8.C.3/1–6. DOI: 10.1109/DASC.2002.1052943.

Gillula, J. H., Huang, H., Vitus, M. P., and Tomlin, C. J. (2009). Design and analysis of hybrid systems with applications to robotic aerial vehicles. In *Proc. 14th Int. Symposium of Robotics Research*, Lucerne, Switzerland.

Gillula, J. H., Huang, H., Vitus, M. P., and Tomlin, C. J. (2010). Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and practice. In *Proc. 2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK.

Grocholsky, B., Makarenko, A., Kaupp, T., and Durrant-Whyte, H. (2003). Scalable control of decentralised sensor platforms. In Zhao, F. and Guibas, L. J., editors, *Proceedings of the 2nd Int. Workshop on Information Processing in Sensor Networks*, volume 2003, pages 96–112, Palo Alto, CA. Springer.

Hoffmann, G. M., Huang, H., Waslander, S. L., and Tomlin, C. J. (2007). Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proc. 2007 AIAA Guidance, Navigation, and Control Conf.*, Hilton Head, SC.

Hoffmann, G. M. and Tomlin, C. J. (2008). Decentralized cooperative collision avoidance for acceleration constrained vehicles. In *Proc. 47th IEEE Conf. Decision and Control*, pages 4357–4363, Cancun, Mexico.

Hoffmann, G. M. and Tomlin, C. J. (2010). Mobile sensor network control using mutual information methods and particle filters. *IEEE Transactions on Automatic Control*, 55(1):32–47.

Hoffmann, G. M., Waslander, S. L., and Tomlin, C. J. (2006). Distributed cooperative search using information-theoretic costs for particle filters with quadrotor applications. In *Proc. 2006 AIAA Guidance, Navigation, and Control Conf.*, pages 21–24, Keystone, CO.

Hoffmann, G. M., Waslander, S. L., and Tomlin, C. J. (2008). Quadrotor helicopter trajectory tracking control. In *Proc. 2008 AIAA Guidance, Navigation, and Control Conf.*, Honolulu, HI.

Hu, J., Prandini, M., and Sastry, S. (2003). Optimal coordinated motions of multiple agents moving on a plane. *SIAM J. Control and Optimization*, 42(2):637–668.

Inalhan, G., Stipanovic, D. M., and Tomlin, C. J. (2002). Decentralized optimization, with application to multiple aircraft coordination. In *Proc. 41st IEEE Conf. Decision and Control*, pages 1147–1155.

Kress-Gazit, H., Fainekos, G. E., and Pappas, G. J. (2008). Translating structured english to robot controllers. *Advanced Robotics Special Issue on Selected Papers from IROS 2007*, 22(12):1343–1359.

Kurzhanski, A. B. and Varaiya, P. (2002). Reachability analysis for uncertain systems – the ellipsoidal technique. *Dynamics of Continuous, Discrete, and Impulsive Systems*, 9(3):347–367.

Kuwata, Y. and How, J. P. (2007). Robust cooperative decentralized trajectory optimization using receding horizon MILP. In *Proc. 2007 American Control Conf.*, pages 522–527, New York, NY.

Lazar, M. and Jokic, A. (2009). Synthesis of trajectory-dependent control lyapunov functions by a single linear program. In *Proc. 12th Int. Conf. on Hybrid Systems: Computation and Control*, volume 5469 of *Lecture Notes in Computer Science*, pages 237–251, San Francisco, CA. Springer-Verlag.

Lozano-Perez, T., Mason, M. T., and Taylor, R. H. (1984). Automatic synthesis of fine-motion strategies for robots. *Int. J. of Robotics Research*, 3(1):3–24.

Lupashin, S., Scho?llig, A., Sherback, M., and D'Andrea, R. (2010). A simple learning strategy for high-speed quadrocopter multi-flips. In *Proc. 2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK.

Lygeros, J., Tomlin, C., and Sastry, S. (1999). Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370.

Mitchell, I. M. (2009). *A Toolbox of Level Set Methods*. `http://people.cs.ubc.ca/~mitchell/ToolboxLS/index.html`.

Mitchell, I. M., Bayen, A. M., and Tomlin, C. J. (2005). A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957.

Nguyen, B. Q., Chuang, Y.-L., Tung, D., Hsieh, C., Jin, Z., Shi, L., Marthaler, D., Bertozzi, A., and Murray, R. M. (2005). Virtual attractive-repulsive potentials for cooperative control of second order dynamic vehicles on the Caltech MVWT. In *Proc. 2005 American Control Conf.*, volume 2, pages 1084–1089, Portland, OR.

Ogren, P., Fiorelli, E., and Leonard, N. E. (2004). Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302.

Oishi, M., Mitchell, I. M., Bayen, A. M., and Tomlin, C. J. (2008). Invariance-preserving abstractions of hybrid systems: Application to user interface design. *IEEE Transactions on Control Systems Technology*, 16(2):229–244.

Purwin, O. and DAndrea, R. (2009). Performing aggressive maneuvers using iterative learning control. In *Proc. 2009 IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan.

Raffard, R., Tomlin, C. J., and Boyd, S. P. (2004). Distributed optimization for cooperative agents: Application to formation flight. In *Proc. 43rd IEEE Conf. Decision and Control*, pages 2453–2459, Atlantis, Bahamas.

Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T., and Samad, T. (2004). Dynamic optimization strategies for three-dimensional conflict resolution of multiple aircraft. *AIAA J. Guidance, Control, and Dynamics*, 27(4):586–594.

Schouwenaars, T., How, J., and Feron, E. (2004). Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees. In *Proc. 2004 AIAA Guidance, Navigation, and Control Conf.*, Providence, RI.

Shim, D. H., Kim, H. J., and Sastry, S. (2003). Decentralized nonlinear model predictive control of multiple flying robots. In *Proc. 42nd IEEE Conf. Decision and Control*, pages 3621–3626, Maui, HI.

Shucker, B., Murphey, T., and Bennett, J. K. (2007). Switching rules for decentralized control with simple control laws. In *Proc. 2007 American Control Conf.*, pages 1485–1492, New York, NY.

Tomlin, C., Lygeros, J., and Sastry, S. (July 2000). Controller design for hybrid systems. *Proceedings of the IEEE*, 88(7).

Tomlin, C. J. (2009). Controllers for safety of continuous systems. *EECS 291 Hybrid Systems Lecture Notes*, (9). `http://inst.eecs.berkeley.edu/~ee291e/sp09/`.

Tomlin, C. J., Pappas, G. J., and Sastry, S. (1998). Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521.

Zefran, M. and Burdick, J. W. (1998). Stabilization of systems with changing dynamics. In *Proc. 1st Int. Conf. on Hybrid Systems: Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*, pages 400–415, Berkeley, CA. Springer-Verlag.