# Toward Reachability-Based Controller Design for Hybrid Systems in Robotics

Jerry Ding†, *Student Member, IEEE*, Jeremy H. Gillula†, *Student Member, IEEE*, Haomiao Huang†, *Student Member, IEEE*, Michael P. Vitus†, *Student Member, IEEE*, Wei Zhang†, *Member, IEEE*, and Claire J. Tomlin, *Fellow, IEEE*

***Abstract*—Modern robotic systems are becoming increasingly complex and more capable of accomplishing sophisticated tasks. Such systems usually involve active interactions with the environment and event-driven changes in operating modes that are described well by hybrid dynamical systems. This paper focuses on verification and controller synthesis for such complex robotic systems in safety-critical applications. The design goal for most of these applications can be formulated as one of driving the system state into a target set without hitting some unsafe regions in the state space during the process. We present a methodology for achieving such a design goal based on reachability computations for hybrid systems. The approach is demonstrated on several robotic applications, and a variety of promising future research directions are identified and discussed in detail.**

## I. Introduction

Robotics has provided the motivation and inspiration for many innovations in planning and control. From non-holonomic motion planning [1] to probabilistic road maps [2], from capture basins [3] to pre-images [4] of obstacles to avoid, from geometric nonlinear control [5], [6] to machine learning methods in robotic control [7], there is a wide range of planning and control algorithms and methodologies which can be traced back to a perceived need or anticipated benefit in autonomous or semi-autonomous systems.

Our research has been similarly inspired by problems in autonomous and semi-autonomous systems. We have focused on safety verification and controller synthesis to satisfy safety specifications, as well as verification and controller synthesis for guaranteeing that desired targets are reached. Among a group of researchers at the interface of robotics, control, and computer-aided verification, we have been interested in particular in safety critical systems in which the automation, or automation combined with human control, is responsible for keeping the system state inside a region of the state space defined to be safe, as well as inside a region that is guaranteed to reach the desired target. For example, a tactical collision avoidance system for manned aircraft and UAVs in a Next Generation Air Traffic Control system must ensure that all pairs of aircraft remain separated by a specified safety distance, under reasonable assumptions on the relative aircraft configurations and individual aircraft actions. At a lower level, an aircraft autopilot must guarantee that the aircraft state remains within the aerodynamic flight envelope of the aircraft, despite changes in set-points and commands from higher-level controllers. Steps have been made toward solving these problems by determining how to compute so called reachable sets for dynamic systems, which are sets of initial states from which the system is guaranteed to remain inside a safe region, while eventually reaching a desired target [8], [9], [10]. In addition, computational advances have been made in finding efficient methods for computing such reachable sets [11]–[17].

The distinguishing examples described above, and many others considered in robotics and autonomous systems, are hybrid. Hybrid systems are characterized by continuous systems with a mode-based operation, where the different modes correspond to different continuous dynamics (typically described by ordinary differential equations) governing the system evolution [18], [19]. These differences could be caused by different control regimes, such as in regular flight or collision avoidance modes, physical changes in the system dynamics based on interaction with the environment, such as locomotion or other aerial/ground robotic interactions, or by a simplification of the control design which lumps together like behaviors of a continuous system into modes.

In this paper, we review the methodology that we have developed for computing reachable sets for hybrid systems, and the corresponding methods for computing controllers which guarantee safety and target capture. We present this methodology in the context of three autonomous/semi-autonomous system examples: aerobatic maneuver design for an autonomous quadrotor aerial vehicle, controller design for quadrotor motion planning under uncertainty, and a game of capture-the-flag in which reachability is used as a tool to aid human controllers.

While this work is motivated by the need to verify the behavior of safety critical dynamical systems and design control laws for such systems with verified behavior, it is necessary to state some caveats. Safety or performance verification of dynamic systems may be roughly classified into two groups: exact verification on very simple models, or approximate verification on more complex models. The methods presented in this paper fall into the second group: the computational methods we employ are convergent numerical methods, thus approximate, yet they may be applied to hybrid systems

†These authors contributed equally to this work.

J. Ding, W. Zhang, and C. Tomlin are with the Department of Electrical Engineering and Computer Science, UC Berkeley, Berkeley, CA, 94720-1770 (email: {jding, weizhang, tomlin}@eecs.berkeley.edu)

J. Gillula is with the Computer Science Department, Stanford University, Stanford, CA, 94305-4035 (email: jgillula@cs.stanford.edu)

H. Huang and M. Vitus are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305-4035 (email: {haomiao, vitus}@stanford.edu)

with continuous dynamics represented by nonlinear ordinary differential equations with both control and disturbance inputs. These computational methods are grid-based, and while the computation in each discrete state may be performed in parallel, the computational complexity of these methods scales exponentially in the dimension of the continuous state. On today's standard laptop computers, we are typically limited to continuous state dimensions of five or less. Nonetheless, a fast computation on a coarse grid allows us to rapidly rule out large parts of the state space as safe, and to focus our finer computations on interesting regions closer to the boundary of the unsafe regions. While none of the system verification methods designed today eliminates the need for detailed testing since the models are always approximation of the actual dynamics, we believe that the methods presented here have the potential to reduce testing, and to direct the test engineer to more useful and informative tests.

The rest of this paper is organized as follows. We first present a simplified form of a general hybrid system, which we use for our analysis. We then present an algorithm for computing reachable sets for hybrid systems, and a controller synthesis method for guaranteeing that the system remains in the reachable set. The majority of the paper focuses on the applications of these methods to the examples mentioned above. We conclude with some new directions that we and other groups are taking in these areas.

## II. Hybrid Systems Model

The last decade has witnessed increased research interest in building complex robotic systems by properly combining multiple simple control laws designed for different subtasks or operating scenarios. Such examples include reactive control for motion planning [20], locomotion control of bipedal robots [21], switching control of Unmanned Aerial Vehicles (UAVs) [22], [10], cooperative control of multi-robot systems [23], [24], and coverage control with unicycle robots [25], among others. While breaking down the overall control task into several simpler ones can simplify the design process, the interaction between the physical dynamics and the rules of the discrete switching logic may result in unexpected system behaviors or even catastrophic failures, making the analysis and safety verification of the overall system significantly more challenging. Hence, to guarantee safety and that specific performance requirements are met, these couplings should be properly incorporated into the mathematical representation of the system, necessitating the use of a hybrid system model.

To simplify the discussion, this paper adopts a slight specialization of the hybrid system models presented in [9]. As illustrated in Figure 1, the state of a hybrid system is described by a combination of a continuous state variable $x$, taking values in the n-dimensional Euclidean space $\mathbb{R}^n$, and a discrete state variable $q$, taking values in a finite (or countable) set $Q$ that represents the different operating modes of the system. The inputs to the system consist of a finite collection of discrete control inputs $\Sigma$, a set of continuous control inputs $U$, and a set of continuous disturbance inputs $D$.

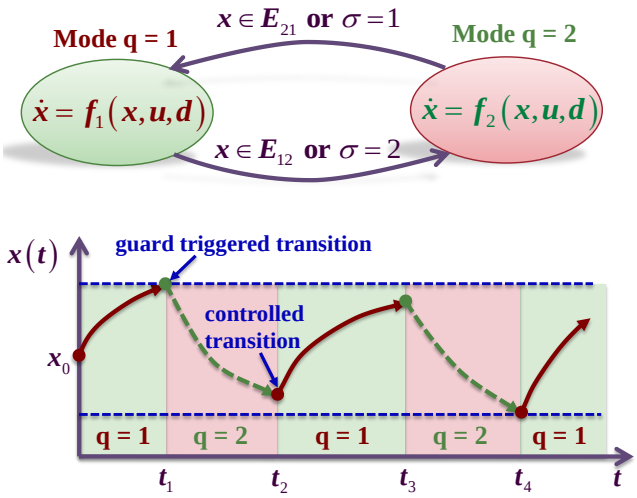


Fig. 1. An example of hybrid system model and its trajectory.

In each mode $i \in Q$, the continuous state evolves according to an ordinary differential equation model

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

where $u$ and $d$ are the continuous control and disturbance inputs, respectively. The evolution of the discrete state is described by a transition function $F : Q \times X \times \Sigma \to Q$, which can be viewed as a slight generalization of the state transition function used in formal descriptions of finite automata and discrete event systems (see for example [26]). As an example, the transition logic for mode 1 in Figure 1 is described by $F(1, x, \sigma) = 2$ when either $x \in E_{12}$ or $\sigma = 2$, and $F(1, x, \sigma) = 1$ otherwise.

Under this model, an execution of the hybrid system proceeds roughly as follows. From an initial state $(i, x_0) \in Q \times X$, the continuous state evolves according to (1), while the discrete state remains constant until the first time $t_1$ when the function $F$ evaluates to a discrete state $j \neq i$. This triggers a discrete jump from $i$ to $j$ while the continuous state remains constant. The continuous state then evolves according to the dynamics in mode $j$, and the whole process repeats. A simple example of a hybrid trajectory is given in Figure 1, where the mode transitions at time instants $t_1$ and $t_4$ are autonomous jumps due to the continuous state $x$ hitting the guard conditions $E_{12}$ and $E_{21}$, while the ones at $t_2$ and $t_3$ are controlled transitions triggered by changes in the discrete command $\sigma$.

The rest of the paper focuses on controller design and synthesis methods for the class of hybrid systems described here. More specifically, the goal is to find control policies for the continuous and discrete inputs in order to drive the hybrid state into a designated region at the end of the control horizon without hitting some known unsafe sets during the process. Such a design goal is difficult to achieve for hybrid systems due to the entanglement between their discrete and continuous dynamics. In the next section, a reachability-based approach will be introduced to tackle this challenging problem.

## III. Reachability

### A. Background

Reachability analysis for hybrid systems has been a prolific area of research over the past two decades. Existing methods in this field can be broadly classified according to the assumptions on the continuous time dynamics (clocks [27], linear [12], [28], nonlinear [9], [14]) and the types of set computation (discrete abstraction [19], [29], polytopes [17], ellipsoids [13], numerical discretization [30], [31]).

The methods for controller design and synthesis in this paper are based upon the Hamilton-Jacobi approach to reachability analysis as described in [9], [31], with the advantages of being able to handle nonlinear system dynamics and bounded time-varying disturbances. The effectiveness of this approach has been demonstrated in applications such as design of aerobatic maneuvers [32], synthesis of robust motion control strategies [10], and planning in adversarial scenarios [33].

### B. Continuous Time Hamilton-Jacobi Reachability

Before going into the specifics of the controller design and synthesis methods, some basic definitions of continuous time reachable sets will be introduced, along with a brief review of how these sets can be computed using the method of Hamilton-Jacobi reachability. In this preliminary discussion, the system dynamics are assumed to be $\dot{x}(t) = f(x(t), u(t), d(t))$, $x(0) = x_0$, evolving in $\mathbb{R}^n$ subject to $u(t) \in U$, $d(t) \in D$.

First, consider a safety verification problem where some unsafe terminal set $A \subset \mathbb{R}^n$ to be avoided is specified, along with a set of permissible initial conditions $X_0$; the task is to prove that $X_0$ does not contain any states from which the system trajectory terminates inside $A$ within some time $\tau$. This involves computing the set of states for which regardless of the input $u$, there exists some choice of disturbance $d$ such that $x(\tau) \in A$. This will be referred to as the **avoid set** over time $\tau$, denoted by $\mathcal{A}(A, \tau)$.

One possible way of computing this set is via optimal control. As a first step, a continuous function $l : \mathbb{R}^n \to \mathbb{R}$ is constructed such that $l(x) \leq 0$ if and only if $x \in A$. ($l$ is commonly referred to as a level set function). Now consider a terminal cost problem where the control seeks to maximize $l(x(\tau))$ while the disturbance tries to minimize the same. The value function at the initial time is then given by $J(x, 0) = \max_{u(\cdot)} \min_{d(\cdot)} l(x(\tau))$, where the maximization and minimization are taken over realizations of the input and disturbance over the interval $[0, \tau]$. Here the input $u$ is allowed to be selected according to a state feedback strategy. Clearly, $\mathcal{A}(A, \tau)$ is the set of states $x$ such that $J(x, 0) \leq 0$ (see Figure 2). Moreover, it has been shown [34] that $J$ is the unique viscosity solution of the Hamilton-Jacobi-Isaacs (HJI) partial differential equation (PDE)

$$\frac{\partial J}{\partial t} + H\left(x, \frac{\partial J}{\partial x}\right) = 0, \; J(x, \tau) = l(x), \tag{2}$$

where the Hamiltonian is defined as

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

Moreover, as discussed in [9], the corresponding optimal control input for avoiding the terminal set $A$ can be synthesized according to

$$u^*(x, t) \in \arg \min_{u \in \mathbb{U}} \max_{d \in \mathbb{D}} p(x, -t)^T f(x, u, d), \; t \in [0, T], \tag{4}$$

where $p = \frac{\partial \phi}{\partial x}$. In cases where one is interested in computing the avoid set with respect to a particular choice of input, for example according to some feedback policy $u(t) = K(x(t))$, then the Hamiltonian reduces to $H(x, p) = \min_{d \in D} p^T f(x, K(x), d)$.

Numerical solutions to equation (2) on a grid can be calculated using the Level Set Toolbox [35]. It should be noted that the computational complexity of the algorithm underlying the Level Set Toolbox depends on both the dimension of the state space as well as the computational cost of performing the static optimization inside the Hamiltonian. In many of the applications we describe in section IV, for example, the inputs $u$ and disturbances $d$ enter in an affine manner and so the optimization in the Hamiltonian can be performed in constant time for each node in the grid on which the equation is being solved. Unfortunately the number of nodes in the grid (and thus the accuracy of the resulting solution) scales exponentially with the number of dimensions in the state space, which in practice restricts this sort of reachability analysis to systems of dimension five or less, primarily due to limitations on the amount of memory available on current computers. New developments may help to alleviate this issue, however, including recent work on a mixed implicit explicit formulation which takes advantage of the fact that for many physical systems several of the state dimensions are simple integrators [36].

It turns out that slight modifications of equation (2) can be used to solve a number of different verification problems. For example, the computational procedure just described can be generalized [31] to the case where the safety specification is to avoid $A$ over the entire time interval $[0, \tau]$. Another example is the computation of a **capture set**, defined as the set of states that can be driven into a target set $R$ over some fixed time horizon. Using a level set representation of $R$, the reachability problem can be again posed as a terminal cost problem where the value function is given by $J(x, 0) = \min_{u(\cdot)} \max_{d(\cdot)} l(x(\tau))$. The corresponding set will be denoted by $\mathcal{R}(R, \tau)$. Finally, under a combination of safety and target attainability objectives, the set of states that can be driven inside $R$ while avoiding $A$ can be computed using a constrained HJI PDE [37]. This set will be referred to as the **reach-avoid set**, denoted by $\mathcal{RA}(R, A, \tau)$. It is worth noting that under a particular choice of feedback control, the reach-avoid set computation simplifies to a set difference between capture set and avoid sets. Specifically, under a particular choice of feedback law for $u$, we have $\mathcal{RA}(R, A, \tau) = \mathcal{R}(R, \tau) \backslash \mathcal{A}(A, \tau)$, which can be computed via a pointwise maximization of value functions derived from the capture set and avoid set calculations.

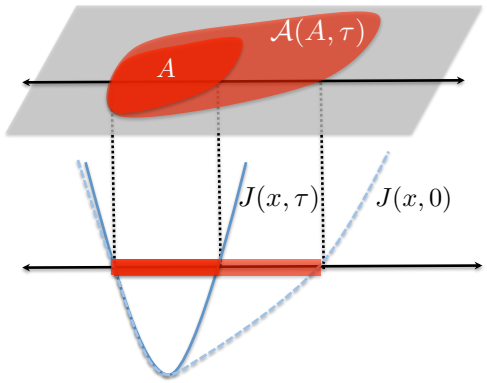


Fig. 2. The zero sub-level set of an appropriate cost function is used to define capture or unsafe regions in the state space. Solving the modified Hamilton-Jacobi PDE backward in time gives the reachable sets.

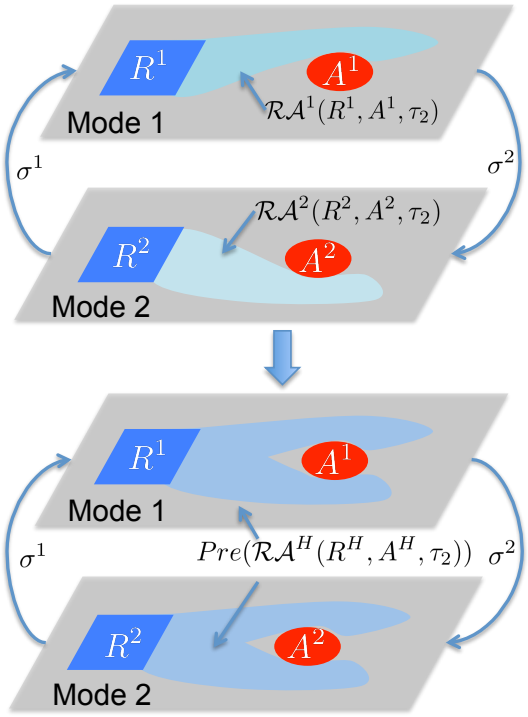


Fig. 3. Reach-avoid sets for hybrid systems are constructed by first determining continuous reachable sets in each mode, and then combining these sets with switching as described in Algorithm 1.

### C. Hybrid System Reachability and Control

Given the large number of design parameters in the model described in section II, one of the first difficulties that can be observed with the control of hybrid systems lies in the problem formulation itself. For example, consider a system consisting of a finite sequence of modes, where the switches between successive modes are completely controlled. In this case, hybrid control involves choosing either the switching times [38] or the switching surfaces [39]. The problem becomes immediately more difficult when the mode sequence is not known ahead of time, in which case the mode sequence needs to be selected in addition to the above design parameters [40]. In the case where there are both controlled and autonomous discrete transitions, the various theoretical and computational issues underlying the control of a hybrid system become even more involved [18].

However, given the description in section II, as well as the example illustrated in Figure 1, one observation that can be made is that every execution of the hybrid system considered in Section II follows a particular structure. Namely, the execution is continuous on a sequence of intervals $[t_0, t_1)$, $(t_1, t_2)$, ..., $(t_{k-1}, t_k]$, and the discrete jumps takes place at the switching times $t_1$, $t_2$, ..., $t_{k-1}$. It turns out that this type of execution also holds for much more general classes of hybrid systems as considered in [41] and [18]. This motivates the following problem formulation: design a controller so that the system trajectory is guaranteed to terminate inside a target set $R^H \subset Q \times \mathbb{R}^n$ while avoiding an unsafe set $A^H \subset Q \times \mathbb{R}^n$, under the class of executions just described, with the timing specifications $t_1 - t_0 \leq \tau_1$, ..., $t_k - t_{k-1} \leq \tau_k$, where $\tau_1$, ..., $\tau_k$ are given. It is important to note that the parameter $\tau_i$ is only an upper bound on the time between the $i$-th and the $i+1$-th mode switch. As such, we allow for executions where the duration of $(t_i, t_{i+1})$ is strictly less than $\tau_i$.

The approach taken in this paper, based upon ideas described in [9], [41], is to construct a controller with verifiable safety and attainability properties from a formal reachability analysis. For the purpose of the application scenarios that will be presented in Section IV, we describe a reachability and controller synthesis algorithm for hybrid systems where the discrete transitions are not forced. Specifically, it is assumed that for each $q \in \mathcal{Q}$ and $x \in X$, there exists a discrete control input $\sigma \in \Sigma$ such that $F(q, x, \sigma) = q$. The implication is that if a transition to another mode is enabled by changes in the continuous state, the controller can still choose to remain in the same mode. When this assumption does not hold, the computational procedures become significantly more involved. The interested reader is referred to [9] for further details.

Now consider the case of one discrete switch ($k = 2$). On the time interval $(t_1, t_2]$, continuous time reachability can be used to compute the set of states that can be driven inside the target set in each mode $i$ while avoiding the unsafe set, regardless of the choice of disturbances. These sets are denoted by $\mathcal{RA}^i(R^i, A^i, \tau_2)$, where $R^i$, $A^i$ are the components of the target and unsafe sets in mode $i$. They are shown in Figure 3 for a simple two mode system. We denote the collection of continuous time reach-avoid set computed on $(t_1, t_2]$ by $\mathcal{RA}^H(R^H, A^H, \tau_2) = \bigcup_{i \in Q} \{i\} \times \mathcal{RA}^i(R^i, A^i, \tau_2)$.

To account for the discrete switch at time $t_1$, it is necessary to introduce the pre-image operator $Pre(S)$, taking as input a subset $S$ of the hybrid state space, and producing as output the set of hybrid states that can switch into $S$ under the transition function $F$ in Section II, either due to changes in the continuous state or under some choice of discrete command $\sigma$. The hybrid reach-avoid set on $[t_1, t_2]$ is then given by $Pre(\mathcal{RA}^H(R^H, A^H, \tau_2))$. For example, under the controlled switches shown in Figure 3, the pre-image of the set $\mathcal{RA}^H(R^H, A^H, \tau_2)$ is the union of the sets $\mathcal{RA}^1(R^1, A^1, \tau_2)$ and $\mathcal{RA}^2(R^2, A^2, \tau_2)$ for both mode 1 and 2. More generally, the set of feasible initial conditions (or a subset thereof) for the

hybrid control problem can be computed using Algorithm 1.

**Algorithm 1** Computation of Hybrid Reach-Avoid Set

**Require:** $R^H, A^H \subset Q \times \mathbb{R}^n$, $\tau_1, ..., \tau_k$
1: $S_0 \Leftarrow R^H$
2: **for** $j = 0$ to $k-2$ **do**
3: Choose $S_{j+1} \subseteq Pre(\mathcal{RA}^H(S_j, A^H, \tau_{k-j}))$
4: **end for**
5: $S_k \Leftarrow \mathcal{RA}^H(S_{k-1}, A^H, \tau_1)$
6: **return** $S_k$

Several remarks are in order. First, at each step of the algorithm, the reach-avoid set update $S_{j+1}$ can be chosen either as equal to the preimage $Pre(\mathcal{RA}^H(S_j, A^H, \tau_{k-j}))$ or as a strict subset, in cases where one may be only interested in computing a subset of the complete reach-avoid set. Second, the computation of the $Pre$ operator in general involves the inversion of the transition relation $F$, which can be a nontrivial task. In Section IV, it will be illustrated through examples how this computation can be performed for discrete transitions that do not depend on the continuous state, as well as in certain cases where there is dependence on the continuous state.

For applications where the reachability computation in Algorithm 1 can be carried out in a tractable manner, a controller satisfying the desired objectives can be designed as follows. At time $t_0$, the system is initialized inside $S_k$ and the continuous control is chosen either as the fixed feedback law used to compute the set $\mathcal{RA}^H(S_{k-1}, A^H, \tau_1)$ or as the optimal control synthesized according to equation (4). This ensures that the system state is driven inside $S_{k-1}$ within $\tau_1$ time units, while avoiding the set $A^H$. Once this occurs, there exists, by the definition of the set $S_{k-1}$, a choice of discrete control to switch the system state into the set $\mathcal{RA}^H(S_{k-2}, A^H, \tau_2)$. Specifically, for a given $(q, x) \in S_{k-1}$, the discrete control can be chosen from the set

$$\left\{\sigma \in \Sigma : (F(q, x, \sigma), x) \in \mathcal{RA}^H(S_{k-2}, A^H, \tau_2)\right\}.$$

Once the system state enters $\mathcal{RA}^H(S_{k-2}, A^H, \tau_2)$, a continuous control can be chosen so as to drive the system state into $S_{k-2}$, and the process repeats until $S_0 = R^H$ is reached.

It can be observed that the performance of the controller obtained from this design procedure will be dependent on the accuracy of the continuous time reachability calculations. As discussed in [35], the accuracy of the numerical solutions obtained from the Level Set Toolbox is directly related to the size of the discrete grid on which the computation is carried out. For applications requiring stringent performance guarantees, conservative approximations of the reach-avoid sets can be obtained when bounds on the numerical errors are available.

To close this section, we remark that in digital control applications where the controls can be only exerted at sampling instants, a possible approach is to discretize the continuous input range of $u$, and reduce the problem to a selection of discrete input levels and switching controls at sampling instants [10]. This approach is briefly discussed in Section IV-B.

## IV. Applications

In this section, we will describe several application examples where the theoretical techniques outlined in the preceding sections has been used to design and verify control schemes for complex systems. These examples have been chosen to illustrate the power and flexibility of reachability analysis in system verification and controller design and synthesis. Due to space constraints, each example is covered briefly; the reader is encouraged to review the cited papers for more detail.

### A. Aerobatic Maneuver Design and Execution

The first example shows reachability-based techniques applied to the design of guaranteed safe aerobatic maneuvers [22]. In this work, reachability analysis was used to design and implement a backflip maneuver for a quadrotor helicopter (Figure 4), part of the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC). Reachable sets were used to guarantee that the quadrotor would be able to safely complete the backflip even under worst-case disturbances.

*1) Reachable Sets for Attainability and Safety:* Since the quadrotor's propellors cannot generate negative thrust, the motors must be turned off during inverted flight. As a result, the backflip was divided into three modes as shown in Figure 5: 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. For ease of analysis and visualization, the quadrotor's continuous dynamics were decoupled into the rotational state space, which was analyzed to ensure attainability, and the vertical state space, which was analyzed to ensure safety. This application can thus be viewed as a special case of the hybrid

Fig. 4. STARMAC quadrotor performing an autonomous backflip maneuver.

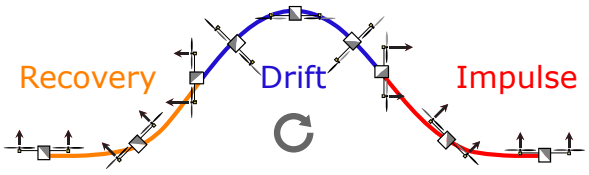


Fig. 5. The backflip maneuver, broken into three modes. The vehicle travels from right to left, spinning clockwise. The size of each arrow indicates the relative thrust from each rotor.

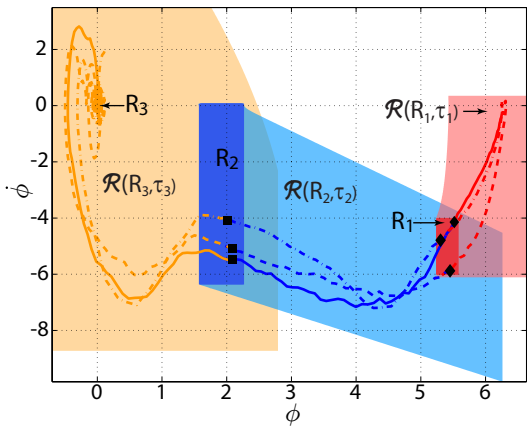


Fig. 6. Three experimental validations (solid, dash and dash-dot lines) of the backflip maneuver overlaid on the composite reach sets plotted in the rotational state space. The maneuver begins in $\mathcal{R}(R_1, \tau_1)$ and ends in $R_3$. The transitions from the impulse to drift mode are shown as black diamonds, and the transitions from the drift to the recovery mode are indicated by the black squares.

control problem described in Section III-C, with a sequence of three modes and two discrete switches occuring between the impulse and drift, and drift and recovery modes.

For this particular example, Algorithm 1 is initialized using the final target set $R_3$. The continuous time reach-avoid set calculation is decoupled into a capture set computation in the rotational state space and an avoid set computation in the vertical state space, under a particular choice of closed-loop controller for each mode. Details of the avoid set computations are omitted here, but the reader is encouraged to refer to [22] for more information. The preimage of a set in mode $i$ of the mode sequence is simply the same set in both mode $i$ and $i-1$, resulting in a simple computation for the $Pre$ operator. The resulting capture sets are shown in Figure 6. (Calculating these sets using the Level Set Toolbox on a standard laptop computer, e.g. 2GHz dual-core CPU, 4GB of RAM, took at most a few minutes of computation time.) In the figures, $\mathcal{R}(R_3, \tau_3)$ is the set in the recovery mode, $\mathcal{R}(R_2, \tau_2)$ is the set in the drift mode, and $\mathcal{R}(R_1, \tau_1)$ is the capture set in the impulse mode. Using the procedures described in Section III-C, a provably safe switching controller can be designed for the back-flip maneuver.

*2) Results:* Figure 6 shows the $(\phi, \dot{\phi})$ trajectory of three experimental validations through the designed capture sets for the backflip maneuver. As the figure illustrates, the trajectories are contained within the capture sets, and the switches between the maneuvers are contained within each of their goal regions. Video of the backflip maneuver being successfully performed by the STARMAC vehicle can be viewed at http://hybrid.eecs.berkeley.edu/aerobatics.html.

### B. Synthesis of Robust Motion Control Policies

Another example is the application of reachability tools to the synthesis of motion control policies that are robust to bounded disturbances [10]. In this case, we consider a problem where the task specification is to generate a state feedback policy that steers the robot into a set of desired target configurations $R \subset \mathbb{R}^n$ using a finite set of high level maneuvers $Q$, while avoiding a set of unsafe configurations $A \subset \mathbb{R}^n$, subject to worst-case bounds on run-time uncertainties (e.g. model inaccuracies, actuator noise, environment disturbances). For practical implementation, the feedback policy selects maneuvers based upon sampled measurements of the system state, obtained at $T$ time units apart.

This can be viewed as another special case of the hybrid control problem described in section III-C, where the target and unsafe sets are the product of the sets $R$ and $A$ in each mode $i \in Q$, while the timing specifications are given by $\tau_0 = \tau_1 = \ldots = \tau_{k-1} = T$. Under the assumption that a switch can be taken from mode $i$ to any other mode $j \in Q$, the computation of $Pre(S)$ in each mode is simplified to $\bigcup_{i \in Q} S^i$, where $S^i$ is the component of $S$ in mode $i$. Thus, the reach-avoid set over a time interval $[0, kT]$ is identical for each mode of the system and can be obtained using Algorithm 2.

**Algorithm 2** Computation of Exact Finite Horizon Reach-Avoid Set

**Require:** $R, A \subset \mathbb{R}^n$, $T > 0$

1: $S_0 \Leftarrow R$
2: **for** $j = 0$ to $k-1$ **do**
3: $\quad S_{j+1} \Leftarrow \bigcup_{i \in Q} \mathcal{RA}^i(S_j, A, T) \cup S_j$
4: **end for**
5: **return** $S_k$

It should be noted that the various uncertainties are accounted for in the computation of the set $\mathcal{RA}^i(S_j, A, T)$, which is carried out according to the system dynamics (1), subject to bounds on the disturbance parameter.

As described in section III-C, the result of this reachability calculation also gives an explicit representation of the feedback policy for choosing the switching controls. Specifically, given a state measurement $x(kT)$, the minimum number of time steps needed to reach $R$ while avoiding $A$ can be computed by iterating through the reach-avoid sets and finding the smallest set $S_j$ containing $x(kT)$. By definition of $S_j$, there exists some maneuver $i$ such that $x(kT) \in \mathcal{RA}^i(S_{j-1}, A, T)$. Choosing $i$ then guarantees that the state trajectory will enter $S_{j-1}$ in one sampling interval. In repeating this procedure, the system state is steered through successively smaller reach-avoid sets, until the target set is attained.

For conceptual illustration, this approach is applied to the problem of controlling STARMAC to some neighborhood of the origin in the 2D plane, while satisfying hard velocity bounds, and subject to model uncertainties and motor noise. Under a previously designed inner control loop, the position-velocity dynamics of the STARMAC in the $x$ and $y$ directions can be reasonably modeled as decoupled double integrators. In this applications, the modes of the system are used to represent discrete choices of roll and pitch angles, which affects the vehicle acceleration in the $x$ and $y$ directions. Figure 7 shows a plot of the reach-avoid sets computed using Algorithm 2 in the position-velocity space. The corresponding feedback policy

over a 2.5 second time horizon is implemented onboard the quadrotor and an experimental trajectory is shown in Figure 8. It can be seen that the vehicle indeed achieves the desired objectives in both the $x$ and $y$ directions within the time horizon of interest, despite disturbance effects.

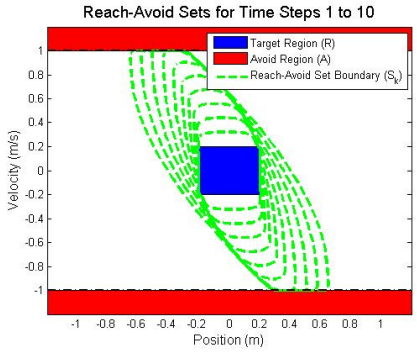


Fig. 7. Reach-avoid sets in position-velocity space over 1 second interval ($T = 0.1$ s).

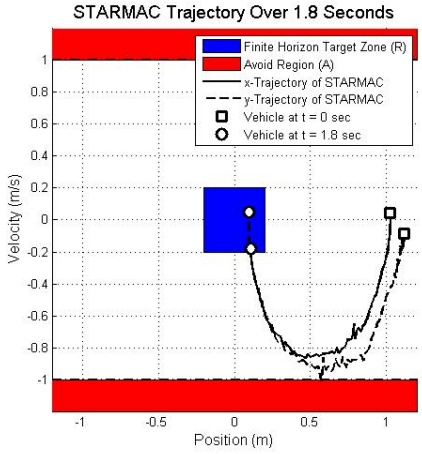


Fig. 8. Position-velocity trajectory of STARMAC over 1.8 seconds.

### C. Reachability Control for Multi-Stage Games

Finally, reachability-based control design can also be used for decision-making and control in multi-stage games. The following example shows how reachability tools can be applied to the game of capture-the-flag, a challenging adversarial scenario where reachability analysis can be used to guide human agents [33].

The simplest version of capture-the-flag involves a single attacker first attempting to reach a flag region and then subsequently returning to a return region, as seen in Figure 9. The attacker wins by completing these objectives in sequence, and the defender wins by preventing this, either by directly capturing the attacker (coming within some radius of the attacker), or simply blocking the attacker through the threat of capture.

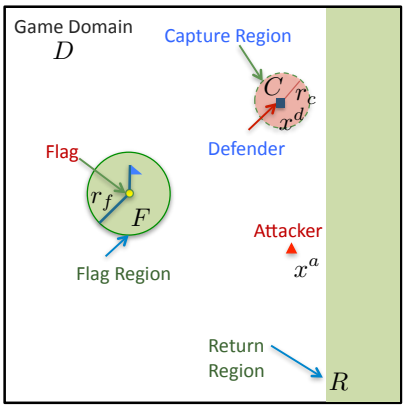


Fig. 9. The basic configuration of the capture-the-flag game.

This game can be modeled as a two mode hybrid system, where the modes encode different stages of the game, and the continuous states encode the joint configuration of the two players. A switch from the first stage to the second stage is enabled by the attacker position entering the flag zone.

The hybrid reachability procedure given in Section III-C can be performed using the return region in the second stage as the final target, and the capture zone as the unsafe region for the attacker in both stages. For the discrete switch, the pre-image of the reach-avoid set in the second stage is computed by taking its intersection with configurations where the attacker is inside the flag region. The result of the reachability analysis produces the set of winning initial configurations for the attacker. Moreover, since the game is zero-sum, the complement of this set are the configurations where victory is assured for the defender. Under an optimal control framework, the control inputs ensuring victory for either player can be directly synthesized from the value functions produced by the Hamilton-Jacobi calculation, giving a complete solution to the problem.

The inputs and reach-avoid sets can be used for automated agents, but they are also natural tools for assisting a human playing the game. The optimal control inputs can be used to guide the player, and the reach-avoid sets are intuitive visual tools for displaying game information. Although the actual reach-avoid sets are calculated in 4 dimensions, a 2D visualization can be created by fixing the player's own location and then showing a slice of the 4D reach-avoid set. Figure 10 shows the attacking player's perspective, where the attacker position is fixed and the sets show all the locations the defender can start from to prevent the attacker from reaching its objective. In this case the regions show defender victory conditions for preventing the attacker from reaching just the flag ($F_D$), just the return zone ($R_D$), and playing the full game ($W_D$).

Figure 10 also shows simulation results showing trajectories for the players as they follow the computed optimal inputs. Figure 10b shows the attacker being blocked and captured, and Figure 10d shows the attacker successfully reaching the flag and then returning.

By posing planning for capture-the-flag as a reachability problem, control inputs for each player can be found that

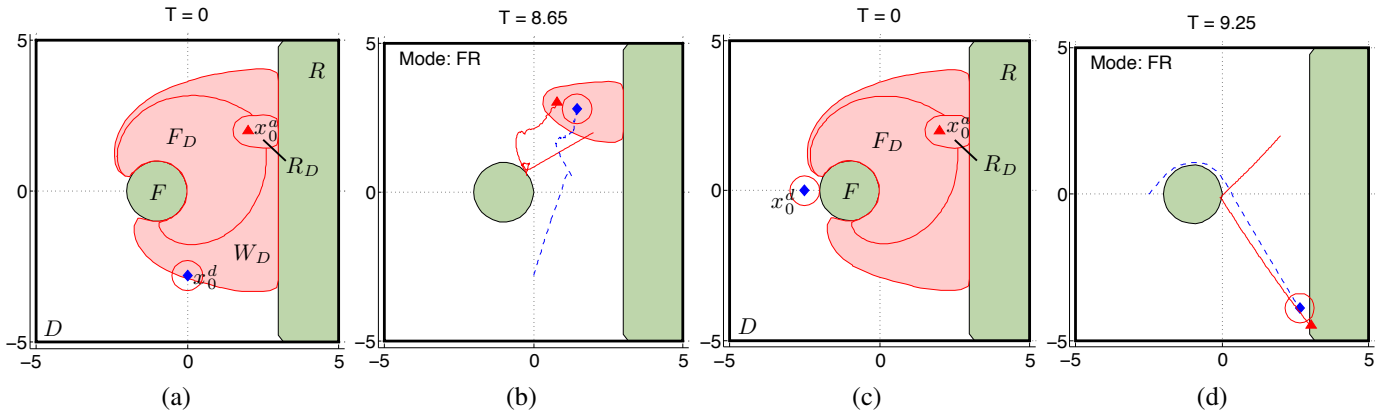


Fig. 10. Two scenarios showing combined winning regions $W_D$ for both game modes. In (a) the defender starts inside $W_D$ and successfully prevents the attacker from returning with the flag, as seen in (b). (c) shows a case where the defender is outside of $W_D$, and is unable to prevent the attacker's successful flag capture and return, as shown in (d).

guarantee victory if victory is possible. Currently, the utility of the reachability analysis is constrained by computational limits as the state-space expands exponentially with the number of continuous states. Efforts are under way to find approximations that would allow for larger games to be analyzed.

## V. Conclusions & New Vistas

The applications discussed here and in other work illustrate the potential of reachability-based control design, a potential that is only beginning to be realized. The techniques described here give a control designer a range of tools for verifying and designing control systems with safety properties that are guaranteed to within the limits of the models used. The tools and applications discussed above show how reachability analysis can be used to verify mode switching for complex automated maneuvers, design mode sequences for planning scenarios, and be used for control and analysis in complex adversarial scenarios. Recent progress in tools for efficient reachability analysis have greatly expanded the potential scope of hybrid reachability's use in control and verification, with many possibilities for the future.

One area where reachability analysis can make a substantial impact is in the integration of reach-avoid sets with environment sensing and obstacle avoidance. By treating maneuvers and their corresponding reach-avoid sets as control primitives, techniques such as the one described in Section IV-B can be used to quickly generate safe, feasible trajectories for robotic vehicles. A project is underway to apply these methods to a quadrotor UAV being operated in a complex, obstacle-rich environment by a remote human operator. In this project the human is generally responsible for the operation of the UAV, while automated controllers are used to fly the UAV safely through tight spaces and difficult maneuvers.

This also highlights another important advantage of reachability analysis: reachable sets are attractive and intuitive tools for assisting human operators. Tools are also being developed for complex situations with teams of multiple human and robotic agents. A significant part of this research will involve field experiments using a smartphone-based capture-the-flag game system currently being developed.

Finally, another project that is being pursued examines the use of reachability-based tools for robust machine learning. While many commonly used machine learning algorithms demonstrate excellent performance in robotic tasks, the convergence guarantees associated with these algorithms are typically asymptotic in nature; few have guarantees about their robustness under more realistic assumptions of limited sample sizes. This is particularly problematic when it is necessary to run these algorithms online, for example in a reinforcement learning scenario where a robot is simultaneously learning a model and must make decisions about how to act given that model. In such a scenario a spurious sample could cause the robot's model to temporarily be wildly incorrect, causing it to take an unsafe or even catastrophic action.

This situation may be avoided by combining machine learning techniques with reachability-style tools that make use of physics-based models and reasonable assumptions about worst-case errors. By doing so, it is hoped that the resulting toolset will provide the same level of excellent performance common to machine learning algorithms, while simultaneously providing the kinds of guarantees about safety that are the essence of the reachability-based techniques described in this paper.

The ability to generate safety and attainability guarantees within model error for control of complex robotic systems is a powerful tool. Such guarantees could be used in practice to quickly rule out large parts of the system state space as safe, and to focus detailed simulation and testing on scenarios which operate close to the boundary of the safe set, leading to shorter design and testing times. Such methods could increase the number of scenarios in which robotic and other automation technology can be used, particularly in safety-critical areas where unexpected or unpredicted robotic behavior might result in human injury. By using such formal methods, designers can have increased confidence that the robotic systems they create will always function in a safe, reliable manner.

## VI. Acknowledgments

The authors wish to thank Professor Ian Mitchell for his insight into the computational complexity of level set methods.

## REFERENCES


[1] J. P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, "A motion planner for nonholonomic mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 5, pp. 577 –593, Oct. 1994.

[2] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566 –580, Aug. 1996.

[3] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501 –518, Oct. 1992.

[4] T. Lozano-Perez, "Spatial planning: A configuration space approach," *Computers, IEEE Transactions on*, vol. C-32, no. 2, pp. 108 –120, February 1983.

[5] M. Spong and M. Vidyasagar, "Robust linear compensator design for nonlinear robotic control," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 4, pp. 345 –351, August 1987.

[6] R. M. Murray, L. Zexiang, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.

[7] A. Coates, P. Abbeel, and A. Y. Ng, "Apprenticeship learning for helicopter control," *Commun. ACM*, vol. 52, no. 7, pp. 97–105, 2009.

[8] E. Asarin, O. Maler, and A. Pnueli, "Reachability analysis of dynamical systems having piecewise-constant derivatives," *Theoretical Computer Science*, vol. 138, no. 1, pp. 35 – 65, 1995.

[9] C. Tomlin, J. Lygeros, and S. Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, July 2000.

[10] J. Ding, E. Li, H. Huang, and C. J. Tomlin, "Reachability-based synthesis of feedback policies for motion planning under bounded disturbances," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[11] S. Prajna, A. Jadbabaie, and G. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *Automatic Control, IEEE Transactions on*, vol. 52, no. 8, pp. 1415 –1428, August 2007.

[12] A. Bemporad, F. D. Torrisi, and M. Morari, "Optimization-based verification and stability characterization of piecewise affine and hybrid systems," in *Hybrid Systems: Computation and Control*, 2000, pp. 45–58.

[13] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis," in *Hybrid Systems: Computation and Control*, 2000, pp. 202–214.

[14] E. Asarin, T. Dang, and A. Girard, "Reachability analysis of nonlinear systems using conservative approximation," in *Hybrid Systems: Computation and Control*, 2003, pp. 20–35.

[15] A. Girard, C. L. Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *Hybrid Systems: Computation and Control*, 2006, pp. 257–271.

[16] M. Kvasnica, P. Grieder, and M. Baotić, "Multi-Parametric Toolbox (MPT)," 2004. [Online]. Available: http://control.ee.ethz.ch/~mpt/

[17] A. Chutinan and B. Krogh, "Computational techniques for hybrid system verification," *Automatic Control, IEEE Transactions on*, vol. 48, no. 1, pp. 64 – 75, January 2003.

[18] M. Branicky, V. Borkar, and S. Mitter, "A unified framework for hybrid control: model and optimal control theory," *Automatic Control, IEEE Transactions on*, vol. 43, no. 1, pp. 31–45, January 1998.

[19] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas, "Discrete abstractions of hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 971 –984, July 2000.

[20] J. M. Esposito, "Simulation and control of hybrid systems with applications to mobile robotics," Ph.D. dissertation, University of Pennsylvania, 2002.

[21] E. R. Westervelt, J. W. Grizzle, C. Chevallerea, J. H. Choi, and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Taylor & Francis/CRC, 2007.

[22] J. Gillula, G. M. Hoffmann, H. Huang, M. P. Vitus, and C. J. Tomlin, "Applications of hybrid reachability analysis to robotic aerial vehicles," *International Journal of Robotics Research*, vol. 30, no. 3, March 2011.

[23] B. Shucker, T. Murphey, and J. K. Bennett, "Switching rules for decentralized control with simple control laws," in *Proc. 2007 American Control Conf.*, New York, NY, July 2007, pp. 1485–1492.

[24] R. Alur, J. Esposito, M. Kim, V. Kumar, and I. Lee, "Formal modeling and analysis of hybrid systems: A case study in multi-robot coordination," in *Proceedings of the World Congress on Formal Methods*, ser. LNCS 1708. Springer, 1999, pp. 212–232.

[25] A. Kwok and S. Martínez, "Unicycle coverage control via hybrid modeling," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 528–532, 2010.

[26] P. Ramadge and W. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81 –98, January 1989.

[27] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, pp. 183 – 235, 1994.

[28] E. Asarin, B. Olivier, T. Dang, and O. Maler, "Approximate reachability analysis of piecewise-linear dynamical systems," in *Lecture Notes in Computer Science, Hybrid Systems: Computation and Control*, vol. 1790. Berlin, Germany: Springer-Verlag, 2000, pp. 20–31.

[29] A. Girard, G. Pola, and P. Tabuada, "Approximately bisimilar symbolic models for incrementally stable switched systems," *Automatic Control, IEEE Transactions on*, vol. 55, no. 1, pp. 116 –126, January 2010.

[30] P. Saint-Pierre, "Hybrid kernels and capture basins for impulse constrained systems," in *Hybrid Systems: Computation and Control*, 2002, pp. 378–392.

[31] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.

[32] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "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, May 2010.

[33] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, accepted.

[34] L. C. Evans and P. E. Souganidis, "Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations," *Indiana Univ. Math. J.*, vol. 33, no. 5, pp. 773–797, 1984.

[35] I. M. Mitchell, "The flexible, extensible and efficient toolbox of level set methods," *Journal of Scientific Computing*, vol. 35, no. 2-3, June 2008.

[36] ——, "Scalable calculation of reach sets and tubes for nonlinear systems with terminal integrators: a mixed implicit explicit formulation," in *Proceedings of the 14th international conference on Hybrid systems: computation and control*. New York, NY, USA: ACM, 2011, pp. 103–112.

[37] I. Mitchell, "Application of level set methods to control and reachability problems in continuous and hybrid systems," Ph.D. dissertation, Stanford University, 2002.

[38] X. Xu and P. Antsaklis, "Optimal control of switched systems based on parameterization of the switching instants," *Automatic Control, IEEE Transactions on*, vol. 49, no. 1, pp. 2 – 16, January 2004.

[39] M. Boccadoro, Y. Wardi, M. Egerstedt, and E. Verriest, "Optimal control of switching surfaces in hybrid dynamical systems," *Discrete Event Dynamic Systems*, vol. 15, pp. 433–448, 2005.

[40] H. Axelsson, Y. Wardi, M. Egerstedt, and E. Verriest, "Gradient descent approach to optimal mode scheduling in hybrid dynamical systems," *Journal of Optimization Theory and Applications*, vol. 136, pp. 167–186, 2008.

[41] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349–370, 1999.