Computing the Viability Kernel Using Maximal Reachable Sets[*]

Shahab Kaynama
Department of Electrical and
Computer Engineering
University of British Columbia
Vancouver, BC, Canada
kaynama@ece.ubc.ca

John Maidens
Department of Electrical and
Computer Engineering
University of British Columbia
Vancouver, BC, Canada
jmaidens@ece.ubc.ca

Meeko Oishi
Department of Electrical and
Computer Engineering
University of New Mexico
Albuquerque, NM, USA
oishi@unm.edu

Ian M. Mitchell
Department of Computer
Science
University of British Columbia
Vancouver, BC, Canada
mitchell@cs.ubc.ca

Guy A. Dumont
Department of Electrical and
Computer Engineering
University of British Columbia
Vancouver, BC, Canada
guyd@ece.ubc.ca

ABSTRACT

We present a connection between the viability kernel and maximal reachable sets. Current numerical schemes that compute the viability kernel suffer from a complexity that is exponential in the dimension of the state space. In contrast, extremely efficient and scalable techniques are available that compute maximal reachable sets. We show that under certain conditions these techniques can be used to conservatively approximate the viability kernel for possibly high-dimensional systems. We demonstrate the results on two practical examples, one of which is a seven-dimensional problem of safety in anesthesia.

Categories and Subject Descriptors

J.2 [**Physical Sciences & Engineering**]: Engineering; I.6.4 [**Simulation & Modeling**]: Model Validation & Analysis

Keywords

reachability, viability, controlled-invariance, set-theoretic methods, scalability, safety-critical systems

1. INTRODUCTION

Reachability analysis and viability theory provide solid frameworks for control synthesis and trajectory analysis of constrained dynamical systems in a set-valued fashion (cf. [1, 18, 3]) and have

been utilized in diverse applications such as aircraft collision avoidance and air traffic management [2, 27, 32], stabilization of underwater vehicles [35], and control of uncertain oscillatory systems [7], among others.

Reachability analysis identifies the set of states backward (forward) reachable by a constrained dynamical system from a given target (initial) set of states. The notions of *maximal* and *minimal* reachability analysis were introduced in [29]. Their corresponding constructs differ in how the time variable and the bounded input are quantified. In formation of the maximal reachability construct, the input tries to steer as many states as possible to the target set. In formation of the minimal reachability construct, the trajectories reach the target set regardless of the input applied. Based on these differences, the maximal and minimal *reachable sets* and *tubes* (the set of states traversed by the trajectories over the time horizon [29, 19]) are formed.

Viability theory can provide insight into the behavior of the trajectories inside a given constraint set. The *viability kernel* is the set of initial states for which there exists an input (drawing from a specified set) such that the system respects the state constraint for all time. Another closely related construct is the *invariance kernel* which contains the set of states that remain in the constraint set for all possible inputs for all time.

It is shown in [29] and [1] that the minimal reachable tube and the viability kernel are the only constructs that can be used to prove safety/viability of the system and to synthesize inputs (controllers) that preserve this safety. Since the viability kernel and the minimal reachable tube are duals of one another, they need not be treated separately. In this paper we only focus on the former.

In [17], we formally examined the existing connections between various backward constructs generated by reachability and viability frameworks. Here, we will draw a new connection between the viability kernel and the maximal reachable sets of a constrained dynamical system.

Backward constructs are computed using two separate categories of algorithms [29]: The *Eulerian methods* (e.g. [32, 36, 6, 9]) are capable of computing the viability kernel (and by duality, the minimal reachable tubes). Although versatile in terms of ability to handle complex dynamics and constraints, these algorithms rely on gridding the state space and therefore their computational complexity increases exponentially with the dimension of the state, ren-

dering them impractical for systems of dimensionality higher than three or four. The second category of algorithms are *Lagrangian methods* (e.g. [19, 22, 12, 11, 13]) that follow the trajectories and compute the maximal reachable sets and tube in a scalable and efficient manner. Their computational complexity is usually polynomial in time and space, making them suitable for application to high-dimensional systems.

Our main contribution in this paper is as follows: By bridging the gap between the viability kernel and the maximal reachable sets we pave the way for more efficient computation of the viability kernel through the use of Lagrangian algorithms. Significant reduction in the computational costs can be achieved since instead of a single calculation with exponential complexity one can perform a series of calculations with polynomial complexity.

Complexity reduction for the computation of the viability kernel and the minimal reachable tube has previously been addressed using Hamilton-Jacobi projections [33] and structure decomposition [38, 16, 15, 31].

In Section 2 we formally define the viability kernel and the maximal reachable set and formulate our problem. Section 3 presents our main results: the connection between these constructs in continuous time and discrete time. Computational algorithms are provided in Section 4, and the results are demonstrated on two practical examples in Section 5. Conclusions are provided in Section 6.

1.1 Basic Notations

For any two subsets $\mathcal{A}$ and $\mathcal{S}$ of the Euclidean space $\mathbb{R}^n$, the *erosion* of $\mathcal{A}$ by $\mathcal{S}$ is defined as $\mathcal{A} \ominus \mathcal{S} := \{x \in \mathbb{R}^n \mid \mathcal{S}_x \subseteq \mathcal{A}\}$ where $\mathcal{S}_x := \{y + x \mid y \in \mathcal{S}\}, \forall x \in \mathbb{R}^n$. We denote by $\overset{\circ}{\mathcal{S}}$ the interior of $\mathcal{S}$ and use $\mathcal{B}(\gamma)$ to denote a norm-ball of radius $\gamma \in \mathbb{R}^+$ centered at the origin in $\mathbb{R}^n$.

2. PROBLEM FORMULATION

Consider a continuously valued dynamical system

$$\mathcal{L}(x(t)) = f(x(t), u(t)), \quad x(0) = x_0 \tag{1}$$

with state space $\mathcal{X} := \mathbb{R}^n$, state vector $x(t) \in \mathcal{X}$, and input $u(t) \in \mathcal{U}$ where $\mathcal{U}$ is a compact and convex subset of $\mathbb{R}^m$. Depending on whether the system evolves in continuous time ($t \in \mathbb{R}^+$) or discrete time ($t \in \mathbb{Z}^+$), $\mathcal{L}(\cdot)$ denotes the derivative operator or the unit forward shift operator, respectively. In the continuous-time case, we assume that the vector field $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is Lipschitz in x and continuous in u. Let $\mathscr{U}_{[0,t]} := \{u : [0, t] \to \mathbb{R}^m$ measurable, $u(t) \in \mathcal{U}$ a.e.$\}$. With an arbitrary, finite time horizon $\tau > 0$, for every $t \in [0, \tau]$, $x_0 \in \mathcal{X}$, and $u(\cdot) \in \mathscr{U}_{[0,t]}$, there exists a unique trajectory $x_{x_0}^u : [0, t] \to \mathcal{X}$ that satisfies the initial condition $x_{x_0}^u(0) = x_0$ and the differential/difference equation (1) almost everywhere. When clear from the context, we shall drop the subscript and superscript from the trajectory notation.

Take a state constraint ("safe") set $\mathcal{K}$ with a nonempty interior. We examine the following *backward* constructs:

Definition 1. (**Viability Kernel**) The (finite horizon) viability kernel of $\mathcal{K}$ is the set of all initial states in $\mathcal{K}$ for which there exists an input such that the trajectories emanating from those states remain within $\mathcal{K}$ for all time $t \in [0, \tau]$:

$$Viab_{[0,\tau]}(\mathcal{K}) := \big\{x_0 \in \mathcal{X} \mid \exists u(\cdot) \in \mathscr{U}_{[0,\tau]}, \forall t \in [0, \tau],$$
$$x_{x_0}^u(t) \in \mathcal{K}\big\}. \tag{2}$$

Definition 2. (**Maximal Reachable Set**) The maximal reachable set at time t is the set of initial states for which there exists an input such that the trajectories emanating from those states reach $\mathcal{K}$ exactly at time t:

$$Reach_t^\sharp(\mathcal{K}) := \big\{x_0 \in \mathcal{X} \mid \exists u(\cdot) \in \mathscr{U}_{[0,t]}, x_{x_0}^u(t) \in \mathcal{K}\big\}. \tag{3}$$

Problem 1. Express the viability kernel $Viab_{[0,\tau]}(\mathcal{K})$ in terms of the maximal reachable sets $Reach_t^\sharp(\mathcal{K})$, $t \in [0, \tau]$.

The viability kernel has traditionally been computed using Eulerian methods [6, 32, 26]. By addressing Problem 1, we enable the use of efficient Lagrangian methods for the computation of the viability kernel for possibly high-dimensional systems.

3. MAIN RESULTS

3.1 Continuous-Time Systems

Consider the case in which (1) is the continuous-time system

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0, \quad t \in \mathbb{R}^+. \tag{4}$$

We will show that we can approximate $Viab_{[0,\tau]}(\mathcal{K})$ by considering a nested sequence of sets that are reachable in small sub-time intervals of $[0, \tau]$.

Definition 3. We say that a vector field $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is bounded on $\mathcal{K}$ if there exists a norm $\|\cdot\| : \mathcal{X} \to \mathbb{R}^+$ and a real number $M > 0$ such that for all $x \in \mathcal{K}$ and $u \in \mathcal{U}$ we have $\|f(x, u)\| \leq M$.

Definition 4. A partition $P = \{t_0, t_1, \ldots, t_n\}$ of $[0, \tau]$ is a set of distinct points $t_0, t_1, \ldots, t_n \in [0, \tau]$ with $t_0 = 0$, $t_n = \tau$ and $t_0 < t_1 < \cdots < t_n$. Further, we denote
- the number n of intervals $[t_{k-1}, t_k]$ in P by $|P|$,
- the size of the largest interval by $\|P\| := \max_{k=1}^{|P|}\{t_{k+1} - t_k\}$, and
- the set of all partitions of $[0, \tau]$ by $\mathscr{P}([0, \tau])$.

Definition 5. For a signal $u : [0, \tau] \to \mathcal{U}$ and a partition $P = \{t_0, \ldots, t_n\}$ of $[0, \tau]$, define the tokenization $\{u_k\}_k$ of u corresponding to P as the set of functions $u_k : [0, t_k - t_{k-1}] \to \mathcal{U}$ such that

$$u_k(t) = u(t + t_{k-1}). \tag{5}$$

Conversely, for a set of functions $u_k : [0, t_k - t_{k-1}] \to \mathcal{U}$, define their concatenation $u : [0, \tau] \to \mathcal{U}$ as

$$u(t) = u_k(t - t_{k-1}) \quad \text{if} \quad t \neq 0, \tag{6a}$$
$$u(0) = u_1(0) \tag{6b}$$

where k is the unique integer such that $t \in (t_{k-1}, t_k]$.

Definition 6. The $\|\cdot\|$-distance of a point $x \in \mathcal{X}$ from a nonempty set $\mathcal{S} \subset \mathcal{X}$ is defined as
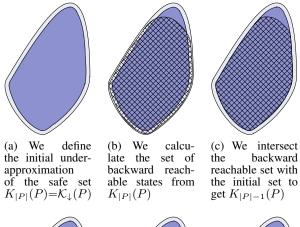
$$\text{dist}(x, \mathcal{S}) := \inf_{s \in \mathcal{S}} \|x - s\|. \tag{7}$$

For a fixed set $\mathcal{S}$, the map $x \mapsto \text{dist}(x, \mathcal{S})$ is continuous.

3.1.1 Computing an Under-Approximation of the Viability Kernel

Assume that the vector field f is bounded by M in the norm $\|\cdot\|$. We begin by defining an under-approximation of the state constraint set (Figure 1(a)):

$$\mathcal{K}_\downarrow(P) := \{x \in \mathcal{K} \mid \text{dist}(x, \mathcal{K}^c) \geq M\|P\|\}. \tag{8}$$

(a) We define the initial under-approximation of the safe set $K_{|P|}(P) = \mathcal{K}_\downarrow(P)$

(b) We calculate the set of backward reachable states from $K_{|P|}(P)$

(c) We intersect the backward reachable set with the initial set to get $K_{|P|-1}(P)$

(d) Next, we calculate the set of backward reachable states from $K_{|P|-1}(P)$

(e) Again, we intersect the backward reachable set with the initial set to get a new set $K_{|P|-2}(P)$

(f) By repeating this process, we reach an under-approximation $\overset{\circ}{K}_0(P)$ of the viability kernel.
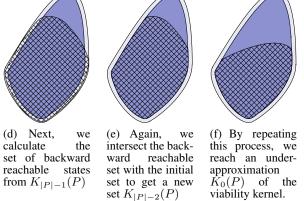
Figure 1: Iteratively constructing an under-approximation of $Viab_{[0,\tau]}(\mathcal{K})$.

We under-approximate $\mathcal{K}$ by a distance $M\|P\|$ because we are only considering the system's state at discrete times $t_0, t_1, \ldots, t_n$. At a time t in the interval $[t_i, t_{i+1}]$, a trajectory $x(\cdot)$ can travel a distance of at most

$$\|x(t) - x(t_i)\| \le \int_{t_i}^t \|\dot{x}(\tau)\| d\tau \le M(t - t_i) \le M\|P\| \quad (9)$$

from its initial location $x(t_i)$. As we shall see, formulating the subset (8) will ensure that the state does not leave $\mathcal{K}$ at any time during $[0, \tau]$.

This set defines the first step of our recursion. We then define a sequence of $|P|$ sets recursively:

$$K_{|P|}(P) = \mathcal{K}_\downarrow(P), \tag{10a}$$
$$K_{k-1}(P) = \mathcal{K}_\downarrow(P) \cap Reach^\sharp_{t_k - t_{k-1}}(K_k(P))$$
$$\text{for} \quad k \in \{1, \ldots, |P|\}. \tag{10b}$$

At each time step, we calculate the set of states from which you can reach $K_k(P)$, then intersect this set with the set of safe states (see Figure 1). The final set $K_0(P)$ is an approximation of $Viab_{[0,\tau]}(\mathcal{K})$.

Note that the resulting set depends on our choice of a partition P of the time interval $[0, \tau]$. We claim that for any partition P, $K_0(P)$ is an under-approximation.

PROPOSITION 1. *Suppose that the vector field $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is bounded on a set $\mathcal{K} \subseteq \mathcal{X}$. Then for any partition P of $[0, \tau]$ the final set $K_0(P)$ defined by the recurrence relation (10) satisfies*

$$K_0(P) \subseteq Viab_{[0,\tau]}(\mathcal{K}). \tag{11}$$

PROOF. Since f is bounded on $\mathcal{K}$, there exists a norm $\|\cdot\|$ and a real number $M > 0$ with $\|f(x, u)\| \le M$ for all $x \in \mathcal{K}$. Now, fix a partition P of $[0, \tau]$ and take a point $x_0 \in K_0(P)$. By the construction of $K_0(P)$, this means that for each $k = 1, \ldots, |P|$ there is some point $x_k \in K_k(P)$ and an input $u_k : [0, t_k - t_{k-1}] \to \mathcal{U}$ such that x_k can be reached from x_{k-1} at time $t_k - t_{k-1}$ using input u_k. Thus, taking the concatenation of the inputs u_k, we get an input $u : [0, \tau] \to \mathcal{U}$ such that the solution $x : [0, \tau] \to \mathcal{X}$ to the initial value problem $\dot{x} = f(x, u)$, $x(0) = x_0$, satisfies $x(t_k) = x_k \in K_k(P) \subseteq \{x \in \mathcal{K} \mid \text{dist}(x, \mathcal{K}^c) \ge M\|P\|\}$. We claim that this guarantees that $x(t) \in \mathcal{K}$ for all $t \in [0, \tau]$. Indeed, any $t \in [0, \tau]$ lies is some interval $[t_k, t_{k+1}]$. Since f is bounded by M, we have

$$\|x(t) - x(t_k)\| \le M(t - t_k) < M(t_{k+1} - t_k) \le M\|P\|. \tag{12}$$

Further, $x(t_k) \in K_k(P)$ implies that $\text{dist}(x(t_k), \mathcal{K}^c) \ge M\|P\|$. Combining these, we see that

$$\text{dist}(x(t), \mathcal{K}^c) \ge \text{dist}(x(t_k), \mathcal{K}^c) - \|x(t) - x(t_k)\|$$
$$> M\|P\| - M\|P\| = 0 \tag{13}$$

and hence $x(t) \in \mathcal{K}$. Thus, $x_0 \in Viab_{[0,\tau]}(\mathcal{K})$. $\square$

3.1.2 Precision of the Approximation

The approximation can be made to be arbitrarily precise by choosing a sufficiently fine partition. This is true in the sense that the union of the approximating sets $K_0(P)$ taken over all possible partitions P of $[0, \tau]$ is bounded between the viability kernels of $\mathcal{K}$ and its interior $\overset{\circ}{\mathcal{K}}$.

PROPOSITION 2. *Suppose that the vector field $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is bounded on a set $\mathcal{K} \subseteq \mathcal{X}$. Then we have*

$$Viab_{[0,\tau]}(\overset{\circ}{\mathcal{K}}) \subseteq \bigcup_{P \in \mathscr{P}([0,\tau])} K_0(P) \subseteq Viab_{[0,\tau]}(\mathcal{K}). \tag{14}$$

In particular, when $\mathcal{K}$ is open,

$$\bigcup_{P \in \mathscr{P}([0,\tau])} K_0(P) = Viab_{[0,\tau]}(\mathcal{K}). \tag{15}$$

PROOF. The second inclusion in (14) follows directly from Proposition 1. To prove the first inclusion, take a state $x_0 \in Viab_{[0,\tau]}(\overset{\circ}{\mathcal{K}})$. There exists an input $u : [0, \tau] \to \mathcal{U}$ such that the solution $x(\cdot)$ to the initial value problem $\dot{x} = f(x, u)$, $x(0) = x_0$, satisfies $x(t) \in \overset{\circ}{\mathcal{K}}$ for all $t \in [0, \tau]$. Since $\overset{\circ}{\mathcal{K}}$ is open, for any $x \in \overset{\circ}{\mathcal{K}}$ we have $\text{dist}(x, \mathcal{K}^c) > 0$. Further, $x : [0, \tau] \to \mathcal{X}$ is continuous so the function $t \mapsto \text{dist}(x(t), \mathcal{K}^c)$ is continuous on the compact set $[0, \tau]$. Thus, we can define $d > 0$ to be its minimum value. Now take a partition P of $[0, \tau]$ such that $M\|P\| < d$. We need to show that $x_0 \in K_0(P)$.

First note that our partition P is chosen such that $\text{dist}(x(t), \mathcal{K}^c) > M\|P\|$ for all $t \in [0, \tau]$. Hence $x(t_k) \in K_{|P|}(P)$ for all $k = 0, \ldots, |P|$. To show that $x(t_{k-1}) \in Reach^\sharp_{t_k - t_{k-1}}(K_k(P))$ for all $k = 1, \ldots, |P|$, consider the tokenization $\{u_k\}_k$ of the input u corresponding to P. It is easy to verify that for all k, we can reach $x(t_k)$ from $x(t_{k-1})$ at time $t_k - t_{k-1}$ using input u_k. Thus, in particular, we have $x_0 = x(t_0) \in Reach^\sharp_{t_1 - t_0}(K_1(P))$. So $x_0 \in K_0(P)$. Hence $Viab_{[0,\tau]}(\overset{\circ}{\mathcal{K}}) \subseteq \bigcup_{P \in \mathscr{P}([0,\tau])} K_0(P)$. $\square$

3.2 Discrete-Time Systems

Consider the case in which (1) is the discrete-time system

$$x(t + 1) = f(x(t), u(t)), \quad x(0) = x_0, \quad t \in \mathbb{Z}^+. \tag{16}$$

Computing $Viab_{[0,\tau]}(\mathcal{K})$ under this system is a particular case of the results presented in Section 3.1. Define a sequence of sets recursively as

$$K_n = \mathcal{K}, \qquad\qquad (17a)$$

$$K_{k-1} = \mathcal{K} \cap Reach_1^\sharp(K_k), \quad k \in \{1,\dots,n\} \qquad (17b)$$

where $\tau = n$ and $Reach_1^\sharp(\cdot)$ is the unit time-step maximal reachable set.

PROPOSITION 3. *Let K_0 be the final set obtained from the recurrence relation* (17). *Then,*

$$Viab_{[0,\tau]}(\mathcal{K}) = K_0. \qquad (18)$$

PROOF. Notice that the time variable t is integer valued. As a result, the tokenization of the input signal u is a discrete sequence $\{u_k\}_k$ with $u_k := u(t)$ with $t = k-1$ for $k = 1,\dots,n$.

To show $K_0 \subseteq Viab_{[0,\tau]}(\mathcal{K})$, via recursion (17) we have that at each step k there exists u_k such that $x_{k-1} \in K_{k-1}$ reaches $x_k \in K_k$. Thus, $x_0 \in K_0$ implies there exists a concatenation $u(\cdot) = \{u_k\}_k \in \mathscr{U}_{[0,\tau]}$ such that $x(t) \in \mathcal{K}$ for all $t \in [0,\tau]$. Therefore, $x_0 \in Viab_{[0,\tau]}(\mathcal{K})$.

To show $Viab_{[0,\tau]}(\mathcal{K}) \subseteq K_0$, take $x_0 \in Viab_{[0,\tau]}(\mathcal{K})$. There exists $u(\cdot) = \{u_k\}_k$ such that $x(t) \in \mathcal{K}$ for every t. Using the tokenization of $\{u_k\}_k$ we can verify that for some u_k we can reach $x_k := x(t+1)$ from $x_{k-1} := x(t)$. Hence, $x_{k-1} \in Reach_1^\sharp(K_k)$ for all $k \in \{1,\dots,n\}$. In particular, for $k = 1$ we have $x_0 := x(0) \in Reach_1^\sharp(K_1)$. Thus, $x_0 \in \mathcal{K} \cap Reach_1^\sharp(K_1) = K_0$. $\square$

Remark 1. Note that the above iterative scheme is closely related to the set-valued description of the discrete viability kernel presented in [36, 6] and the recursive construction of the controlled-invariant set for discrete-time systems presented in [3].

4. COMPUTATIONAL ALGORITHMS

Thanks to the results in the previous section, any technique that is capable of computing the maximal reachable set can be used to compute the viability kernel. Most currently available Lagrangian methods yield an (under- and/or over-) approximation of the maximal reachable set. The viability kernel should not be over-approximated since an over-approximation would contain initial states for which the viability of the system is inevitably at stake. Thus, to correctly compute $Viab_{[0,\tau]}(\mathcal{K})$ all approximations must be in the form of under-approximations.

Every step of the recursions (10) and (17) involves a reachability computation and an intersection operation. Ideally, the sets that are being intersected should be drawn from classes of shapes that are closed under such an operation, e.g. polytopes. However, the currently available reachability techniques that are based on polytopes (e.g. [24]) do not, in general, scale well with the dimension of the state. Moreover, the scalable reachability techniques, such as the methods of zonotopes [12], ellipsoids [19, 23], and support functions [11], generate sets that may prove to be difficult to transform into a polytope. For instance, one may compute a polytopic under-approximation of the reachable sets using their support functions based on the approach presented in [25]. However, that approach requires calculation of the facet representation of the resulting polytopes from their vertices before each intersection operation, which is known to be computationally demanding in higher dimensions.

4.1 A Piecewise Ellipsoidal Approach

Here we showcase our results using an efficient algorithm, based on ellipsoidal techniques [19] implemented in the Ellipsoidal Toolbox (ET) [22], that sacrifices accuracy in exchange for scalability.

We consider the case in which (1) is a linear time-invariant (LTI) system

$$\mathcal{L}(x(t)) = Ax(t) + Bu(t) \qquad (19)$$

with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$.

An *ellipsoid* in $\mathbb{R}^n$ is defined as

$$\mathcal{E}(q, Q) := \left\{ x \in \mathbb{R}^n \mid \langle (x-q), Q^{-1}(x-q) \rangle \le 1 \right\} \qquad (20)$$

with center $q \in \mathbb{R}^n$ and shape matrix $\mathbb{R}^{n \times n} \ni Q = Q^T \succ 0$. A *piecewise ellipsoidal* set is the union of a finite number of ellipsoids.

Among many advantages, ellipsoidal techniques [19, 22] allow for an efficient computation of *under-approximations* of the maximal reachable sets, making them a particularly attractive choice for the reachability computations involved in our formulation of the viability kernel.

Suppose $\mathcal{K}$ and $\mathcal{U}$ are (or can be closely under-approximated as) compact ellipsoids with nonempty interior. Consider the continuous-time case and the recursion (10). (The arguments in the discrete-time case are similar.) Given a partition P and some $k \in \{1,\dots,|P|\}$, let $K_k(P) = \mathcal{E}(x_\delta, X_\delta) \subset \mathcal{X}$. As in [20], with $\mathcal{N} := \{v \in \mathbb{R}^n \mid \langle v, v \rangle = 1\}$ and $\delta := t_k - t_{k-1}$ we have

$$Reach_{\delta-t}^\sharp(K_k(P)) = \bigcup_{\ell_\delta \in \mathcal{N}} \mathcal{E}(x^*(t), X_\ell^-(t)), \quad \forall t \in [0,\delta],$$
$$\qquad (21)$$

where $x^*(t)$ and $X_\ell^-(t)$ are the center and the shape matrix of the *internal approximating* ellipsoid at time t that is tangent to $Reach_{\delta-t}^\sharp(K_k(P))$ in the direction $\ell(t) \in \mathbb{R}^n$. For a fixed $\ell(\delta) = \ell_\delta \in \mathcal{N}$, the direction $\ell(t)$ is obtained from the adjoint equation $\dot{\ell}(t) = -A^T \ell(t)$. The center $x^*(t)$ (with $x^*(\delta) = x_\delta$) and the shape matrix $X_\ell^-(t)$ (with $X_\ell^-(\delta) = X_\delta$) are determined from differential equations described in [21]. (cf. [23] for their discrete-time counterparts.)

In practice, only a finite number of directions is used for the maximal reachable set computations. Let $\mathcal{M}$ be a finite subset of $\mathcal{N}$. Then,

$$Reach_{\delta-t}^\sharp(K_k(P)) \supseteq \bigcup_{\ell_\delta \in \mathcal{M}} \mathcal{E}(x^*(t), X_\ell^-(t)), \quad \forall t \in [0,\delta].$$
$$\qquad (22)$$

Note that the under-approximation in (22) is in general an arbitrarily shaped, non-convex set. Performing our desired operations on this set while maintaining efficiency may be difficult, if not impossible.

Now, consider the final backward reachable set $Reach_\delta^\sharp(K_k(P))$ and let $Reach_\delta^{\sharp(\tilde{\ell}_\delta)}(K_k(P))$ denote the maximal reachable set corresponding to a single terminal direction $\tilde{\ell}_\delta := \tilde{\ell}(\delta) \in \mathcal{M}$. We have that

$$Reach_\delta^{\sharp(\tilde{\ell}_\delta)}(K_k(P)) = \mathcal{E}(x^*(0), X_{\tilde{\ell}}^-(0))$$
$$\subseteq \bigcup_{\ell_\delta \in \mathcal{M}} \mathcal{E}(x^*(0), X_\ell^-(0)) \qquad (23)$$
$$\subseteq Reach_\delta^\sharp(K_k(P)).$$

Therefore, the reachable set computed for a single direction is an *ellipsoidal subset* of the actual reachable set.

Let $\bigcirc(\cdot)$ be a function that maps a set to its maximum volume inscribed ellipsoid. Algorithms 1 and 2 compute a piecewise ellipsoidal under-approximation of $Viab_{[0,\tau]}(\mathcal{K})$ for continuous-time and discrete-time systems, respectively.

Algorithm 1 Piecewise ellipsoidal approximation of $Viab_{[0,\tau]}(\mathcal{K})$ (continuous-time)

1: Choose $P \in \mathscr{P}([0,\tau])$ ▷ Affects precision of approximation
2: $K_{|P|}(P) \leftarrow \mathcal{K} \ominus \mathcal{B}(M\|P\|)$
                   ▷ Find $\{x \in \mathcal{K} \mid \text{dist}(x, \mathcal{K}^c) \geq M\|P\|\}$
3: $K_0^*(P) \leftarrow \emptyset$
4: **while** $\mathcal{M} \neq \emptyset$ **do**
5:     $l \leftarrow \ell_\tau \in \mathcal{M}$
6:     $k \leftarrow |P|$
7:     **while** $k \neq 0$ **do**
8:        **if** $K_k(P) = \emptyset$ **then**
9:           $K_0(P) \leftarrow \emptyset$
10:           **break**
11:        **end if**
12:        $\mathcal{G} \leftarrow Reach_{t_k - t_{k-1}}^{\sharp(l)}(K_k(P))$
             ▷ Compute the maximal reach set along the direction l
13:        $K_{k-1}(P) \leftarrow \bigcirc(K_{|P|}(P) \cap \mathcal{G})$
             ▷ Find the maximum volume inscribed ellipsoid in $K_{|P|}(P) \cap \mathcal{G}$
14:        $k \leftarrow k - 1$
15:     **end while**
16:     $K_0^*(P) \leftarrow K_0^*(P) \cup K_0(P)$
17:     $\mathcal{M} \leftarrow \mathcal{M} \backslash \{l\}$
18: **end while**
19: **return** $(K_0^*(P))$

Algorithm 2 Piecewise ellipsoidal approximation of $Viab_{[0,\tau]}(\mathcal{K})$ (discrete-time)

1: $K_n \leftarrow \mathcal{K}$
2: $K_0^* \leftarrow \emptyset$
3: **while** $\mathcal{M} \neq \emptyset$ **do**
4:     $l \leftarrow \ell_\tau \in \mathcal{M}$
5:     $k \leftarrow n$
6:     **while** $k \neq 0$ **do**
7:        **if** $K_k = \emptyset$ **then**
8:           $K_0 \leftarrow \emptyset$
9:           **break**
10:        **end if**
11:        $\mathcal{G} \leftarrow Reach_1^{\sharp(l)}(K_k)$
12:        $K_{k-1} \leftarrow \bigcirc(K_n \cap \mathcal{G})$
13:        $k \leftarrow k - 1$
14:     **end while**
15:     $K_0^* \leftarrow K_0^* \cup K_0$
16:     $\mathcal{M} \leftarrow \mathcal{M} \backslash \{l\}$
17: **end while**
18: **return** (K_0^*)

PROPOSITION 4. *For a given partition $P \in \mathscr{P}([0,\tau])$, let $K_0^*(P)$ be the set generated by Algorithm 1. Then,*

$$K_0^*(P) \subseteq Viab_{[0,\tau]}(\mathcal{K}). \tag{24}$$

PROOF. Let $\widetilde{K}_0(P)$ denote the final set constructed recursively by (10). Also, for a fixed direction l, let $K_0^{(l)}(P)$ denote the set produced at the end of each outer loop in Algorithm 1. Notice that via (23), for every $l \in \mathcal{M}$, $K_0^{(l)}(P) \subseteq \widetilde{K}_0(P)$. Therefore, $\bigcup_{l \in \mathcal{M}} K_0^{(l)}(P) \subseteq \widetilde{K}_0(P)$. Thus, $K_0^*(P) = \bigcup_{l \in \mathcal{M}} K_0^{(l)}(P) \subseteq Viab_{[0,\tau]}(\mathcal{K})$. $\square$

Remark 2. A similar argument holds for the discrete-time case in Algorithm 2, i.e. $K_0^* \subseteq Viab_{[0,\tau]}(\mathcal{K})$.

4.1.1 $\bigcirc(\cdot)$: *Computing the Maximum Volume Inscribed Ellipsoid*

Notice that in the continuous-time case, the sets $\mathcal{Y} := K_{|P|}(P)$ and $\mathcal{G} := Reach_{t_k - t_{k-1}}^{\sharp(l)}(K_k(P))$ are compact ellipsoids for every $l \in \mathcal{M}$, $P \in \mathscr{P}([0,\tau])$, and $k \in \{1, \ldots, |P|\}$. Similarly in the discrete-time case, $\mathcal{Y} := K_n$ and $\mathcal{G} := Reach_1^{\sharp(l)}(K_k)$ are compact ellipsoids for every $l \in \mathcal{M}$ and $k \in \{1, \ldots, n\}$. Their intersection is, in general, not an ellipsoid but can be easily under-approximated by one. The operation $\bigcirc(\cdot)$ under-approximates this intersection by computing the maximum volume inscribed ellipsoid in $\mathcal{Y} \cap \mathcal{G}$. The result is an ellipsoid that, while aiming to minimize the accuracy loss, can be used directly as the target set for the reachability computation in the subsequent time step.

Let us re-write the general ellipsoid as $\mathcal{E}(q, Q) = \{Hx + q \mid \|x\|_2 \leq 1\}$ with $H = Q^{\frac{1}{2}}$. Assume $\mathcal{Y} \cap \mathcal{G} \neq \emptyset$ and suppose $\mathcal{Y} = \mathcal{E}(q_1, Q_1)$ and $\mathcal{G} = \mathcal{E}(q_2, Q_2)$. Following [4], the computation of the maximum volume inscribed ellipsoid in $\mathcal{Y} \cap \mathcal{G}$ (a readily-available feature in ET) can be cast as a convex semidefinite program (SDP):

$$\underset{H \in \mathbb{R}^{n \times n}, q \in \mathbb{R}^n, \lambda_i \in \mathbb{R}}{\text{minimize}} \quad \log \det H^{-1} \tag{25a}$$

$$\text{subject to} \quad \begin{bmatrix} 1 - \lambda_i & 0 & (q - q_i)^T \\ 0 & \lambda_i I & H \\ q - q_i & H & Q_i \end{bmatrix} \succeq 0 \tag{25b}$$

$$\lambda_i > 0, \quad i = 1, 2. \tag{25c}$$

Using the optimal values for H and q, we will have $\bigcirc(\mathcal{Y} \cap \mathcal{G}) = \mathcal{E}(q, H^T H)$.

4.1.2   *Loss of Accuracy*

A set generated by Algorithms 1 or 2 could be an inaccurate approximation of $Viab_{[0,\tau]}(\mathcal{K})$, especially for large time horizons. The loss of accuracy is mainly attributed to the function $\bigcirc(\cdot)$, the under-approximation of the intersection at every iteration with its maximum volume inscribed ellipsoid. This approximation error propagates through the algorithms making them subject to the "wrapping effect".

In the continuous-time case, the quality of approximation is also affected by the choice of time interval partition (Proposition 2). Choosing a finer partition increases the quality of approximation. However, doing so would also require a larger number of intersections to be performed in the intermediate steps of the recursion. As such, one would expect that the error generated by $\bigcirc(\cdot)$ would be amplified. Luckily, since with a finer partition the reachable sets change very little from one time step to the next, the intersection error at every iteration becomes smaller. The end result is a smaller accumulative error and therefore a better approximation.

We show this using a trivial example: Consider the double integrator

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \tag{26}$$

subject to ellipsoidal constraints $u(t) \in \mathcal{U} := [-0.25, 0.25]$ and $x(t) \in \mathcal{K} := \mathcal{E}(\mathbf{0}, \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix}), \forall t \in [0,1]$. We employ eight different partitions P of the time interval such that we have $|P| = 13, 21, 34, 55, 89, 144, 233, 377$, all with equi-length sub-time intervals. The linear vector field is bounded on $\mathcal{K}$ in the infinity norm by $M = \|\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\| \sup_{x \in \mathcal{K}} \|x\| + \|\begin{bmatrix} 0 \\ 1 \end{bmatrix}\| \sup_{u \in \mathcal{U}} \|u\| = 0.75$. Thus, in Algorithm 1, $K_{|P|}(P) = \mathcal{K} \ominus \mathcal{B}(0.75 \times \|P\|)$. A piecewise ellipsoidal under-approximation of $Viab_{[0,1]}(\mathcal{K})$ for every partition P (with $|\mathcal{M}| = 10$ randomly chosen initial directions) is shown in Figure 2. Notice that as $|P|$ increases, the fidelity of approxi-
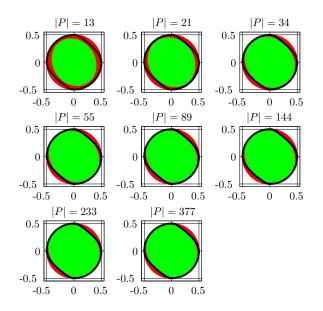
Figure 2: For the set $\mathcal{K}$ (red), $K_0(P)$ (green) under-approximates $Viab_{[0,1]}(\mathcal{K})$ (outlined in thick black lines via [30]) using Algorithm 1 under the double integrator dynamics. A finer time interval partition results in better approximation.

mation improves. A plot of the error in the accuracy of the under-approximation as a function of $|P|$ is provided in Figure 3.

5. PRACTICAL EXAMPLES

All computations are performed on a dual core Intel-based computer with 2.8 GHz CPU, 6 MB of L2 cache and 3 GB of RAM running single-threaded 32-bit MATLAB 7.5.

5.1 Flight Envelope Protection (Continuous-Time)

Consider the longitudinal aircraft dynamics $\dot{x}(t) = Ax(t) + B\delta_e(t)$,

$$A = \begin{bmatrix} -0.003 & 0.039 & 0 & -0.322 \\ -0.065 & -0.319 & 7.740 & 0 \\ 0.020 & -0.101 & -0.429 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.010 \\ -0.180 \\ -1.160 \\ 0 \end{bmatrix}$$

with state $x = [u, v, \dot{\theta}, \theta]^T \in \mathbb{R}^4$ comprised of deviations in aircraft velocity [ft/s] along and perpendicular to body axis, pitch-rate [crad/s], and pitch angle [crad] respectively[1], and with input $\delta_e \in [-13.3°, 13.3°] \subseteq \mathbb{R}$ the elevator deflection. These matrices represent stability derivatives of a Boeing 747 cruising at an altitude of 40 kft with speed 774 ft/s [5]. The state constraint set

$$\mathcal{K} = \mathcal{E}\left(\begin{bmatrix} 0 \\ 0 \\ 2.18 \\ 0 \end{bmatrix}, \begin{bmatrix} 1075.84 & 0 & 0 & 0 \\ 0 & 67.24 & 0 & 0 \\ 0 & 0 & 42.7716 & 0 \\ 0 & 0 & 0 & 76.0384 \end{bmatrix}\right)$$

represents the flight envelope. We require $x(t) \in \mathcal{K}$, $\forall t \in [0, 2]$.

A partition P is chosen such that $|P| = 400$ with equi-length sub-time intervals. Algorithm 1 (with $|\mathcal{M}| = 8$) computes via ET a piecewise ellipsoidal under-approximation of the viability kernel $Viab_{[0,2]}(\mathcal{K})$ as shown in Figures 4 and 5. Note that for any state belonging to this set, there exists an input that can protect the flight
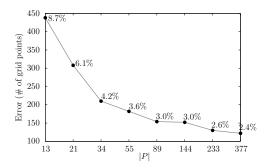
---

[1] crad $= 0.01$ rad $\approx 0.57°$.

Figure 3: Convergence plot of the error as a function of $|P|$ for the double-integrator example. Error is quantified as the fraction of grid points (total of 71×71) contained in the set difference between the level-set approximation of the viability kernel and its piecewise ellipsoidal under-approximation.

envelope over the specified time horizon. The overall computation time was roughly 10 mins. In comparison, the level-set approximation of the viability kernel (also shown in Figure 5) is computed in 5.4 hrs with significantly larger memory footprint over a grid with 45 nodes in each dimension using the Level-Set Toolbox [30]. Since the computed sets are 4D, we plot a series of 3D and 2D projections of these 4D objects.

5.2 Safety in Anesthesia Automation (Discrete-Time)

To improve patient recovery, lessen anesthetic drug usage, and reduce time spent at drug saturation levels, a variety of approaches to controlling depth of anesthesia have been proposed e.g. in [37, 14, 39, 8, 34, 28].

Over the past few years, an interdisciplinary team of researchers at the University of British Columbia has been developing an automated drug delivery system for anesthesia. As part of this effort, an open-loop bolus-based neuromuscular blockade system was developed and clinically validated in [10]. Discrete-time Laguerre-based LTI models of the dynamic response to rocuronium were identified using data collected from more than 80 patients via clinical trials. To obtain regulatory certificates to fully close the loop while employing an infusion-based administration of the drugs, mathematical guarantees of safety and performance of the system are likely to be required. The viability kernel and the continual reachability set [17], respectively, can provide such guarantees. (We note that while both this paper and [17] use maximal reachable sets for computation, the method in [17] computes the continual reachability set while here we compute the viability kernel.)

Consider the problem of computing the viability kernel for a constrained discrete-time LTI system (sampled every 20 s) that describes the pharmacological response of a patient under anesthesia. The therapeutic target is defined in the output space (as opposed to the state space) and the output signal should track a reference setpoint. As in [17], to perform the desired analysis we reformulate the problem by projecting the output bounds onto the state space while making the control action regulatory. As such, the original dynamics are augmented and transformed into an appropriate coordinate system of dimension seven. In this new state space, the first state z_1 represents the drug pseudo-occupancy (a metric related to the patient's plasma concentration of the anesthetic [10]) minus its setpoint value of 0.9 units, the next five states are the second to sixth Laguerre states transformed from the original coordinates, and the last state z_7 is a constant corresponding to the pseudo-
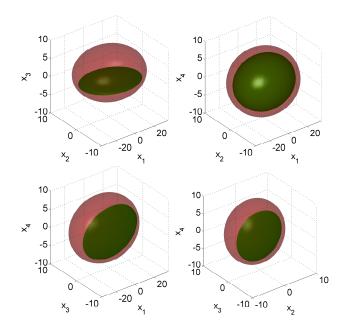
Figure 4: 3D projections of the under-approximation of $Viab_{[0,2]}(\mathcal{K})$ **for Example 5.1. The flight envelope** $\mathcal{K}$ **is the red transparent region. The green piecewise ellipsoidal sets under-approximate the viability kernel.**

Figure 5: 2D projections of the under-approximation of $Viab_{[0,2]}(\mathcal{K})$ **for Example 5.1. The constraint set** $\mathcal{K}$ **(red) and a piecewise ellipsoidal under-approximation of the viability kernel (green) are shown. The level-set approximation of the viability kernel, computed via [30], is outlined in thick black lines.**

occupancy setpoint. The states are assumed to be constrained by a slab in $\mathbb{R}^7$ that is only bounded in the z_1 direction. Note that with this formulation, the last state z_7 is allowed to take on values that are not needed; of actual interest is the behavior of the remaining states when z_7 equals the pseudo-occupancy setpoint. The input constraint, which represents the actuator's physical limitations (i.e. hard bounds on rocuronium infusion rate), is a closed and bounded interval in $\mathbb{R}$.

The input constraint set is a one-dimensional ellipsoid. To under-approximate the state constraint with a non-degenerate ellipsoid we use *a priori* knowledge about the typical values of the (Laguerre) states $z_2, \ldots, z_6$ and bound them by an ellipsoid with a large spectral radius of $\lambda_{\max} = 30$ in those directions. (This imposed constraint can be further relaxed if necessary.) Guaranteeing that this ellipsoidal target set $\mathcal{K}$, which is our desired clinical effect, is not violated during the surgery provides a certificate of safety of the closed-loop system. Therefore, for a 30 min long surgery for instance, we require $z(t) \in \mathcal{K}, \forall t \in [0, 90]$ despite bounded input authority. Using appropriately synthesized infusion policies, the states belonging to the viability kernel of $\mathcal{K}$ under the extended system will never leave the desired clinical effect for the duration of the surgery.

We under-approximate $Viab_{[0,90]}(\mathcal{K})$ in 986 s using Algorithm 2 with $|\mathcal{M}| = 30$. Of the 30 randomly chosen initial directions used in the ellipsoidal computations, 15 resulted in nonempty ellipsoids that make up the piecewise ellipsoidal under-approximation of the viability kernel (Figure 6). Note that no similar computations are currently possible in such high dimensions using Eulerian methods directly.

6. CONCLUSIONS AND FUTURE WORK

We presented a connection between the viability kernel (and by duality, the minimal reachable tube) and the maximal reachable sets of possibly nonli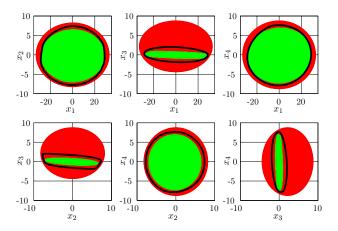near systems. Owing to this connection, the efficient and scalable Lagrangian techniques can be used to approximate the viability kernel. Motivated by a high-dimensional problem of guaranteed safety in control of anesthesia, we proposed a scalable algorithm that computes a piecewise ellipsoidal under-approximation of the viability kernel for LTI systems based on ellipsoidal techniques for reachability.

Empirically quantifying the computational complexity of the piecewise ellipsoidal algorithm is a work under way for which we expect a polynomial complexity in the order of $|\mathcal{M}||P| \left(\mathcal{O}(\mathfrak{R}_\delta) + \mathcal{O}(\mathfrak{S}) \right)$ where $\mathcal{O}(\mathfrak{R}_\delta)$ is the complexity of computing the maximal reachable set along a given direction over the time interval δ and $\mathcal{O}(\mathfrak{S})$ is the complexity of solving the SDP (25).

While the presented algorithm has shown to be effective and efficient, it may be subject to excessive conservatism particularly for large time horizons. We are currently developing alternative approaches that yield a more accurate under-approximation of the viability kernel while still preserving the scalability property.

Finally, the presented connection between the viability kernel and the maximal reachable sets paves the way to synthesizing "safety-preserving" optimal control laws in a more efficient and scalable manner.

7. REFERENCES

[1] J.-P. Aubin. *Viability Theory*. Systems and Control: Foundations and Applications. Birkhäuser, Boston, MA, 1991.

[2] A. M. Bayen, I. M. Mitchell, M. Oishi, and C. J. Tomlin. Aircraft autolander safety analysis through optimal control-based reach set computation. *Journal of Guidance, Control, and Dynamics*, 30(1):68–77, 2007.

[3] F. Blanchini and S. Miani. *Set-Theoretic Methods in Control*. Springer, 2008.

[4] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[5] A. E. Bryson. *Control of Spacecraft and Aircraft*. Princeton Univ. Press, 1994.

[6] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Set-valued numerical analysis for optimal control and differential games. In M. Bardi, T. Raghavan, and
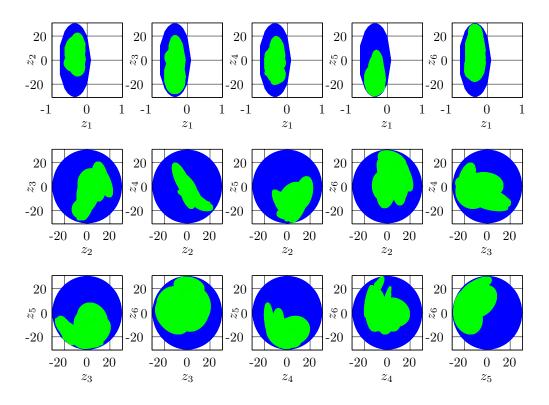
Figure 6: 2D projections of the under-approximation of $Viab_{[0,90]}(\mathcal{K})$ for Example 5.2 for the first six states when z_7 equals the setpoint value. The constraint set $\mathcal{K}$ (blue) and a piecewise ellipsoidal under-approximation of the provably safe regions (green) are shown.

T. Parthasarathy, editors, *Stochastic and Differential Games: Theory and Numerical Methods*, number 4 in Annals of the International Society of Dynamic Games, pages 177–247, Boston, MA, 1999. Birkhäuser.

[7] A. N. Daryin, A. B. Kurzhanski, and I. V. Vostrikov. Reachability approaches and ellipsoidal techniques for closed-loop control of oscillating systems under uncertainty. In *Proc. IEEE Conference on Decision and Control*, pages 6385–6390, San Diego, CA, 2006.

[8] G. Dumont, A. Martinez, and J. Ansermino. Robust control of depth of anesthesia. *International Journal of Adaptive Control and Signal Processing*, 23:435–454, 2009.

[9] Y. Gao, J. Lygeros, and M. Quincampoix. The reachability problem for uncertain hybrid systems revisited: a viability theory perspective. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control, LNCS 3927*, pages 242–256, Berlin Heidelberg, 2006. Springer-Verlag.

[10] T. Gilhuly. *Modeling and control of neuromuscular blockade*. PhD thesis, University of British Columbia, Vancouver, Canada, 2007.

[11] A. Girard and C. Le Guernic. Efficient reachability analysis for linear systems using support functions. In *IFAC World Congress*, Seoul, Korea, July 2008.

[12] A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control, LNCS 3927*, pages 257–271. Springer-Verlag, 2006.

[13] Z. Han and B. H. Krogh. Reachability analysis of nonlinear

systems using trajectory piecewise linearized models. In *Proc. American Control Conference*, pages 1505–1510, Minneapolis, MN, 2006.

[14] C. Ionescu, R. De Keyser, B. Torrico, T. De Smet, M. Struys, and J. Normey-Rico. Robust predictive control strategy applied for propofol dosing using BIS as a controlled variable during anesthesia. *IEEE Transactions on Biomedical Engineering*, 55(9):2161–2170, 2008.

[15] S. Kaynama and M. Oishi. Complexity reduction through a Schur-based decomposition for reachability analysis of linear time-invariant systems. *International Journal of Control*, 84(1):165–179, 2011.

[16] S. Kaynama and M. Oishi. A modified Riccati transformation for complexity reduction in reachability analysis of linear time-invariant systems. *IEEE Transactions on Automatic Control*, 2011. (accepted; preprint available at www.ece.ubc.ca/∼kaynama).

[17] S. Kaynama, M. Oishi, I. M. Mitchell, and G. A. Dumont. The continual reachability set and its computation using maximal reachability techniques. In *Proc. IEEE Conference on Decision and Control, and European Control Conference*, pages 6110–6115, Orlando, FL, 2011.

[18] A. B. Kurzhanski and I. Vályi. *Ellipsoidal Calculus for Estimation and Control*. Birkhäuser, Boston, MA, 1996.

[19] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In N. Lynch and B. Krogh, editors, *Hybrid Systems: Computation and Control, LNCS 1790*, pages 202–214, Berlin Heidelberg, 2000. Springer-Verlag.

[20] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for

reachability analysis: internal approximation. *Systems & Control Letters*, 41:201–211, 2000.

[21] A. B. Kurzhanski and P. Varaiya. On reachability under uncertainty. *SIAM Journal on Control and Optimization*, 41(1):181–216, 2002.

[22] A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal Toolbox (ET). In *Proc. IEEE Conference on Decision and Control*, pages 1498–1503, San Diego, CA, Dec. 2006.

[23] A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 52(1):26–38, 2007.

[24] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari. Multi-Parametric Toolbox (MPT). In R. Alur and G. J. Pappas, editors, *Hybrid Systems: Computation and Control, LNCS 2993*, pages 448–462, Berlin, Germany, 2004. Springer.

[25] C. Le Guernic. *Reachability analysis of hybrid systems with linear continuous dynamics*. PhD thesis, Université Grenoble 1 – Joseph Fourier, 2009.

[26] J. Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6):917–927, June 2004.

[27] K. Margellos and J. Lygeros. Air traffic management with target windows: An approach using reachability. In *Proc. IEEE Conference on Decision and Control*, pages 145–150, Shanghai, China, Dec 2009.

[28] T. Mendonca, J. Lemos, H. Magalhaes, P. Rocha, and S. Esteves. Drug delivery for neuromuscular blockade with supervised multimodel adaptive control. *IEEE Transactions on Control Systems Technology*, 17(6):1237–1244, November 2009.

[29] I. M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Hybrid Systems: Computation and Control, LNCS 4416*, pages 428–443, Berlin Heidelberg, 2007. Springer-Verlag.

[30] I. M. Mitchell. A toolbox of level set methods. Technical report, UBC Department of Computer Science, TR-2007-11, June 2007.

[31] I. M. Mitchell. Scalable calculation of reach sets and tubes for nonlinear systems with terminal integrators: a mixed implicit explicit formulation. In *Proc. Hybrid Systems: Computation and Control*, pages 103–112, Chicago, IL, 2011. ACM.

[32] 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*, 50(7):947–957, July 2005.

[33] I. M. Mitchell and C. J. Tomlin. Overapproximating reachable sets by Hamilton-Jacobi projections. *Journal of Scientific Computing*, 19(1–3):323–346, 2003.

[34] P. Oliveira, J. P. Hespanha, J. M. Lemos, and T. Mendonça. Supervised multi-model adaptive control of neuromuscular blockade with off-set compensation. In *Proc. European Control Conference*, 2009.

[35] D. Panagou, K. Margellos, S. Summers, J. Lygeros, and K. J. Kyriakopoulos. A viability approach for the stabilization of an underactuated underwater vehicle in the presence of current disturbances. In *Proc. IEEE Conference on Decision and Control*, pages 8612–8617, Dec. 2009.

[36] P. Saint-Pierre. Approximation of the viability kernel.

Applied Mathematics and Optimization, 29(2):187–209, Mar 1994.

[37] O. Simanski, A. Schubert, R. Kaehler, M. Janda, J. Bajorat, R. Hofmockel, and B. Lampe. Automatic drug delivery in anesthesia: From the beginning until now. In *Proc. Mediterranean Conf. Contr. Automation*, Athens, Greece, 2007.

[38] D. M. Stipanović, I. Hwang, and C. J. Tomlin. Computation of an over-approximation of the backward reachable set using subsystem level set functions. In *Proc. IEE European Control Conference*, Cambridge, UK, Sept. 2003.

[39] S. Syafiie, J. Niño, C. Ionescu, and R. De Keyser. NMPC for propofol drug dosing during anesthesia induction. In *Nonlinear Model Predictive Control*, volume 384, pages 501–509. Springer Berlin Heidelberg, 2009.