ANNUAL REVIEWS

Annual Review of Control, Robotics, and Autonomous Systems

Hamilton–Jacobi Reachability: Some Recent Theoretical Advances and Applications in Unmanned Airspace Management

Mo Chen and Claire J. Tomlin

Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California 94720, USA; email: mochen72@berkeley.edu, tomlin@berkeley.edu

Annu. Rev. Control Robot. Auton. Syst. 2018. 1:333–58

The *Annual Review of Control, Robotics, and Autonomous Systems* is online at control.annualreviews.org

https://doi.org/10.1146/annurev-control-060117-104941

Keywords

safety-critical systems, Hamilton–Jacobi reachability, computational challenges, unmanned airspace, system decomposition

Abstract

Autonomous systems are becoming pervasive in everyday life, and many of these systems are complex and safety-critical. Formal verification is important for providing performance and safety guarantees for these systems. In particular, Hamilton–Jacobi (HJ) reachability is a formal verification tool for nonlinear and hybrid systems; however, it is computationally intractable for analyzing complex systems, and computational burden is in general a difficult challenge in formal verification. In this review, we begin by briefly presenting background on reachability analysis with an emphasis on the HJ formulation. We then present recent work showing how high-dimensional reachability verification can be made more tractable by focusing on two areas of development: system decomposition for general nonlinear systems, and traffic protocols for unmanned airspace management. By tackling the curse of dimensionality, tractable verification of practical systems is becoming a reality, paving the way for more pervasive and safer automation.

1. INTRODUCTION

Autonomous systems have become increasingly pervasive in everyday life. These systems include unmanned aerial systems, self-driving cars, and many other types of robots. By now, it goes without saying that they have many potential applications, limited only by our imagination. In recent years, a tremendous amount of progress has been made in autonomous systems research in areas such as modeling, planning, sensing, and perception. In addition, the availability of computing power and hardware platforms has helped bridge the gap between theory and practical implementation.

Despite the recent successes in automation, the use of robots and interactions with robots remain quite limited. For example, one of the current uses of unmanned aerial vehicles (UAVs) is surveying areas with few people and no other air traffic. In general, robotic operations are restricted to controlled environments and involve either a single robot or a few robots. These robots also have limited interactions with other robotic agents and humans. There are likely many reasons for this, one of which is simple: If we put many robots close to each other and to humans, we would not know for sure whether they would harm each other or those humans.

Safety is crucial for enabling more effective use of autonomous systems, many of which are safety-critical systems—that is, systems in which failure is extremely costly or even fatal. Formal safety analysis will allow autonomous systems to become provably robust to changes in the environment and to other agents, as well as enable them to operate in much denser configurations. This would mean, for example, that thousands of UAVs could fly in an urban area. Safety analysis is also essential for allowing autonomous systems to interact closely and physically with humans.

1.1. Safety-Critical Autonomous Systems

On an intuitive level, maintaining safety could mean simply avoiding an obstacle, such as a tree. Sometimes the obstacle may be an agent that can also control the way it moves, like an aircraft. On a broader level, maintaining safety means keeping within a set of safe operating conditions. Staying away from obstacles is one specific example, but this concept is quite general. For example, safe operating conditions can be defined in terms of not only position but also any other variables of interest, such as velocity and angle, or even voltage, chemical concentration, human comfort, and degree of trust in automation.

Verification of systems is challenging for many reasons. First, all possible system behaviors need to be accounted for, which makes most simulation-based approaches insufficient and raises the need for formal verification methods. In addition, many practical systems operate in complex environments and are affected by disturbances such as weather conditions; the environments can be unpredictable and may even contain adversarial agents. The systems also evolve in continuous time with complex, nonlinear dynamics.

Perhaps the most difficult challenge of all is that these systems often have high-dimensional configuration spaces. High dimensionality means that many variables are needed to describe the state of a system; this could occur if the system of interest is highly complex and/or if there are many agents in the system.

1.2. Hamilton–Jacobi Reachability as a Safety Analysis Tool

The focus of this review is Hamilton–Jacobi (HJ) reachability analysis, one of the most powerful formal verification tools for guaranteeing the performance and safety properties of systems. The idea is quite simple. Imagine someone riding a bicycle toward a tree (see **Figure 1*a***). Obviously, the rider does not want to run into the tree. To avoid doing so, the rider must change the bicycle's

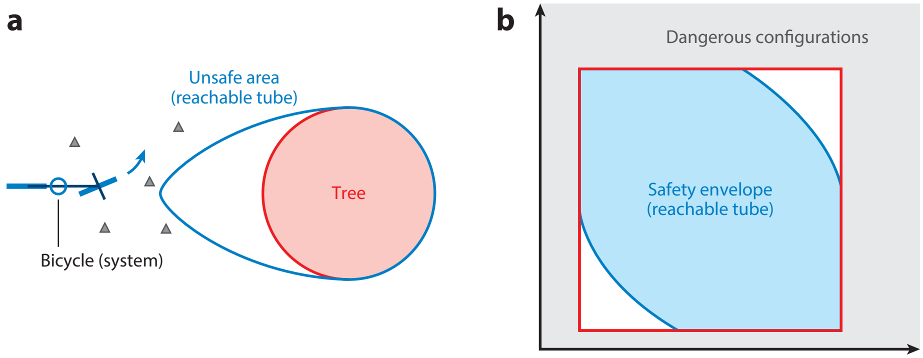

Figure 1

Two examples of reachability analysis. (*a*) A person riding a bicycle toward a tree (*interior of the red circle*). Reachability analysis quantifies when the rider needs to steer the bicycle away to avoid the tree, and the reachable tube represents the unsafe area that the rider must stay out of. (*b*) A more general example, in which a system needs to take action to avoid dangerous configurations (*area outside of the red square*). Here, the reachable tube represents the safe area—a region of sufficient distance (by some suitable metric) from the unsafe conditions.

direction of travel early enough while taking into account variables such as the bicycle's momentum and steering capabilities and any disturbances that might affect steering, such as rough terrain.

Reachability analysis quantifies exactly what it means to steer away early enough. This is done by computing the backward reachable tube (BRT) or (in some cases) a backward reachable set (BRS)—in this example, the region that the rider must stay out of in order to avoid the obstacle. In a more generalized setting, where we would like to keep a system within safe operating conditions, reachability analysis tells us the distance (with respect to a suitable metric) from the unsafe conditions the system needs to maintain (see **Figure 1*b***).

Besides the HJ formulation, there are many other methods related to reachability analysis. In general, none of the current methods, including the HJ formulation, simultaneously address all of the challenges that need to be overcome. For example, dReach (1) and C2E2 (2) excel in determining whether system trajectories from a small set of initial conditions could potentially enter a set of unsafe states, but they do not provide the BRS or BRT—the set of all initial states from which entering some target set is inevitable. Owing to the challenges of computing BRSs and BRTs, the state-of-the-art methods need to make trade-offs on different axes of considerations, such as computational scalability, generality of system dynamics, existence of control and/or disturbance variables, and flexibility in the representation of sets.

For example, the methods presented in References 3–7 have had success in analyzing relatively high-dimensional affine systems using sets of prespecified shapes, such as polytopes or hyperplanes. Other, potentially less scalable methods are able to handle systems with more complex dynamics (4, 8–11). Computational scalability varies among these different methods, with the most scalable methods requiring that the system dynamics do not involve control and disturbance variables. The work in Reference 12 accounts for both control and disturbances but is only applicable to linear systems. Methods that can account for general nonlinear systems (e.g., 13) also sometimes represent sets using simple shapes, such as polytopes, potentially sacrificing representation fidelity in favor of the other aspects mentioned earlier. HJ formulations (14–17) excel in handling general nonlinear dynamics, control and disturbance variables, and flexible set representations via a grid-based approach; however, these methods are the least computationally scalable. Still other methods make a variety of other assumptions in order to make desirable trade-offs (18–20). In addition,

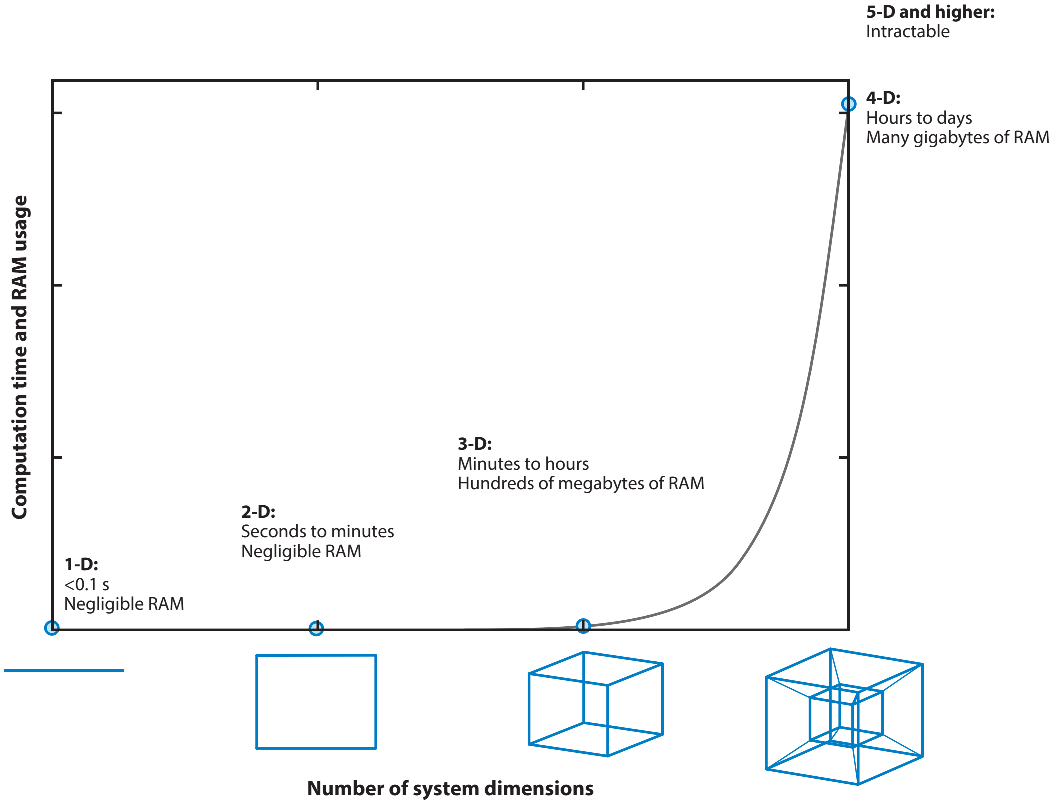

Figure 2

The computational complexity of Hamilton–Jacobi reachability.

under some special scenarios, it may be possible to obtain small computational benefits while minimizing trade-offs in other axes of consideration by exploiting system structure (21–26).

This review focuses on recent developments of the HJ formulation of reachability. It is applicable to general controlled nonlinear systems that involve disturbances or adversarial behaviors and, despite the presence of these factors, computes the exact reachable set rather than approximations. The trade-off is that HJ reachability is the most computationally expensive method. As with every other formal verification method, the computational burden makes HJ reachability intractable for high-dimensional systems. This review presents recently developed methods to alleviate this difficulty, which is referred to as the curse of dimensionality.

In the case of HJ reachability, the computational complexity is exponential with respect to the number of system dimensions (see **Figure 2**). With HJ reachability, 1-D and 2-D reachable sets can be computed very quickly and do not use much RAM, 3-D reachable sets can take minutes to hours to compute and require hundreds of megabytes of RAM, and 4-D reachable sets typically take many hours to days to compute and require many gigabytes of RAM. Owing to computation time and memory limitations, reachable sets of 5 or more dimensions have been considered intractable to compute via the HJ formulation; the work presented in this review has begun to address this challenge.

Unfortunately, high-dimensional systems are the systems for which performance and safety guarantees are most urgently needed, given the recent developments in automation and systems modeling. In this review, we present progress toward tractable formal verification of complex, high-dimensional systems via reachability analysis. The solutions presented involve two broad, complementary approaches:

1. Structural solutions: The behavior of multiagent systems can be nonintuitive and difficult to monitor. In these cases, imposing various structural assumptions (such as having air highways) on the system can significantly reduce problem complexity while allowing intuitive human participation.
2. System decomposition: For general high-dimensional systems, this review presents recently developed techniques to decompose a full dynamical system into multiple subsystems, reducing the computation cost by many orders of magnitude and enabling previously intractable analyses.

Before diving into the specifics of the recent work, we first summarize some research done on HJ reachability in the last decade to provide the background on which the more recent works build.

2. BACKGROUND

HJ reachability analysis falls under the umbrella of optimal control problems and differential games, which are important and powerful theoretical tools for analyzing a wide variety of systems, particularly in safety-critical scenarios. They have been extensively studied in the past several decades (14, 15, 22, 27–30) and have been successfully applied to practical problems such as pairwise collision avoidance (15), aircraft in-flight refueling (31), vehicle platooning (32), and many others (33, 34). With the recent growing interest in using safety-critical autonomous systems such as autonomous cars and UAVs for civil purposes (35–39), the need for flexible tools that can provide safety guarantees has substantially increased.

Intuitively, in an optimal control problem, one seeks to find the cheapest way a system described by an ordinary differential equation model can perform a certain task. In a differential game, a system is controlled by two adversarial agents competing to respectively minimize and maximize a joint cost function. HJ reachability is a common and effective way to analyze both optimal control problems and differential games because of the guarantees that it provides and its flexibility with respect to the system dynamics.

In a reachability problem, one is given some system dynamics described by an ordinary differential equation and a target set that describes the set of final conditions under consideration. Depending on the application, the target set can represent a set of either desired or undesired states. The goal in reachability analysis is to compute various definitions of the BRS, BRT, forward reachable set, or forward reachable tube. This review focuses mainly on backward reachability. When the target set is a set of desired states, the BRS or BRT represents the set of states from which the system can be guaranteed to be driven to the target set despite the worst-case disturbance. By contrast, when the target set is a set of undesired states, the BRS or BRT represents the set of states from which the system may be driven into the target set under some disturbance despite its best control efforts to remain outside. Because of the theoretical guarantees that reachability analysis provides, it is ideal for analyzing the newest problems involving autonomous systems. We define several frequently used formal definitions of BRSs and BRTs in Section 2.2.1.

In addition, HJ reachability is a powerful tool because BRSs and BRTs can be used for synthesizing controllers that steer the system away from a set of unsafe states (safety controllers) in order to guarantee safety as well as controllers that steer the system into a set of goal states (goal satisfaction controllers) to guarantee goal satisfaction. Unlike many formulations of reachability, the HJ formulations are flexible in terms of system dynamics, enabling the analysis of controlled nonlinear systems under disturbances. Furthermore, HJ reachability analysis is complemented by many numerical tools that are readily available to solve the associated HJ partial differential equation (40–42). However, the computation is typically done on a grid, making the problem complexity scale exponentially with the number of states and therefore with the number of vehicles. Consequently, HJ reachability computations are intractable for large numbers of vehicles.

We now formalize the above notions and describe the HJ formulation more specifically, although much of the content in the following sections is agnostic to the reachability formulation.

2.1. System Dynamics

Let $s \in [-\infty, 0]$ be the time and $z \in \mathbb{R}^n$ be the system state, which evolves according to the ordinary differential equation

$$\frac{\mathrm{d}z(s)}{\mathrm{d}s} = \dot{z}(s) = f(z(s), u(s), d(s)), u(s) \in \mathcal{U}, d(s) \in \mathcal{D}. \qquad 1.$$

In general, the theory we present is applicable when some states are periodic dimensions (such as angles), but for simplicity we will consider $\mathbb{R}^n$. The control and disturbance are respectively denoted by $u(s)$ and $d(s)$, with the control function $u(\cdot)$ and disturbance function $d(\cdot)$ respectively drawn from the following set of measurable functions:

$$\begin{aligned} u(\cdot) \in \mathbb{U}(t) &= \{\phi : [t, 0] \to \mathcal{U} : \phi(\cdot) \text{ is measurable}\}, \\ d(\cdot) \in \mathbb{D}(t) &= \{\phi : [t, 0] \to \mathcal{D} : \phi(\cdot) \text{ is measurable}\}, \end{aligned}$$

where $\mathcal{U} \subset \mathbb{R}^{n_u}$ and $\mathcal{D} \subset \mathbb{R}^{n_d}$ are compact and $t < 0$. The system dynamics, or flow field, $f : \mathbb{R}^n \times \mathcal{U} \times \mathcal{D} \to \mathbb{R}^n$ is assumed to be uniformly continuous, bounded, and Lipschitz continuous in z for fixed u and d.[1] Therefore, given $u(\cdot) \in \mathbb{U}, d(\cdot) \in \mathbb{D}$, there exists a unique trajectory solving Equation 1 (43). We will denote solutions, or trajectories, of Equation 1 starting from state z at time t under control $u(\cdot)$ and disturbance $d(\cdot)$ as $\zeta(s; z, t, u(\cdot), d(\cdot)) : [t, 0] \to \mathbb{R}^n$. ζ satisfies Equation 1 with an initial condition almost everywhere:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}s}\zeta(s; z, t, u(\cdot), d(\cdot)) &= f(\zeta(s; z, t, u(\cdot), d(\cdot)), u(s), d(s)), \\ \zeta(t; z, t, u(\cdot), d(\cdot)) &= z. \end{aligned} \qquad 2.$$

For time-invariant system dynamics, the time variables in trajectories can be shifted by any constant τ:

$$\zeta(s; z, t, u(\cdot), d(\cdot)) = \zeta(s + \tau; z, t + \tau, u(\cdot), d(\cdot)), \forall z \in \mathbb{R}^n. \qquad 3.$$

The interaction between disturbance and control is modeled using a differential game, as described by Mitchell et al. (15). We define a strategy for the disturbance as the mapping $\gamma : \mathcal{U} \to \mathcal{D}$ that determines a disturbance signal that reacts to the control signal based on the state. The mapping γ is drawn from only nonanticipative strategies $\gamma \in \Gamma(t)$, and we write $d(\cdot) = \gamma[u](\cdot)$. Nonanticipative strategies are defined as follows:

$$\begin{aligned} \gamma \in \Gamma(t) := \{\mathcal{K} : \mathbb{U}(t) \to \mathbb{D}(t), u(r) &= \hat{u}(r) \text{ for almost every } r \in [t, s] \\ \Rightarrow \mathcal{K}[u](r) &= \mathcal{K}[\hat{u}](r) \text{ for almost every } r \in [t, s]\}. \end{aligned} \qquad 4.$$

[1] When the context is clear, we omit the notation "(s)" from variables such as z and u referring to function values.

Roughly speaking, this means that the disturbance may react only to current and past control signals. Mitchell et al. (15) provided a detailed discussion of this information pattern.

2.2. Hamilton–Jacobi Reachability Analysis

In HJ reachability, we begin with the system dynamics given by an ordinary differential equation and a target set that represents the set of unsafe states. We then solve the HJ equation to obtain various desired forms of reachable sets or tubes, which could represent states leading to danger. To avoid danger, the system may apply any control until it reaches the boundary of a reachable set. At the boundary, applying the optimal safety controller would guarantee avoidance. We present the most commonly used definitions in this section and more specialized definitions in their respective sections below.

2.2.1. Backward reachable sets and tubes. We consider two different definitions of the BRS and two different definitions of the BRT.

Intuitively, a BRS represents the set of states $z \in \mathbb{R}^n$ from which the system can be driven into some set $\mathcal{T} \subseteq \mathbb{R}^n$ at the end of a time horizon of duration $|t|$. We call $\mathcal{T}$ the target set. First, we define the maximal BRS; in this case, the system seeks to enter $\mathcal{T}$ using some control function. We can think of $\mathcal{T}$ as a set of goal states. The maximal BRS represents the set of states from which the system is guaranteed to reach $\mathcal{T}$. The second definition is for the minimal BRS; in this case, the BRS is the set of states that will lead to $\mathcal{T}$ for all possible controls. Here, we often consider $\mathcal{T}$ to be an unsafe set, such as an obstacle. The minimal BRS represents the set of states that leads to violation of safety requirements. The formal definitions of the two BRSs are given below.[2]

Definition 1. The maximal BRS is defined as

$$\mathcal{R}(t) = \{z : \forall \gamma \in \Gamma(t), \exists u(\cdot) \in \mathbb{U}, \zeta(0; z, t, u(\cdot), \gamma[u](\cdot)) \in \mathcal{T}\}.$$

Definition 2. The minimal BRS is defined as

$$\mathcal{A}(t) = \{z : \exists \gamma \in \Gamma(t), \forall u(\cdot) \in \mathbb{U}, \zeta(0; z, t, u(\cdot), \gamma[u](\cdot)) \in \mathcal{T}\}.$$

Whereas BRSs indicate whether a system can be driven into $\mathcal{T}$ at the end of a time horizon, BRTs indicate whether a system can be driven into $\mathcal{T}$ at any time during the time horizon of duration $|t|$. **Figure 3** demonstrates the difference. BRTs are important notions, especially in safety-critical applications, in which we are interested in determining the minimal BRT—the set of states that could lead to danger at any time within a specified time horizon. The formal definitions of the two BRTs are given below.

Definition 3. The maximal BRT is defined as

$$\bar{\mathcal{R}}(t) = \{z : \forall \gamma \in \Gamma(t), \exists u(\cdot) \in \mathbb{U}, \exists s \in [t, 0], \zeta(s; z, t, u(\cdot), \gamma[u](\cdot)) \in \mathcal{T}\}.$$

Definition 4. The minimal BRT is defined as

$$\bar{\mathcal{A}}(t) = \{z : \exists \gamma \in \Gamma(t), \forall u(\cdot) \in \mathbb{U}, \exists s \in [t, 0], \zeta(s; z, t, u(\cdot), \gamma[u](\cdot)) \in \mathcal{T}\}.$$

[2] In the literature, the argument of $\mathcal{R}$, $\mathcal{A}$, $\bar{\mathcal{R}}$, or $\bar{\mathcal{A}}$ is sometimes some nonnegative number $\tau = -t$; however, for simplicity, we will use the nonpositive number t to refer to the time horizon of the BRS and BRT.

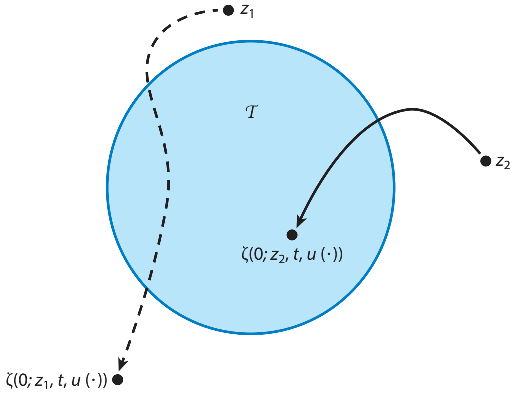

Figure 3

The difference between a backward reachable set (BRS) and a backward reachable tube (BRT). State z_1 is in the BRT but not in the BRS; state z_2 is in both the BRS and the BRT.

The terms maximal and minimal refer to the role of the optimal control (44). In the maximal (or minimal) case, the control causes the BRS or BRT to contain as many (or as few) states as possible, in order to have maximal (or minimal) size.

2.2.2. Computing reachable sets and tubes. HJ formulations such as those described in References 14, 15, 17, and 27 cast the reachability problem as an optimal control or differential game problem and directly compute BRSs and BRTs in the full state space of the system. The numerical methods (e.g., 44) for obtaining the optimal solution all involve solving an HJ partial differential equation on a grid that represents a discretization of the state space. For the time-invariant case, we now summarize necessary details related to the HJ partial differential equations and what their solutions represent in terms of the cost function of the corresponding optimization problem. Reference 22 presents a recent time-varying formulation.

Let the target set $\mathcal{T} \subseteq \mathbb{R}^n$ be represented by the implicit surface function $V_0(z)$ as $\mathcal{T} = \{z : V_0(z) \leq 0\}$. Such a function always exists since we can choose $V_0(\cdot)$ to be the signed distance function from $\mathcal{T}$. Consider the optimization problem

$$V_{\mathcal{R}}(t, z) = \sup_{\gamma[u](\cdot) \in \Gamma(t)} \inf_{u(\cdot) \in \mathbb{U}} V_0(\zeta(0; z, t, u(\cdot), \gamma[u](\cdot))) \quad \text{subject to Equation 2,} \qquad 5.$$

with the optimal control given by

$$u^*_{\mathcal{R}}(\cdot) = \arg \sup_{\gamma[u](\cdot) \in \Gamma(t)} \inf_{u(\cdot) \in \mathbb{U}} V_0(\zeta(0; z, t, u(\cdot), \gamma[u](\cdot))). \qquad 6.$$

The value function $V_{\mathcal{R}}(t, z)$ is the implicit surface function representing the maximal BRS: $\mathcal{R}(t) = \{z : V_{\mathcal{R}}(t, z) \leq 0\}$.

Similarly, consider the optimization problem

$$V_{\mathcal{A}}(t, z) = \inf_{\gamma[u](\cdot) \in \Gamma(t)} \sup_{u(\cdot) \in \mathbb{U}} V_0(\zeta(0; z, t, u(\cdot), \gamma[u](\cdot))) \quad \text{subject to Equation 2,} \qquad 7.$$

with optimal control

$$u^*_{\mathcal{A}}(\cdot) = \arg \inf_{\gamma[u](\cdot) \in \Gamma(t)} \sup_{u(\cdot) \in \mathbb{U}} V_0(\zeta(0; z, t, u(\cdot), \gamma[u](\cdot))). \qquad 8.$$

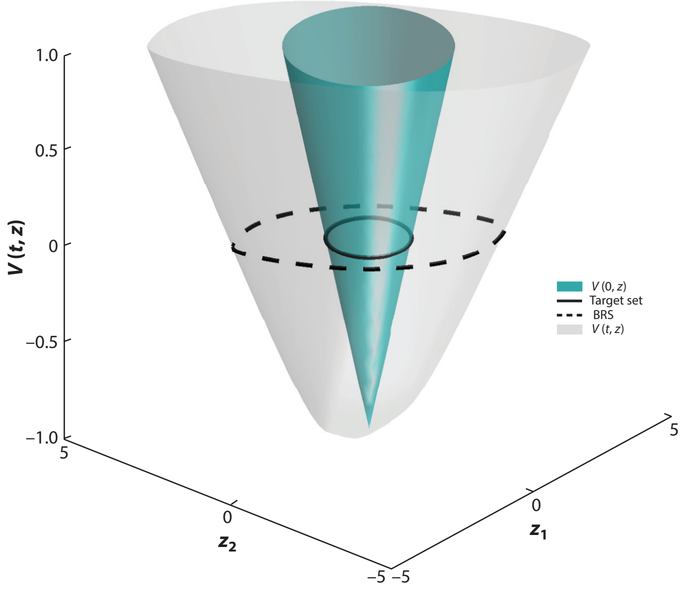

Figure 4

An illustration of Hamilton–Jacobi reachability for a 2-D state space. The zero-sublevel set of $V(0, z)$ represents the target set (boundary in *solid black line*), and the zero-sublevel set of $V(t, z)$ represents the backward reachable set (BRS) (boundary in *dashed black line*). The system of interest can reach the target set within time t if it starts inside the BRS.

Analogously, we also have $\mathcal{A}(t) = \{z : V_{\mathcal{A}}(t, z) \leq 0\}$. **Figure 4** provides a simple 2-D example demonstrating the relationships among the target set, implicit surface function, BRS, and value function.

The value functions $V_{\mathcal{R}}(t, z)$ and $V_{\mathcal{A}}(t, z)$ are the viscosity solution (45, 46) of the HJ partial differential equation

$$\begin{aligned} D_s V(s, z) + H(z, \nabla V(s, z)) &= 0, \quad s \in [t, 0], \\ V(0, z) &= V_0(z). \end{aligned} \qquad 9.$$

The Hamiltonian $H(z, \lambda)$ depends on the system dynamics and on the optimal control problem. For example, for the optimal control problem represented by Equation 5, the Hamiltonian is given by

$$H(z, \lambda) = \min_{u \in \mathcal{U}} \max_{d \in \mathcal{D}} \lambda \cdot f(z, u, d). \qquad 10.$$

Once the value function $V_{\mathcal{R}}$ is computed, the optimal control problem represented by Equation 6 can be obtained by the expression

$$u^*_{\mathcal{R}}(s) = \arg \min_{u \in \mathcal{U}} \max_{d \in \mathcal{D}} \nabla V_{\mathcal{R}}(s, z) \cdot f(z, u, d). \qquad 11.$$

Similarly, for the optimal control problem represented by Equation 7, the Hamiltonian is given by

$$H(s, z, \lambda) = \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \lambda \cdot f(z, u, d), \qquad 12.$$

and the optimal control is given by

$$u^*_{\mathcal{A}}(s) = \arg \max_{u \in \mathcal{U}} \min_{d \in \mathcal{D}} \nabla V_{\mathcal{A}}(s, z) \cdot f(z, u, d). \qquad 13.$$

Furthermore, by the dynamic programming principle, the optimal value on optimal trajectories must be constant:

$$\begin{aligned} V_{\mathcal{R}}(s, \zeta(s; z, \tau, u^*_{\mathcal{R}}(\cdot)) &= V_{\mathcal{R}}(\tau, z)\, \forall \tau, s \in [t, 0], \\ V_{\mathcal{A}}(s, \zeta(s; z, \tau, u^*_{\mathcal{A}}(\cdot)) &= V_{\mathcal{A}}(\tau, z)\, \forall \tau, s \in [t, 0]. \end{aligned} \qquad 14.$$

For the BRT, several equivalent formulations have been proposed. For example, one can modify the value function to keep track of the minimum value of the function $V_0(\cdot)$ that the system trajectory achieves over some time horizon, so that Equations 5 and 7 respectively become

$$\begin{aligned} \bar{V}_{\mathcal{R}}(t, z) &= \sup_{\gamma[u](\cdot)\in\Gamma(t)} \inf_{u(\cdot)\in\mathbb{U}} \min_{s\in[t,0]} V_0(\zeta(0; z, t, u(\cdot), \gamma[u](\cdot))), \\ \bar{V}_{\mathcal{A}}(t, z) &= \inf_{\gamma[u](\cdot)\in\Gamma(t)} \sup_{u(\cdot)\in\mathbb{U}} \min_{s\in[t,0]} V_0(\zeta(0; z, t, u(\cdot), \gamma[u](\cdot))). \end{aligned} \qquad 15.$$

We encourage reading the details of this formulation given in Reference 22 (which contains a very general time-varying reach–avoid framework) or of other formulations, such as those given in References 14–17. However, for this review, it suffices to note that $\bar{V}_{\mathcal{R}}$ and $\bar{V}_{\mathcal{A}}$ are the viscosity solution of the following HJ variational inequality:

$$\begin{aligned} \min\{D_s \bar{V}(s, z) + H(z, \nabla \bar{V}(s, z)), V_0(z) - \bar{V}(s, z)\} &= 0, \quad s \in [t, 0], \\ \bar{V}(0, z) &= V_0(z), \end{aligned} \qquad 16.$$

where $H(z, \lambda)$ is given by Equations 10 and 12 for $\bar{V}_{\mathcal{R}}$ and $\bar{V}_{\mathcal{A}}$, respectively.

3. SYSTEM DECOMPOSITION

There are several drawbacks to using reachability analysis on large systems, whether one is using the HJ or other formulations. In this section, we briefly introduce several methods that alleviate the computational burden in the HJ context by exploiting system structure. In particular, we introduce in detail a new method for decomposing a system into smaller, self-contained subsystems. This method, presented in Section 3.1, is based on the new concept of self-contained subsystems commonly found in vehicle dynamics and mechanical systems. Section 3.2 briefly outlines various other methods that exploit system structure.

3.1. Decomposition via Self-Contained Subsystems

In this section, we present a system decomposition method for computing BRSs and BRTs of a class of nonlinear systems. The method is applicable when self-contained subsystems can be defined as in Equation 17. It drastically reduces dimensionality without making any other trade-offs by first computing BRSs for lower-dimensional subsystems and then reconstructing the full-dimensional BRS. When reconstructing the minimal BRS $\mathcal{A}$ by taking the intersection of lower-dimensional minimal BRSs $\mathcal{A}_i$, and when reconstructing the maximal BRS $\mathcal{R}$ by taking the union of lower-dimensional maximal BRSs $\mathcal{R}_i$, any approximation errors present arise only from the lower-dimensional computations; no additional approximation errors are incurred. Crucially, the subsystems can be coupled through common states, controls, and disturbances. The treatment of this coupling distinguishes this work from others that consider completely decoupled subsystems, potentially obtained through transformations (47, 48).

The theory summarized in this section is compatible with any methods that compute BRSs and BRTs (e.g., 4, 8, 9, 21, 24). In addition, combining different decomposition methods can achieve even more dimensionality reduction. Reference 49 presents a more detailed account of the material presented in this section.

3.1.1. Summary of formulation and definitions. In this section, we seek to obtain the BRSs and BRTs in Definitions 1–4 (given in Section 2.2.1) via computations in lower-dimensional subspaces under the assumption that Equation 1 can be decomposed into self-contained subsystems represented by Equation 17. Such a decomposition is common, since many systems involve components that are loosely coupled. In particular, the evolution of position variables in vehicle dynamics is often weakly coupled through other variables, such as heading.

Let the state $z \in \mathbb{R}^n$ be partitioned as $z = (z_1, z_2, z_c)$, with $z_1 \in \mathbb{R}^{n_1}; z_2 \in \mathbb{R}^{n_2}; z_c \in \mathbb{R}^{n_c}; n_1, n_2 > 0; n_c \geq 0$; and $n_1 + n_2 + n_c = n$. Note that n_c could be zero. These states are grouped into subsystems by defining the self-contained subsystem states $x_1 = (z_1, z_c) \in \mathbb{R}^{n_1+n_c}$ and $x_2 = (z_2, z_c) \in \mathbb{R}^{n_2+n_c}$, where x_1 and x_2 in general share the common states in z_c.

Definition 5 (self-contained subsystem). Consider the following form of system dynamics:

$$\begin{aligned} \dot{z}_1 &= f_1(z_1, z_c, u), \\ \dot{z}_2 &= f_2(z_2, z_c, u), \\ \dot{z}_c &= f_c(z_c, u). \end{aligned} \tag{17.}$$

Each of the subsystems with states defined as $x_i = (z_i, z_c)$ is called a self-contained subsystem:

$$\begin{array}{cc} \dot{z}_1 = f_1(z_1, z_c, u) & \dot{z}_2 = f_2(z_2, z_c, u) \\ \dot{z}_c = f_c(z_c, u) & \dot{z}_c = f_c(z_c, u) \\ \text{(subsystem 1)} & \text{(subsystem 2).} \end{array}$$

Note that the two subsystems are coupled through the common state partition z_c and control u. When the subsystems are coupled through u, the subsystems have shared control. An example of a system that can be decomposed into self-contained subsystems is the Dubins car with constant speed v:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix}, \qquad \omega \in \mathcal{U}, \tag{18.}$$

with state $z = (p_x, p_y, \theta)$ representing the x position, y position, and heading, and control $u = \omega$ representing the turn rate. This system can be decomposed into self-contained subsystems as follows:

$$\begin{aligned} \dot{x}_1 &= \begin{bmatrix} \dot{z}_1 \\ \dot{z}_c \end{bmatrix} = \begin{bmatrix} \dot{p}_x \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ \omega \end{bmatrix}, \\ \dot{x}_2 &= \begin{bmatrix} \dot{z}_2 \\ \dot{z}_c \end{bmatrix} = \begin{bmatrix} \dot{p}_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\sin\theta \\ \omega \end{bmatrix}, \\ u &= \omega, \end{aligned} \tag{19.}$$

where the overlapping state is θ, and the subsystem controls and their shared component is the control u itself.

Projection operations are defined for a state and for a set. The projection of a state $z = (z_1, z_2, z_c)$ onto a subsystem state space $\mathbb{R}^{n_i+n_c}$ is given by

$$\text{proj}_i(z) = x_i = (z_i, z_c). \tag{20.}$$

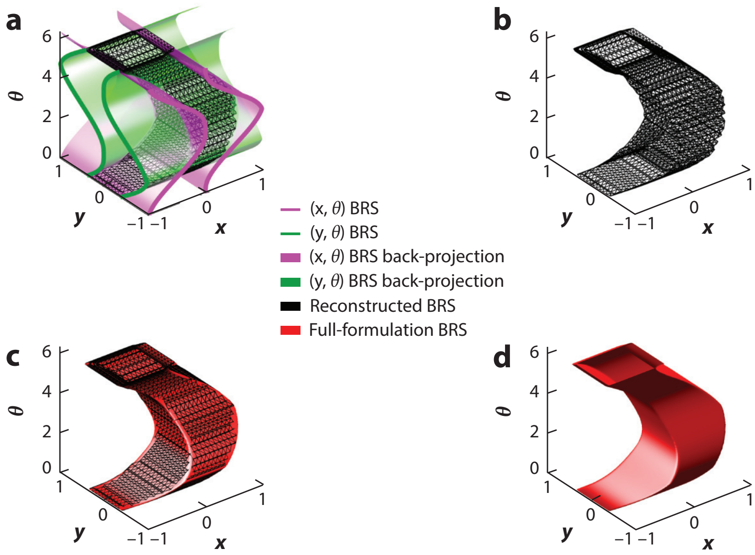

Figure 5

Comparison of the Dubins car backward reachable set (BRS) $\mathcal{A}(t = -0.5)$ computed using the full formulation and via decomposition. (*a*) BRSs in the lower-dimensional subspaces (*green* and *magenta*) and how they are combined to form the full-dimensional BRS (*black*). (*b*) The BRS computed via decomposition shown by itself. (*c*) The BRSs computed using decomposition (*black*) and the full formulation (*red*) superimposed on each other, showing that they are indistinguishable. (*d*) The BRS computed using the full formulation shown by itself. Figure adapted from Reference 49.

The back-projection operator applied to a point or a set is defined as

$$\text{proj}^{-1}(x_i) = \{z \in \mathbb{R}^n : (z_i, z_c) = x_i\}, \qquad 21.$$
$$\text{proj}^{-1}(\mathcal{S}_i) = \{z \in \mathbb{R}^n : \exists x_i \in \mathcal{S}_i, (z_i, z_c) = x_i\},$$

where $\mathcal{S}_i$ is some set in $\mathbb{R}^{n_i+n_c}$.

Chen et al. (49) assumed that the full system target set $\mathcal{T}$ is representable in terms of the subsystem target sets $\mathcal{T}_1 \subseteq \mathbb{R}^{n_1+n_c}$ and $\mathcal{T}_2 \subseteq \mathbb{R}^{n_2+n_c}$ in one of the following ways:

$$\mathcal{T} = \text{proj}^{-1}(\mathcal{T}_1) \cap \text{proj}^{-1}(\mathcal{T}_2), \qquad 22.$$
$$\mathcal{T} = \text{proj}^{-1}(\mathcal{T}_1) \cup \text{proj}^{-1}(\mathcal{T}_2).$$

Although it is more restrictive than a purely grid-based representation in the full-dimensional space, this decomposition of sets can still yield relatively complex shapes (e.g., those shown in **Figures 5** and **8** below).

Next, we define the subsystem BRSs $\mathcal{R}_i$ and $\mathcal{A}_i$ the same way as in Definitions 1 and 2 but with the subsystems and subsystem target sets $\mathcal{T}_i, i = 1, 2$, respectively:

$$\mathcal{R}_i(t) = \{x_i : \exists u(\cdot), \xi_i(0; x_i, t, u(\cdot)) \in \mathcal{T}_i\}, \qquad 23.$$
$$\mathcal{A}_i(t) = \{x_i : \forall u(\cdot), \xi_i(0; x_i, t, u(\cdot)) \in \mathcal{T}_i\}.$$

Given the above definitions, the authors of Reference 49 achieved the following:

- Decomposition of BRSs: Full-dimensional BRSs are efficiently computed by performing computations in lower-dimensional subspaces. Specifically, subsystem BRSs $\mathcal{R}_i(t)$ or $\mathcal{A}_i(t)$ are computed, and then the full system BRS $\mathcal{R}(t)$ or $\mathcal{A}(t)$ is reconstructed by taking the union or intersection of back-projections of subsystem BRSs. This process greatly reduces the computational burden by decomposing the full system into two lower-dimensional subsystems.

- Decomposition of BRTs: BRTs are useful since they provide guarantees over a time horizon as opposed to at a particular time. However, often BRTs cannot be decomposed the same way as BRSs; instead, BRTs can be reconstructed from BRSs.
- Treatment of disturbances: The theoretical framework is modified to incorporate the presence of disturbances. Slightly conservative BRSs and BRTs can still be obtained in this case.

3.1.2. Numerical examples. We now provide two numerical examples to illustrate decomposition into self-contained subsystems—the Dubins car for comparison with computations without decomposition, and the 6-D acrobatic quadrotor to show computation of a BRS that was previously intractable using HJ reachability.

3.1.2.1. The Dubins car. The Dubins car is a well-known system whose dynamics are given by Equation 18. This system is only 3-D, and its BRS can be tractably computed in the full-dimensional space, so it is used to compare the full formulation with our decomposition method. The Dubins car dynamics can be decomposed according to Equation 19. For this example, Chen et al. (49) computed the BRS from the target set representing positions near the origin in both the p_x and p_y dimensions:

$$\mathcal{T} = \{(p_x, p_y, \theta) : |p_x|, |p_y| \leq 0.5\}. \qquad 24.$$

Such a target set $\mathcal{T}$ can be used to model an obstacle that the vehicle must avoid. Given $\mathcal{T}$, the interpretation of the BRS $\mathcal{A}(t)$ is the set of states from which a collision with the obstacle will occur after a duration of $|t|$. From $\mathcal{T}$, the BRS $\mathcal{A}(t)$ at $t = -0.5$ is computed. **Figure 5*c*,*d*** shows the resulting full-formulation BRS. To compute the BRS using the decomposition method, the unsafe set $\mathcal{T}$ is written as

$$\begin{aligned}\mathcal{T}_1 &= \{(p_x, \theta) : |p_x| \leq 0.5\}, \mathcal{T}_2 = \{(p_y, \theta) : |p_y| \leq 0.5\},\\ \mathcal{T} &= \text{proj}^{-1}(\mathcal{T}_1) \cap \text{proj}^{-1}(\mathcal{T}_2).\end{aligned} \qquad 25.$$

From $\mathcal{T}_1$ and $\mathcal{T}_2$, the lower-dimensional BRSs $\mathcal{A}_1(t)$ and $\mathcal{A}_2(t)$ are computed and then used to reconstruct the full-dimensional BRS $\mathcal{A}(t)$: $\mathcal{A}(t) = \text{proj}^{-1}(\mathcal{A}_1(t)) \cap \text{proj}^{-1}(\mathcal{A}_2(t))$. **Figure 5*a*** shows the subsystem BRSs and their back-projections, and **Figure 5*a–c*** shows the reconstructed BRS.

Figure 6 illustrates the computation benefits of using the decomposition method. One can see that the direct computation of the BRS in 3-D becomes very time-consuming as the number of grid points per dimension increases; the computation via decomposition takes hardly any time in comparison. Directly computing the BRS with 251 grid points per dimension in 3-D took approximately 80 minutes, while computing the BRS via decomposition in 2-D took only approximately 30 seconds.

Figure 7 compares the BRT $\bar{\mathcal{A}}(t), t = -0.5$ computed directly from the target set in Equation 24 and using the decomposition technique from the subsystem target sets in Equation 25. For this computation, there is a large disturbance applied to all three components of the system dynamics. Thus, the BRT computed using the decomposition technique becomes an overapproximation of the true BRT, as shown in **Figure 7**.

3.1.2.2. The 6-D acrobatic quadrotor. The real utility of decomposition methods in general is to make previously intractable BRS and BRT computations tractable. As described here, BRTs for

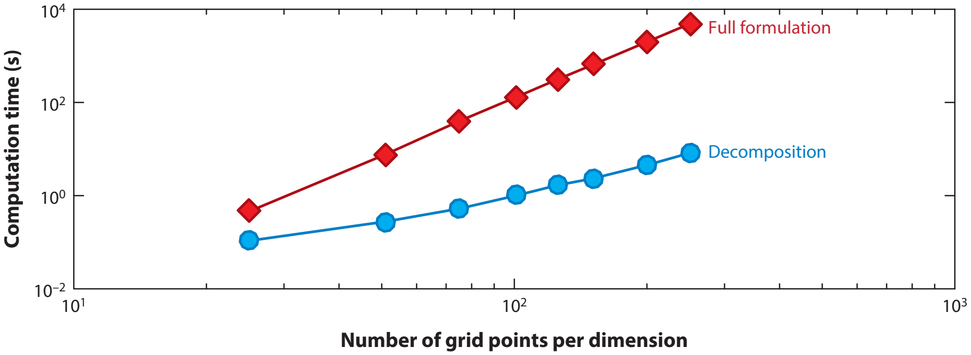

Figure 6

Times for direct computation in 3-D and decomposition in 2-D for the Dubins car, shown on a log scale. The direct computation times in 3-D increase rapidly with the number of grid points per dimension; the computation times in 2-D using decomposition are negligible in comparison. Figure adapted from Reference 49.

the previously intractable 6-D acrobatic quadrotor (50) were computed using the HJ formulation for the first time in Reference 49. The quadrotor has state $z = (p_x, v_x, p_y, v_y, \phi, \omega)$ and dynamics

$$\begin{bmatrix} \dot{p}_x \\ \dot{v}_x \\ \dot{p}_y \\ \dot{v}_y \\ \dot{\phi} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v_x \\ -\frac{1}{m} C_{\mathrm{D}}^{v} v_x - \frac{T_1}{m} \sin\phi - \frac{T_2}{m} \sin\phi \\ v_y \\ -\frac{1}{m}\left(mg + C_{\mathrm{D}}^{v} v_y\right) + \frac{T_1}{m}\cos\phi + \frac{T_2}{m}\cos\phi \\ \omega \\ -\frac{1}{I_{yy}} C_{\mathrm{D}}^{\phi}\omega - \frac{l}{I_{yy}} T_1 + \frac{l}{I_{yy}} T_2 \end{bmatrix}, \qquad 26.$$

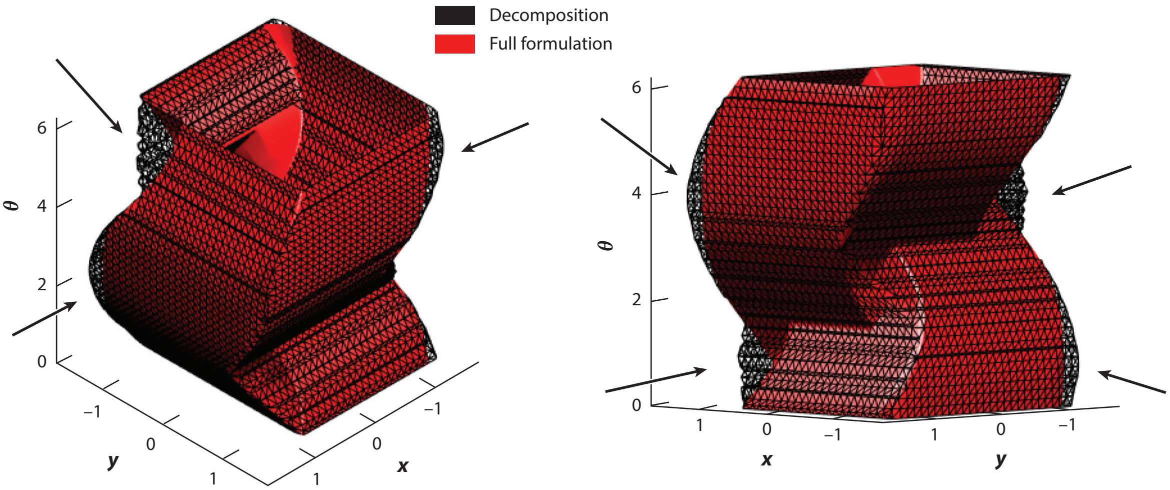

Figure 7

Minimal backward reachable tubes (BRTs) computed directly in 3-D and via decomposition in 2-D for the Dubins car under disturbances with shared components. The reconstructed BRT is an overapproximation of the true BRT, which one can see by noting that the black set is not flush against the red set. The overapproximated regions of the reconstruction are indicated by the arrows. An overapproximation in this case is a conservative approximation; the outside of the BRT represents the set of safe states. Figure adapted from Reference 49.

where x, y, and ϕ represent the quadrotor's horizontal, vertical, and rotational positions, respectively. Their time derivatives represent the velocity with respect to each state. The control inputs T_1 and T_2 represent the thrust exerted on either end of the quadrotor, and the constant system parameters are m for mass, C_{D}^{v} for translational drag, C_{D}^{ϕ} for rotational drag, g for acceleration due to gravity, l for the length from the quadrotor's center to an edge, and I_{yy} for moment of inertia.

The system can be decomposed into two 4-D subsystems:

$$x_1 = (p_x, v_x, \phi, \omega), \qquad x_2 = (p_y, v_y, \phi, \omega). \tag{27.}$$

For this example, Chen et al. (49) computed $\mathcal{A}(t)$ and $\bar{\mathcal{A}}(t)$, which describe the set of initial conditions from which the system may enter the target set despite the best possible control to avoid the target. The target set is defined as a square of length 2 centered at $(p_x, p_y) = (0, 0)$ and described by $\mathcal{T} = \{(p_x, v_x, p_y, v_y, \phi, \omega) : |p_x|, |p_y| \leq 1\}$. This can be interpreted as a positional box centered at the origin that must be avoided for all angles and velocities. From the target set, define $V_0(z)$ such that $V_0(z) \leq 0 \Leftrightarrow z \in \mathcal{T}$. This target set is then decomposed as follows:

$$\begin{aligned} \mathcal{T}_1 &= \{(p_x, v_x, \phi, \omega) : |p_x| \leq 1\}, \\ \mathcal{T}_2 &= \{(p_y, v_y, \phi, \omega) : |p_y| \leq 1\}. \end{aligned}$$

The BRS of each 4-D subsystem is computed and then recombined into the 6-D BRS. To visually depict the 6-D BRS, 3-D slices of the BRS along the positional and pitch axes were computed and are shown in **Figure 8*a***; 3-D slices along the velocity and pitch rate axes are shown in **Figure 8*b***.

3.2. Other Decomposition Techniques

The decomposition technique highlighted above is applicable to general nonlinear systems. In the context of HJ reachability, decomposition techniques for other specific forms of system

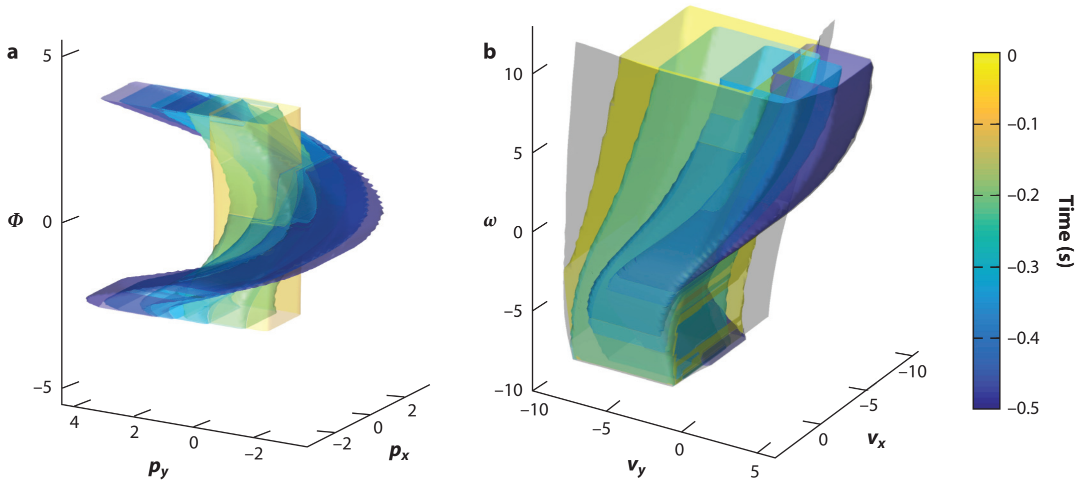

Figure 8

(*a*) 3-D positional slices of reconstructed 6-D backward reachable sets (BRSs) and the backward reachable tube (BRT) in (p_x, p_y, ϕ) space at $v_x = v_y = 1, \omega = 0$, at different points in time. The yellow set represents the target set $\mathcal{T}$; the sets grow darker as time propagates backward. The union of the BRSs is the BRT, shown as the gray surface. (*b*) 3-D velocity slices of the reconstructed 6-D BRSs and BRT in (v_x, v_y, ω) space at $p_x, p_y = 1.5, \phi = 1.5$, at different points in time. Figure adapted from Reference 49.

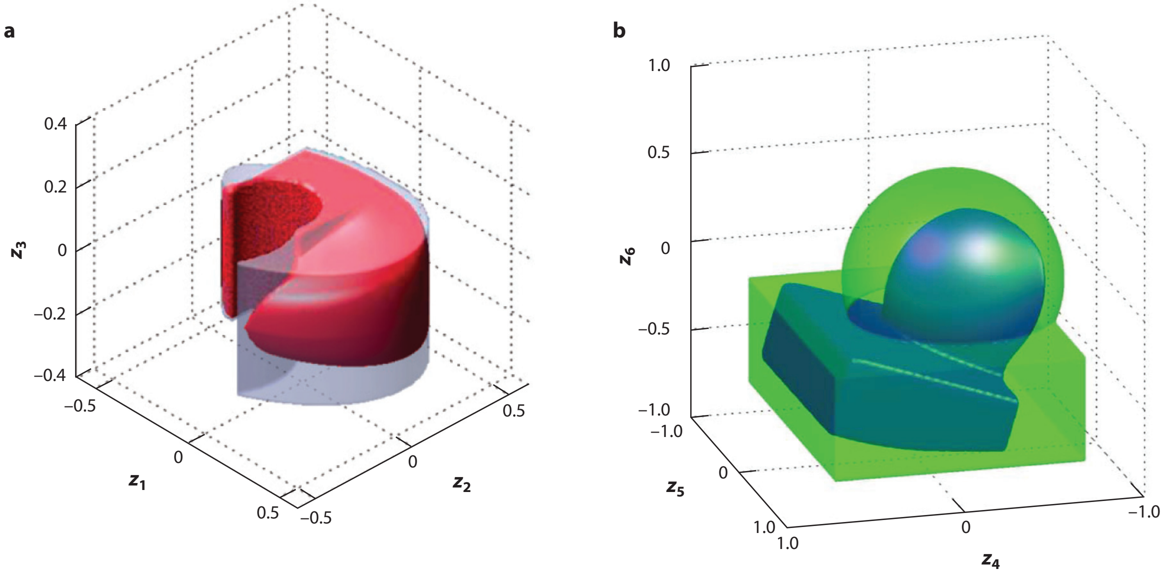

Figure 9

Decomposition results for linear time-invariant systems. (*a*) Overapproximation (*translucent*) of a backward reachable set (BRS) (*solid*). (*b*) Target set (*outside of translucent set*) and approximate BRS (*outside of solid set*). Panel *a* adapted from Reference 51; panel *b* adapted from Reference 26.

dynamics also exist. Kaynama & Oishi (51) proposed a Schur-based decomposition technique for computing reachable sets and synthesizing safety-preserving controllers for linear time-invariant systems. Similar to the work on self-contained subsystems, lower-dimensional reachable sets of subsystems are back-projected and intersected to construct an overapproximation of the reachable set. Kaynama & Oishi (26) used a similar approach for linear time-invariant systems based on a modified Riccati transformation. This method performs decentralized computations in transformed coordinates of subspaces; the result is an approximation of the viability kernel, which is the complement of the minimal reachable set. **Figure 9** shows the conservative approximations obtained from these decomposition techniques.

For systems of general nonlinear dynamics, approximate methods tend to be more conservative in comparison to linear systems. For example, the approximate system decoupling technique described by Chen et al. (24) can be used to simplify system dynamics by treating key state couplings as virtual disturbances. Conservative guarantees on system performance can still be guaranteed; in addition, the idea of disturbance splitting allows a trade-off between the amount of computation and the degree of conservatism. In a similar fashion, the projection-based technique described by Mitchell & Tomlin (23) reduces dimensionality by using virtual disturbances to directly compute projections of reachable sets. **Figure 10** shows computation results from References 23 and 24.

4. UNMANNED AERIAL SYSTEM TRAFFIC MANAGEMENT

UAVs have been used mainly for military operations (52, 53); however, recently there has been an immense surge of interest in their civil applications, and their use is likely to become increasingly prevalent. As a result, government agencies such as the Federal Aviation Administration (FAA) and

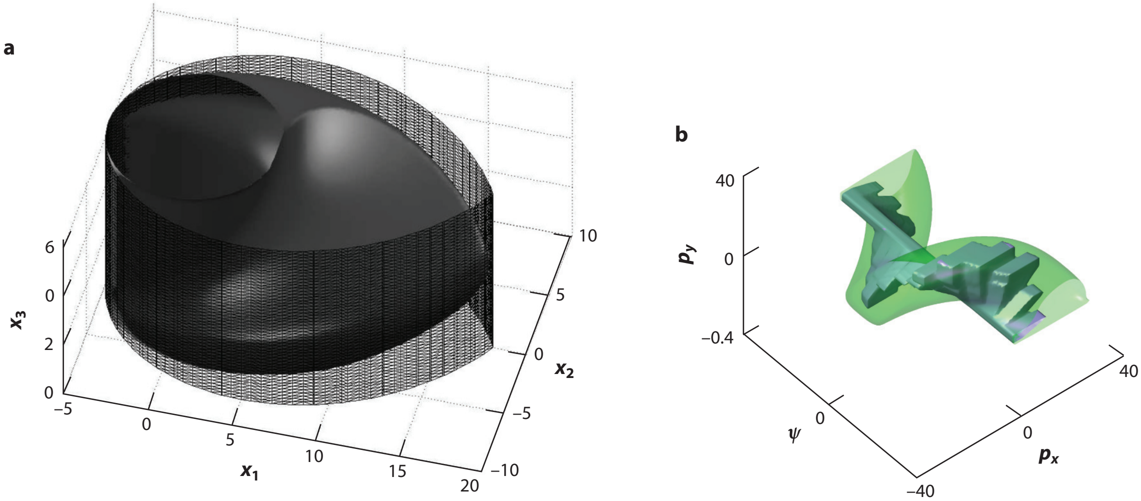

Figure 10

Approximate decomposition results for nonlinear systems. (*a*) Projection-based approximation (*translucent*) of a reachable tube (*solid*). (*b*) Decoupling disturbance-based approximation (*solid gray*) of a reachable set (*translucent green*). Panel *a* adapted from Reference 23; panel *b* adapted from Reference 24.

National Aeronautics and Space Administration (NASA) are also investigating unmanned aerial system traffic management in order to prevent collisions among potentially numerous UAVs (35, 39). In this section, we present recently developed HJ-based approaches for managing the airspace. We first focus on the concept of air highways and unmanned aerial platoons and then briefly summarize several other approaches for addressing the complexity of multiagent systems.

4.1. Air Highways and Unmanned Aerial Platoons

To accommodate potentially thousands of vehicles simultaneously flying in the air, additional structure is needed to allow for tractable analysis and intuitive monitoring by humans. An air highway system on which platoons of vehicles travel accomplishes both goals. In the first part of this section, we propose a flexible and computationally efficient method based on work by Sethian (42) to perform optimal air highway placement given an arbitrary cost map that captures the desirability of having UAVs fly over any geographical location. We demonstrate our method using the San Francisco Bay Area as an example. Once air highways are in place, platoons of UAVs can then fly in fixed formations along the highway to get from their origins to their destinations. The air highway structure greatly simplifies safety analysis while allowing intuitive human participation in unmanned airspace management. Chen et al. (54) provide a more detailed account of the material in this section.

4.1.1. Air highways. Chen et al. (54) proposed the concept of air highways—virtual highways in the airspace in which UAV platoons may be present. UAVs seek to travel from their origins to destinations along a sequence of these air highways. Air highways are intended to be the common pathways for many UAV platoons, whose members may have different origins and destinations. By routing platoons of UAVs onto a few common pathways, the airspace becomes more tractable

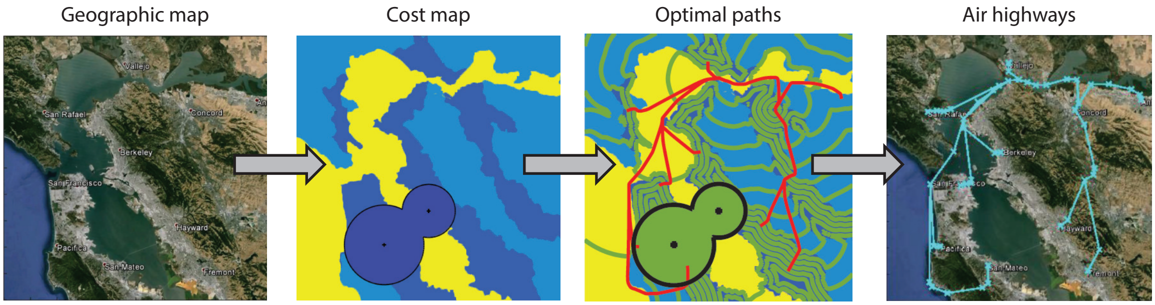

Figure 11

Automatic placement of air highways in the San Francisco Bay Area. In the two center panels, the area enclosed by the black line represents regions around airports, which have the highest cost; the dark blue, yellow, and light blue regions represent cities, water, and other regions, respectively. The origin city was assumed to be Concord, and several other major cities were chosen as destinations. Figure adapted from Reference 54.

to analyze and intuitive to monitor. The authors also proposed the concept of UAV platooning through a hybrid systems approach.

Air highways must account for potential costs, such as people, assets on the ground, and manned aviation—the entities to which UAVs pose the biggest risks (39). Thus, given an origin–destination pair (e.g., two cities), air highways must connect the two points while potentially satisfying other criteria. In addition, ideally it should be possible to recompute optimal air highway locations in real time when necessary in order to update airspace constraints on the fly—in case, for example, airport configurations change or certain airspaces have to be closed (39). With this in mind, Chen et al. (54) defined the air highway placement problem and proposed a fast, simple way to approximate its solution that allows for real-time recomputation. The solution is based on solving the Eikonal equation, which is a specific instance of an HJ equation. The entire air highway placement process can be thought of as converting a cost map over a geographic area in continuous space into a discrete graph whose nodes are waypoints joined by edges, which are the air highways.

Using the San Francisco Bay Area as an example, Chen et al. (54) classified each point on the map into four different regions with descending costs: regions around airports, highly populated cities, water, and other regions. The associated cost of each region reflects the desirability of flying a vehicle over an area in the region. In general, these costs can be arbitrary and determined by government regulation agencies. **Figure 11** shows a San Francisco Bay Area geographic map and cost map along with the optimal (cost-minimizing) paths and resulting air highways.

In general, the cost-minimizing paths to the various destinations overlap and only split up when they are very close to entering their destination cities. This intuitive placement of highways mimics highway networks designed by humans. In addition, since the computation is done on a 2-D domain, the placement of air highways can be done in real time if the cost map changes at a particular time.

4.1.2. Unmanned aerial platoons.

Air highways exhibiting trunk routes that separate near destinations motivate the use of platoons that fly on these highways. Together, the air highway structure and the UAV platooning concept enable the use of reachability to analyze safety and goal satisfaction properties. The structure reduces the likelihood of multiple-way conflicts and

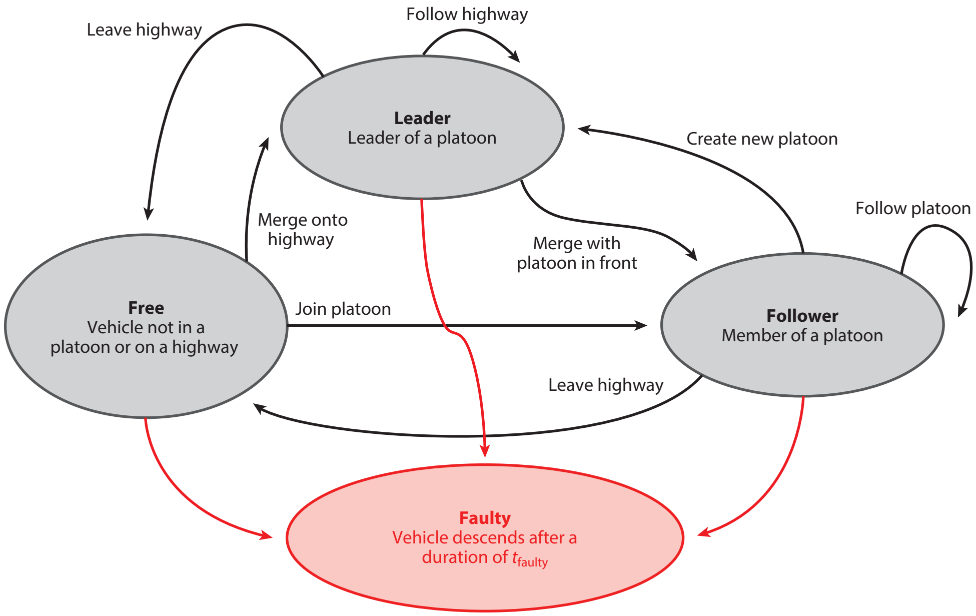

Figure 12

Hybrid modes for vehicles in platoons. Vehicles are in the Free mode before they enter the highway. Figure adapted from Reference 54.

makes pairwise analysis more indicative of the joint safety of all UAVs. In addition to reducing complexity, the proposed structure is intuitive and allows human participation in the monitoring and management of the unmanned airspace.

Organizing UAVs into platoons implies that the UAVs cannot fly in an unstructured way and must have a restricted set of controllers or maneuvers depending on each UAV's role in the airspace. To model UAVs flying in platoons on air highways, we propose a hybrid system whose modes of operations describe a UAV's role in the highway structure. For the hybrid system model, reachability analysis is used to enable successful and safe operation and mode transitions.

In general, the problem of collision avoidance among N vehicles cannot be tractably solved using traditional dynamic programming approaches, such as HJ reachability. Instead, the structure imposed by air highways and platooning enables analysis of the safety and goal satisfaction properties of the vehicles in a tractable manner. From the perspective of each vehicle, the allowable maneuvers become restricted; Chen et al. (54) used a hybrid system model to capture this concept. **Figure 12** summarizes the available maneuvers and associated mode transitions.

Given the above modeling assumptions, Chen et al. (54) provided control strategies to guarantee the success and safety of all mode transitions. The theoretical tool used to provide the safety and goal satisfaction guarantees is HJ reachability. The BRTs computed allow each vehicle to perform complex actions, such as merging onto a highway to form a platoon, joining a new platoon, leaving a platoon to create a new one, or reacting to malfunctioning or intruder vehicles. Several basic controllers are used to perform other, simpler actions, such as following the highway at a constant altitude at a specified speed or maintaining a constant relative position and velocity with respect to the leader of a platoon.

In general, the control strategy of each vehicle has a safety component, which specifies a set of states that it must avoid, and a goal satisfaction component, which specifies a set of states that the vehicle aims to reach. Together, the safety and goal satisfaction controllers guarantee the safety and success of a vehicle in the airspace making any desired mode transition. By combining HJ reachability with a hybrid system structure, the multi-UAV system is able to perform joint maneuvers essential to maintaining structure in the airspace.

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

Using different target sets and dynamics, the reachability-based controllers for vehicle mode transitions are as follows:

- Getting to a target state: The controller used by a vehicle to reach a target state is important in two situations in the platooning context. First, a vehicle in the Free mode can use the controller to merge onto a highway, forming a platoon and changing to the Leader mode. Second, a vehicle in either the Leader mode or the Follower mode can use this controller to change to a different highway, becoming a Leader vehicle.
- Getting to a state relative to another vehicle: In the platooning context, being able to get to a state relative to another moving vehicle is important for forming and joining platoons. For example, a vehicle in the Free mode may join an existing platoon on a highway and change to the Follower mode. Also, vehicles in the Leader or Follower mode may join another platoon and subsequently change to the Follower mode.
- Avoiding collisions: A vehicle can use one of the goal satisfaction controllers described in the two items above when it is not in any danger of colliding with other vehicles. If the vehicle could potentially be involved in a collision within a short period of time, it must switch to a safety controller. The safety controller is available in every mode, and executing the safety controller to perform an avoidance maneuver does not change a vehicle's mode.

In the context of platooning, an unsafe configuration can be defined as follows: A vehicle either is within a minimum separation distance to a reference vehicle in both the x and y directions or is traveling with a speed above the speed limit in either the x or y direction. From this specification, a minimal BRT can be computed to provide a guaranteed-safe controller.

4.1.2.2. Other controllers. HJ reachability is used for relatively complex maneuvers that require safety and goal satisfaction guarantees. For the simpler maneuvers of traveling along a highway and following a platoon, many other well-known methods, such as proportional–integral–derivative (PID) control or model predictive control, would suffice.

4.1.2.3. Numerical simulations. Chen et al. (54) considered several situations that vehicles in a platoon on an air highway may commonly encounter and showed via simulations the behaviors that emerge from the proposed controllers. **Figure 13** shows the results. **Figure 13*a*** illustrates a scenario in which vehicles in the Free mode merge onto an initially unoccupied highway, showing

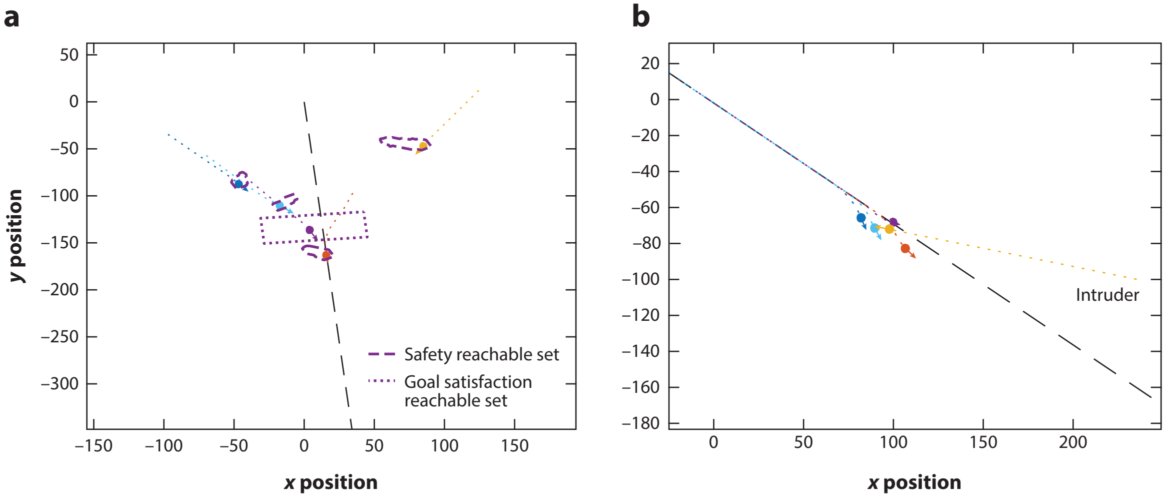

Figure 13

Simulations showing the emerging behavior of the proposed hybrid system framework. (*a*) Vehicles merging onto an air highway, with the goal satisfaction reachable set (*purple dotted line*) and safety reachable sets (*purple dashed lines*) shown for the purple vehicle. (*b*) Safety controllers causing vehicles to respond to an intruder (*yellow*). Figure adapted from Reference 54.

the relevant BRTs for both goal satisfaction and safety. All five vehicles eventually form a single platoon and travel along the highway together.

Figure 13*b* illustrates a platoon's automatic response to an intruder. To avoid collisions, each vehicle checks for safety with respect to the intruder and any vehicles in front of or behind it in the platoon. When the intruder comes in close proximity, the other vehicles spread out to avoid collision. After the danger has passed, the vehicles in the platoon resume normal operation according to the hybrid system model.

4.2. Multiagent Analysis Based on Hamilton–Jacobi Reachability

An air highway network alone is likely not sufficient for UAVs to travel to a final postal address. In References 55 and 56, the authors proposed a sequential trajectory planning scheme. Although HJ reachability is well suited for the robustness requirements needed for the airspace, simultaneous analysis of all vehicles is intractable. Therefore, in Reference 55, the authors assigned a strict priority ordering, with lower-priority vehicles treating higher-priority vehicles as moving obstacles, which allows the reservation of space–time in the airspace for each vehicle. The space–time reservation is dynamically feasible to track when the vehicle experiences disturbances (56) and even in the presence of a single adversarial intruder vehicle under certain assumptions (57). Chen et al. (57) performed a simulation study to demonstrate space–time reservation variants with different assumptions; **Figure 14** shows a simulation involving one adversarial intruder is shown.

An air highway structure and robust routing of UAVs are useful as a first level of safety; additional levels can be provided by last-resort collision avoidance. Mitchell et al. (15) demonstrated guaranteed-safe pairwise collision avoidance, and Chen et al. (58, 59) alleviated the scalability limitations of HJ reachability using a mixed-integer program that exploits the properties of pairwise HJ solutions to provide higher-level control logic. Safety guarantees for three-vehicle collision

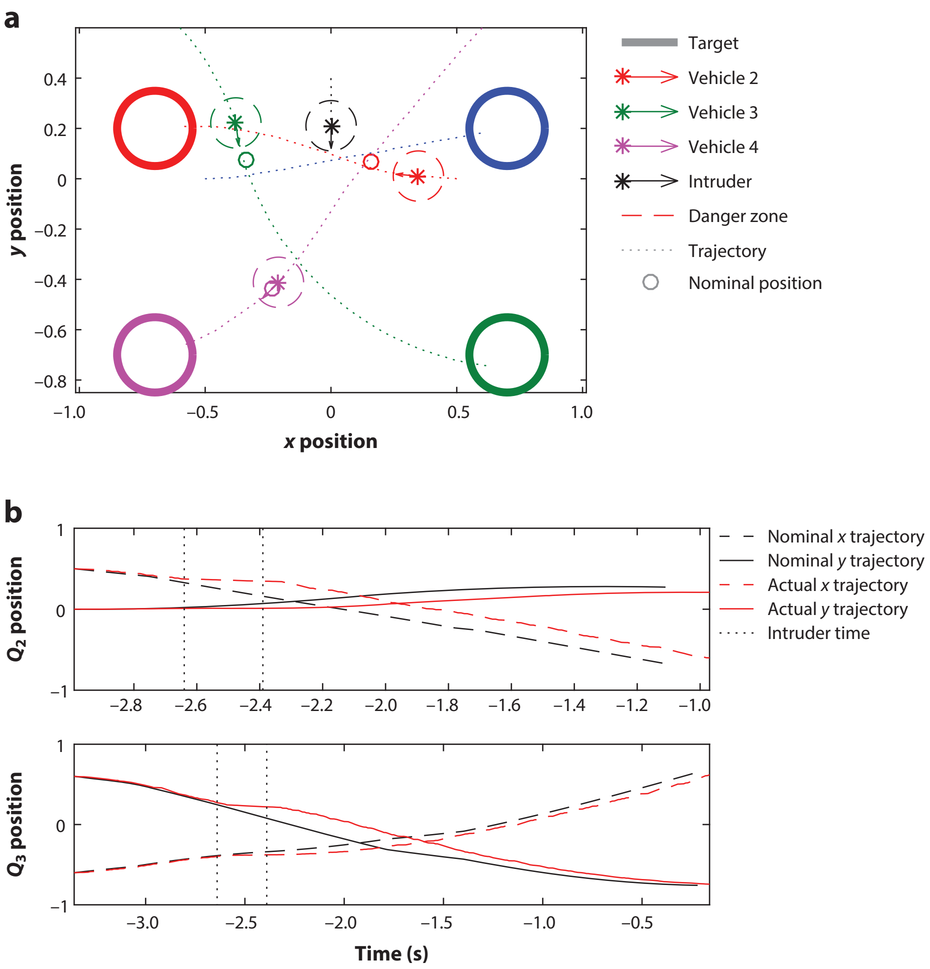

Figure 14

Robust space–time reservations allow multiple unmanned aerial vehicles to arrive at their destinations while avoiding an intruder. (*a*) Two vehicles (*green* and *red*) significantly deviating from their nominal positions to avoid an intruder (*black*). (*b*) Actual and nominal trajectories of the red (Q_2) and green (Q_3) vehicles. Figure adapted from Reference 57.

avoidance—a previously intractable task for HJ reachability—were proved without incurring significant additional computation costs (58). The collision avoidance protocol can also be applied to systems involving more than three vehicles, although no theoretical guarantees can be made. **Figure 15*a*** shows an eight-vehicle collision avoidance simulation.

Rogue UAVs that are unregistered or unauthorized will potentially need to be monitored and captured. Pierson et al. (60) formulated and solved such a multivehicle capture–avoid problem for vehicles of holonomic dynamics; **Figure 15*b*** illustrates their Voronoi cell–based method. To incorporate nonlinear dynamics, a reach–avoid game between a team of attackers and a team of defenders can be solved using HJ reachability combined with maximum matching (59); **Figure 15*c*** shows an example of this man-to-man defense solution.

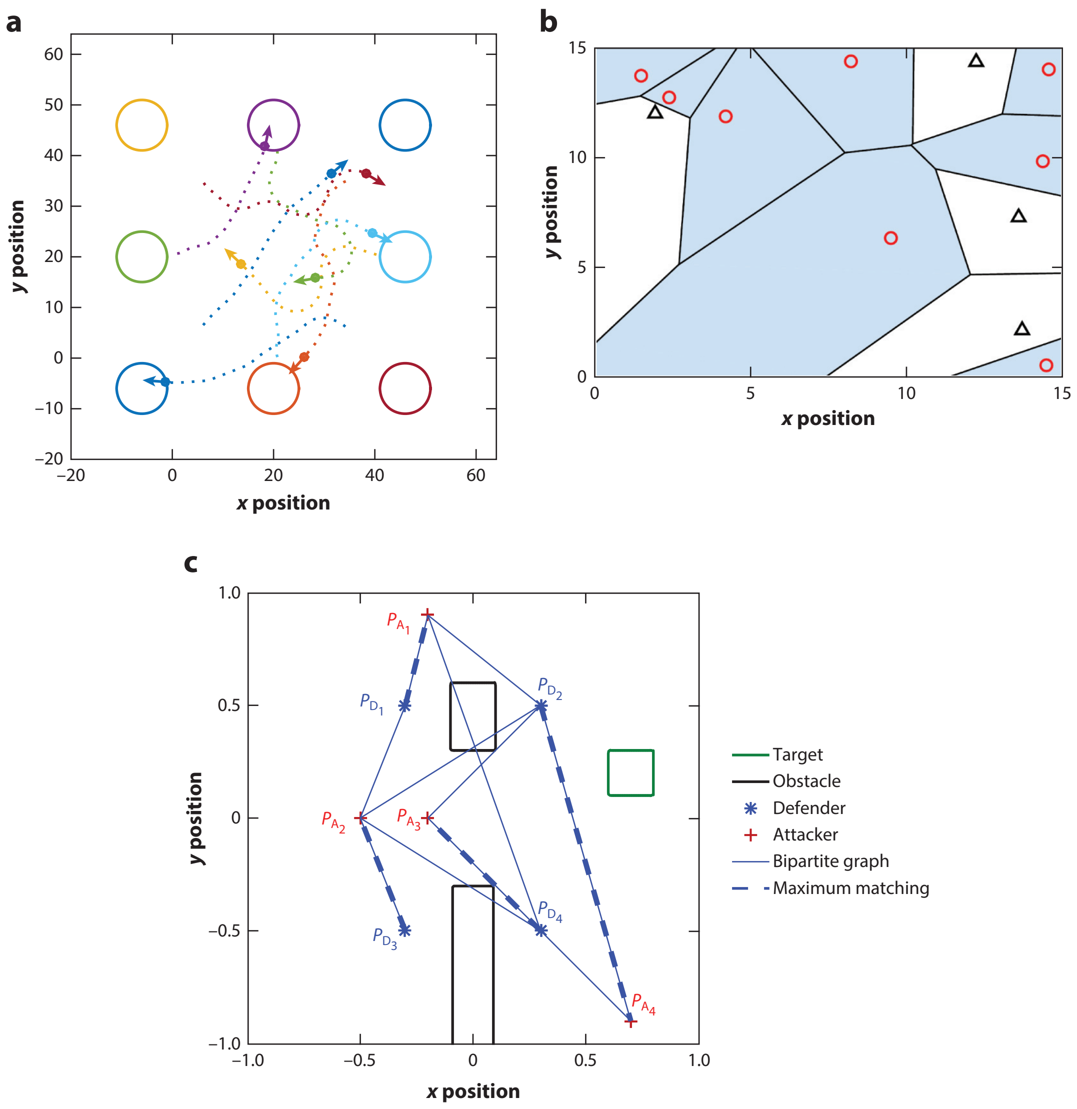

Figure 15

Multivehicle analysis using Hamilton–Jacobi reachability and higher-level logic. (*a*) Simulation of eight-vehicle collision avoidance. A series of pairwise collision avoidance is performed, with the pairs chosen using an integer program. (*b*) Simulation of Voronoi cell–based rogue vehicle capture. Capture is guaranteed through minimization of evaders' safe reachable area. (*c*) The maximum matching process for the interception of a rogue unmanned aerial vehicle. The team of pursuers perform man-to-man defense against a team of evaders. Panel *a* adapted from Reference 58; panel *b* adapted from Reference 60; panel *c* adapted from Reference 59.

5. CONCLUSION

Autonomous systems research has been tremendously successful recently, and safety is now becoming very important, despite the difficulties of safety analysis. With the recent progress in high-dimensional verification via HJ reachability and a combination of low-dimensional verification and higher-level logic, we have made a good start on the path toward more pervasive and verified automation. If large-scale safety analysis could be combined with previous successes in the field in a modular way, we could have safe system design, planning, sensing, and learning; safe large-scale autonomous systems; and safe human–automation interaction in the near future.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

This review was supported by the National Science Foundation under the CPS Frontiers VehiCal project (1545126) and CPS ActionWebs (CNS-931843), by the University of California Philippine-California Advanced Research Institute under project IIID-2016-005, by the Office of Naval Research's Multidisciplinary University Research Initiative Embedded Humans (N00014-16-1-2206), and by NASA under grants NNX12AR18A and UCSCMCA-14-022 (University Affiliated Research Center).

We thank Sylvia Herbert, Mahesh Vashishtha, and Somil Bansal for their collaboration on a paper related to system decomposition (49) and Qie Hu, Jaime Fisac, Kene Akametalu, and Casey Mackin for their collaboration on a paper related to unmanned aerial system traffic management (54), both of which are featured in this review.

LITERATURE CITED

1. Kong S, Gao S, Chen W, Clarke E. 2015. dReach: δ-reachability analysis for hybrid systems. In *Tools and Algorithms for the Construction and Analysis of Systems: TACAS 2015*, ed. C Baier, C Tinelli, pp. 200–5. Berlin: Springer
2. Duggirala PS, Mitra S, Viswanathan M, Potok M. 2015. C2E2: a verification tool for Stateflow models. In *Tools and Algorithms for the Construction and Analysis of Systems: TACAS 2015*, ed. C Baier, C Tinelli, pp. 68–82. Berlin: Springer
3. Greenstreet MR, Mitchell I. 1998. Integrating projections. In *Hybrid Systems: Computation and Control: HSCC 1998*, ed. TA Henzinger, SS Sastry, pp. 159–74. Berlin: Springer
4. Frehse G, Le Guernic C, Donzé A, Cotton S, Ray R, et al. 2011. SpaceEx: scalable verification of hybrid systems. In *Computer Aided Verification: CAV 2011*, ed. G Gopalakrishnan, S Qadeer, pp. 379–95. Berlin: Springer
5. Kurzhanski A, Varaiya P. 2000. Ellipsoidal techniques for reachability analysis: internal approximation. *Syst. Control Lett.* 41:201–11
6. Kurzhanski A, Varaiya P. 2002. On ellipsoidal techniques for reachability analysis. Part II: internal approximations box-valued constraints. *Optim. Methods Softw.* 17:207–37
7. Maidens JN, Kaynama S, Mitchell IM, Oishi MMK, Dumont GA. 2013. Lagrangian methods for approximating the viability kernel in high-dimensional systems. *Automatica* 49:2017–29
8. Chen X, Ábrahám E, Sankaranarayanan S. 2013. Flow*: an analyzer for non-linear hybrid systems. In *Computer Aided Verification: CAV 2013*, ed. N Sharygina, H Veith, pp. 258–63. Berlin: Springer
9. Althoff M. 2015. An introduction to CORA 2015. In *ARCH14-15: 1st and 2nd International Workshop on Applied veRification for Continuous and Hybrid Systems*, ed. G Frehse, M Althoff, pp. 120–51. Manchester, UK: EasyChair
10. Majumdar A, Vasudevan R, Tobenkin MM, Tedrake R. 2014. Convex optimization of nonlinear feedback controllers via occupation measures. *Int. J. Robot. Res.* 33:1209–30
11. Dreossi T, Dang T, Piazza C. 2016. Parallelotope bundles for polynomial reachability. In *HCCC '16: Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pp. 297–306. New York: ACM

12. Nilsson P, Ozay N. 2016. Synthesis of separable controlled invariant sets for modular local control design. In *2016 American Control Conference (ACC)*, pp. 5656–63. New York: IEEE
13. Althoff M, Krogh BH. 2014. Reachability analysis of nonlinear differential-algebraic systems. *IEEE Trans. Autom. Control* 59:371–83
14. Barron EN. 1990. Differential games with maximum cost. *Nonlinear Anal.* 14:971–89
15. Mitchell IM, Bayen AM, Tomlin CJ. 2005. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Trans. Autom. Control* 50:947–57
16. Margellos K, Lygeros J. 2011. Hamilton-Jacobi formulation for reach-avoid differential games. *Trans. Autom. Control* 56:1849–61
17. Bokanowski O, Zidani H. 2011. Minimal time problems with moving targets and obstacles. *IFAC Proc. Vol.* 44:2589–93
18. Darbon J, Osher S. 2016. Algorithms for overcoming the curse of dimensionality for certain Hamilton-Jacobi equations arising in control theory and elsewhere. *Res. Math. Sci.* 3:19
19. Hafner MR, Del Vecchio D. 2009. Computation of safety control for uncertain piecewise continuous systems on a partial order. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*, pp. 1671–77. New York: IEEE
20. Coogan S, Arcak M. 2015. Efficient finite abstraction of mixed monotone systems. In *HSCC '15: Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pp. 58–67. New York: ACM
21. Mitchell I. 2011. Scalable calculation of reach sets and tubes for nonlinear systems with terminal integrators: a mixed implicit explicit formulation. In *HSCC '11: Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control*, pp. 103–12. New York: ACM
22. Fisac JF, Chen M, Tomlin CJ, Sastry SS. 2015. Reach-avoid problems with time-varying dynamics, targets and constraints. In *HSCC '15: Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pp. 11–20. New York: ACM
23. Mitchell IM, Tomlin CJ. 2003. Overapproximating reachable sets by Hamilton-Jacobi projections. *J. Sci. Comput.* 19:323–46
24. Chen M, Herbert S, Tomlin CJ. 2016. Fast reachable set approximations via state decoupling disturbances. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 191–96. New York: IEEE
25. Kaynama S, Oishi M. 2009. Schur-based decomposition for reachability analysis of linear time-invariant systems. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference*, pp. 69–74. New York: IEEE
26. Kaynama S, Oishi M. 2013. A modified Riccati transformation for decentralized computation of the viability kernel under LTI dynamics. *IEEE Trans. Autom. Control* 58:2878–92
27. Varaiya PP. 1967. On the existence of solutions to a differential game. *SIAM J. Control* 5:153–62
28. Evans LC, Souganidis PE. 1984. Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. *Ind. Univ. Math. J.* 33:773–97
29. Tomlin C, Lygeros J, Sastry SS. 2000. A game theoretic approach to controller design for hybrid systems. *Proc. IEEE* 88:949–70
30. Bokanowski O, Forcadel N, Zidani H. 2010. Reachability and minimal times for state constrained nonlinear problems without any controllability assumption. *SIAM J. Control Optim.* 48:4292–316
31. Ding J, Sprinkle J, Sastry SS, Tomlin CJ. 2008. Reachability calculations for automated aerial refueling. In *2008 47th IEEE Conference on Decision and Control*, pp. 3706–12. New York: IEEE
32. Chen M, Hu Q, Mackin C, Fisac J, Tomlin CJ. 2015. Safe platooning of unmanned aerial vehicles via reachability. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 4695–701. New York: IEEE
33. Bayen AM, Mitchell IM, Oishi M, Tomlin CJ. 2007. Aircraft autolander safety analysis through optimal control-based reach set computation. *AIAA J. Guid. Control Dyn.* 30:68–77
34. Huang H, Ding J, Zhang W, Tomlin C. 2011. A differential game approach to planning in adversarial scenarios: a case study on capture-the-flag. In *2011 IEEE International Conference on Robotics and Automation*, pp. 1451–56. New York: IEEE
35. Joint Plan. Dev. Office. 2013. *Unmanned aircraft systems (UAS) comprehensive plan – a report on the nation's UAS path forward*. Tech. Rep., Fed. Aviat. Admin., Washington, DC
36. Amazon. 2017. *Amazon Prime Air*. **http://www.amazon.com/b?node=8037720011**

37. BBC. 2015. Google plans drone delivery service for 2017. *BBC News*, Nov. 2. **http://www.bbc.co.uk/news/technology-34704868**
38. AUVSI News. 2016. UAS aid in South Carolina tornado investigation. *AUVSI News*, Jan. 29. **http://www.auvsi.org/blogs/auvsi-news/2016/01/29/tornado**
39. Prevot T, Rios J, Kopardekar P, Robinson JE III, Johnson M, Jung J. 2016. UAS Traffic Management (UTM) concept of operations to safely enable low altitude flight operations. In *16th AIAA Aviation Technology*, *Integration*, *and Operations Conference*, chap. 2016-3292. Reston, VA: AIAA
40. Mitchell IM. 2008. The flexible, extensible and efficient toolbox of level set methods. *J. Sci. Comput.* 35:300–29
41. Osher S, Fedkiw R. 2003. Hamilton-Jacobi equations. In *Level Set Methods and Dynamic Implicit Surfaces*, pp. 47–54. New York: Springer
42. Sethian JA. 1996. A fast marching level set method for monotonically advancing fronts. *PNAS* 93:1591–95
43. Coddington EA, Levinson N. 1955. Existence and uniqueness of solutions. In *Theory of Ordinary Differential Equations*, pp. 1–42. New York: McGraw-Hill
44. Mitchell IM. 2007. Comparing forward and backward reachability as tools for safety analysis. In *Hybrid Systems: Computation and Control: HSCC 2007*, ed. A Bemporad, A Bicchi, G Buttazzo, pp. 428–43. Berlin: Springer
45. Crandall MG, Lions PL. 1983. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Am. Math. Soc.* 277:1–42
46. Crandall MG, Evans LC, Lions PL. 1984. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Trans. Am. Math. Soc.* 282:487–502
47. Callier FM, Desoer CA. 1991. The system representation $R = [A,B,C,D]$, part II. In *Linear System Theory*, pp. 103–39. New York: Springer
48. Sastry SS. 1999. Linearization by state feedback. In *Nonlinear Systems: Analysis*, *Stability*, *and Control*, pp. 384–448. New York: Springer
49. Chen M, Herbert SL, Vashishtha MS, Bansal S, Tomlin CJ. 2018. Decomposition of reachable sets and tubes for a class of nonlinear systems. *IEEE Trans. Autom. Control.* In press. **https://doi.org/10.1109/TAC.2018.2797194**
50. Gillula JH, Hoffmann GM, Haomiao Huang, Vitus MP, Tomlin CJ. 2011. Applications of hybrid reachability analysis to robotic aerial vehicles. *Int. J. Robot. Res.* 30:335–54
51. Kaynama S, Oishi M. 2011. Complexity reduction through a Schur-based decomposition for reachability analysis of linear time-invariant systems. *Int. J. Control* 84:165–79
52. Tice BP. 1991. Unmanned aerial vehicles – the force multiplier of the 1990s. *Airpower J.* 5:41–55
53. Haulman DL. 2003. *U.S. unmanned aerial vehicles in combat*, *1991–2003.* Tech. Rep., Air Force Hist. Res. Agency, Maxwell Air Force Base, Montgomery, AL
54. Chen M, Hu Q, Fisac JF, Akametalu K, Mackin C, Tomlin CJ. 2017. Reachability-based safety and goal satisfaction of unmanned aerial platoons on air highways. *AIAA J. Guid. Control Dyn.* 40:1360–73
55. Chen M, Fisac JF, Sastry SS, Tomlin CJ. 2015. Safe sequential path planning of multi-vehicle systems via double-obstacle Hamilton-Jacobi-Isaacs variational inequality. In *2015 European Control Conference (ECC)*, pp. 3304–9. New York: IEEE
56. Bansal S, Chen M, Fisac JF, Tomlin CJ. 2017. Safe sequential path planning under disturbances and imperfect information. In *2017 American Control Conference (ACC)*, pp. 5550–55. New York: IEEE
57. Chen M, Bansal S, Fisac JF, Tomlin CJ. 2018. Robust sequential path planning under disturbances and adversarial intruder. *IEEE Trans. Control Syst. Technol.* In press
58. Chen M, Shih JC, Tomlin CJ. 2016. Multi-vehicle collision avoidance via Hamilton-Jacobi reachability and mixed integer programming. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 1695–700. New York: IEEE
59. Chen M, Zhou Z, Tomlin CJ. 2017. Multiplayer reach-avoid games via pairwise outcomes. *IEEE Trans. Autom. Control* 62:1451–57
60. Pierson A, Wang Z, Schwager M. 2017. Intercepting rogue robots: an algorithm for capturing multiple evaders with multiple pursuers. *IEEE Robot. Autom. Lett.* 2:530–37

Annual Review of Control, Robotics, and Autonomous Systems

Volume 1, 2018

Contents

Toward Robotic Manipulation
Matthew T. Mason 1

Autonomous Flight
Sarah Tang and Vijay Kumar 29

Soft Micro- and Nanorobotics
Chengzhi Hu, Salvador Pané, and Bradley J. Nelson 53

Distributed Optimization for Control
Angelia Nedić and Ji Liu 77

Game Theory and Control
Jason R. Marden and Jeff S. Shamma 105

The Bountiful Intersection of Differential Geometry, Mechanics, and Control Theory
Andrew D. Lewis 135

Sampling-Based Methods for Motion Planning with Constraints
Zachary Kingston, Mark Moll, and Lydia E. Kavraki 159

Planning and Decision-Making for Autonomous Vehicles
Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus 187

Synthesis for Robots: Guarantees and Feedback for Robot Behavior
Hadas Kress-Gazit, Morteza Lahijanian, and Vasumathi Raman 211

Invariant Kalman Filtering
Axel Barrau and Silvère Bonnabel 237

Data-Driven Predictive Control for Autonomous Systems
Ugo Rosolia, Xiaojing Zhang, and Francesco Borrelli 259

A Separation Principle for Control in the Age of Deep Learning
Alessandro Achille and Stefano Soatto 287

Privacy in Control and Dynamical Systems
Shuo Han and George J. Pappas 309

Hamilton–Jacobi Reachability: Some Recent Theoretical Advances and Applications in Unmanned Airspace Management
Mo Chen and Claire J. Tomlin 333

Design of Materials and Mechanisms for Responsive Robots
Elliot W. Hawkes and Mark R. Cutkosky 359

Haptics: The Present and Future of Artificial Touch Sensation
Heather Culbertson, Samuel B. Schorr, and Allison M. Okamura 385

Programming Cells to Work for Us
Yili Qian, Cameron McBride, and Domitilla Del Vecchio 411

Autonomy in Rehabilitation Robotics: An Intersection
Brenna D. Argall 441

Medical Technologies and Challenges of Robot-Assisted Minimally Invasive Intervention and Diagnostics
Nabil Simaan, Rashid M. Yasin, and Long Wang 465

Errata

An online log of corrections to *Annual Review of Control, Robotics, and Autonomous Systems* articles may be found at http://www.annualreviews.org/errata/control