Lagrangian methods for approximating the viability kernel in high-dimensional systems[☆]

John N. Maidens [a,1,2], Shahab Kaynama [a,2], Ian M. Mitchell [b], Meeko M.K. Oishi [c], Guy A. Dumont [a]

[a] Electrical and Computer Engineering, University of British Columbia, 2332 Main Mall, Vancouver, BC, V6T 1Z4, Canada
[b] Computer Science, University of British Columbia, 2366 Main Mall, Vancouver, BC, V6T 1Z4, Canada
[c] Electrical and Computer Engineering, University of New Mexico, MSC01 1100, 1 University of New Mexico, Albuquerque, NM 87131, USA

ARTICLE INFO

ABSTRACT

While a number of Lagrangian algorithms to approximate reachability in dozens or even hundreds of dimensions for systems with linear dynamics have recently appeared in the literature, no similarly scalable algorithms for approximating viable sets have been developed. In this paper we describe a connection between reachability and viability that enables us to compute the viability kernel using reach sets. This connection applies to any type of system, such as those with nonlinear dynamics and/or non-convex state constraints; however, here we take advantage of it to construct three viability kernel approximation algorithms for linear systems with convex input and state constraint sets. We compare the performance of the three algorithms and demonstrate that the two based on highly scalable Lagrangian reachability – those using ellipsoidal and support vector set representations – are able to compute the viability kernel for linear systems of larger state dimension than was previously feasible using traditional Eulerian methods. Our results are illustrated on a 6-dimensional pharmacokinetic model and a 20-dimensional model of heat conduction on a lattice.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Viability theory plays an important role in safety verification for control systems (cf. Aubin, Bayen, & Saint-Pierre, 2011), a particularly important problem for high risk, expensive, or safety-critical applications. In many engineered systems, input constraints limit the system's ability to remain within a desired "safe" region of operation. Consider, for example, problems in aerodynamic flight envelope protection (Tomlin, Mitchell, Bayen, & Oishi, 2003) or underwater vehicle operation under constraints (Panagou, Margellos, Summers, Lygeros, & Kyriakopoulos, 2009). For such systems, constraints on the state space determine the "safe set". However, because the control authority for the system is also constrained, there are some configurations in the safe set for which the state may inevitably exit. Hence it is important to identify the subset of the safe set for which the existence of a control input that keeps the state of the system within the safe region can be guaranteed.

This subset, known as the viability kernel (or the controlled invariant set), takes into account the system's dynamics and bounded control authority. For a constraint set K, the viability kernel $Viab(K)$ is the subset of K for which a control input exists that keeps the state of the system within K for the duration of a known (possibly infinite) time horizon.

The viability kernel has traditionally been approximated using Eulerian methods such as the Viability Kernel Algorithm (Saint-Pierre, 1994) and level set approaches (Mitchell, Bayen, & Tomlin, 2005). However, Eulerian methods require gridding the state space and hence their time and memory complexity grow exponentially with the state dimension. In practice, this approach is infeasible for systems with more than 3 or 4 states. Lagrangian methods have been applied previously to the computation of viability kernels, for example in Blanchini and Miani (2008), but the implementation has relied on polyhedral set representations that also do not scale well with the number of states.

There has been recent work to compute the viability kernel for high-dimensional systems based on simulated annealing (Bonneuil, 2006), approximate dynamic programming (Coquelin, Martin, & Munos, 2007) and supervised classification (Deffuant, Chapel, & Martin, 2007). The simulated annealing method has been demonstrated for a chain of integrators in 10 dimensions, taking 22 min to compute each point on viability kernel's boundary. The dynamic programming method has been demonstrated on a 4-dimensional system, taking 163 s to compute a grid of 2×10^5 points. The supervised classification method has been demonstrated on an ecological model with 51 inputs and 6 states (Chapel, Deffuant, Martin, & Mullon, 2008) but still relies on a gridding of the state space; hence its applicability to systems with a large number of *states* is limited. Our results in Section 3.3.2 show a substantial improvement over existing methods in terms of scalability in the state dimension.

Lagrangian methods have been applied successfully to the computation of reachable sets (Chutinan & Krogh, 2003; Kurzhanski & Varaiya, 2000a; Le Guernic & Girard, 2010). In contrast to Eulerian methods, Lagrangian methods use representations that follow the vector field's flow. Since Lagrangian methods do not depend on gridding the state space, it is computationally feasible to analyse high-dimensional systems.

In Section 2, we present a connection between the viability kernel and reachable sets that allows the large class of methods developed for reachability analysis to be applied to the computation of viability kernels. It can be used for any system and set representation which supports the backward maximal reach set and intersection operations (or under-approximations thereof), in theory including nonlinear dynamics and/or non-convex constraints.

In Section 3, we restrict our attention to discrete-time linear systems under convex input and state constraints, a case for which a wealth of efficient Lagrangian reachability techniques exist. We use the results from Section 2 to provide three examples of Lagrangian algorithms for computing the viability kernel, and we compare these three algorithms. The polytope method performs well in terms of accuracy but does not scale well as the state dimension grows, becoming infeasible in greater than four dimensions. In comparison, the time complexity of the ellipsoidal method increases more slowly with the state dimension, but its accuracy is limited. The support vector method strikes a balance between scalability and accuracy. It allows the user to choose a desired accuracy in terms of the number of points on the boundary of the viability kernel that they wish to evaluate. We demonstrate empirically that the runtime of the ellipsoidal and support vector methods appear to be polynomial in state dimension.

While the three algorithms presented in Section 3 apply only to discrete-time systems, the techniques developed in this paper can equally be applied to continuous-time systems, provided that we have a method of computing (or under-approximating) continuous-time reach sets. As an example, in the conference paper (Kaynama, Maidens, Mitchell, Oishi, & Dumont, 2012) we use the continuous-time techniques developed in Section 2.2.2 to under-approximate the viability kernel of a continuous-time system using ellipsoidal techniques.

In Section 4, we provide two applications of our results. We compute the viability kernel for a 6-dimensional discrete-time model of Propofol pharmacokinetics in children, and a 20-dimensional discretized heat equation.

2. Establishing connections between viability and reachability

There is a close relationship between viability theory (Aubin et al., 2011) and constrained reachability (Kurzhanski & Varaiya, 2001). Both frameworks study the evolution of dynamic systems under input and/or state constraints. The relationship between the two theories is often discussed in the context of optimal control theory by formulating both reachability and viability problems in terms of Hamilton–Jacobi equations, for example (Lygeros, 2004).

The Hamilton–Jacobi approach has proven extremely successful in the analysis of low-dimensional systems. Level set methods can be used to approximate the viscosity solution of the Hamilton–Jacobi PDE corresponding to a given viability or reachability problem (Tomlin et al., 2003). Tools are available for computing viable and reachable sets numerically (Mitchell & Templeton, 2005) but they scale poorly with state dimension.

The recent emergence of accurate and scalable methods and tools for approximating reachable sets in high-dimensional systems (Frehse et al., 2011; Kurzhanski & Varaiya, 2000a) has inspired us to attempt to find analogous methods for the approximation of viability kernels. In this section, we expose a connection between viability theory and reachability theory. The results presented here appeared in a preliminary form in a conference paper (Kaynama et al., 2012).

2.1. Preliminaries

We are concerned with analysing systems of the form

$$\begin{cases} \mathcal{L}(x(t)) = f(x(t), u(t)) \\ u(t) \in \mathcal{U} \end{cases} \tag{1}$$

where the time t ranges throughout a *time domain* $\mathbb{T}$. The time domain $\mathbb{T}$ can be either *continuous* ($\mathbb{T} = [0, \tau] \subseteq \mathbb{R}_+$) or *discrete* ($\mathbb{T} = [0, \tau] \cap \mathbb{Z}_+$). If $0 < \tau < \infty$ this problem is said to have a *finite horizon*; otherwise, if $\tau = \infty$, it is said to have an *infinite horizon*. $\mathcal{L}$ is the differential operator corresponding to the given time domain (differentiation in the case of a continuous-time system and differencing in the case of a discrete-time system). The system's *state x* ranges over the finite-dimensional vector space $\mathbb{R}^d$ and the system's *input* is constrained to a nonempty, compact, convex subset $\mathcal{U} \subseteq \mathbb{R}^m$.[3] When (1) evolves under continuous time, we assume that the function $f : \mathbb{R}^d \times \mathcal{U} \to \mathbb{R}^d$ is sufficiently smooth to guarantee the existence and uniqueness of solutions to the corresponding initial value problem.

Viability theory is concerned with ensuring that a system's state x remains within a set of *viability constraints* $K \subseteq \mathbb{R}^d$. Any trajectory of system (1) that leaves the set K at some point in time is considered to be no longer *viable*.

We call a set S viable under K if for every initial state $x_0 \in S$ there exists some measurable input $u_0 : \mathbb{T} \to \mathcal{U}$ such that the solution $x(\cdot)$ to the initial value problem

$$\begin{cases} \mathcal{L}(x(t)) = f(x(t), u_0(t)) \\ x(0) = x_0 \end{cases} \tag{2}$$

satisfies $x(t) \in K$ for all $t \in \mathbb{T}$.

The *viability kernel* of a set of viability constraints K is the largest viable set contained in K. Equivalently, the viability kernel is defined as follows:

$$Viab_{\mathbb{T}}(K) = \{x_0 \in K \mid \exists u_0 : \mathbb{T} \to \mathcal{U} \; \forall t \in \mathbb{T} \; x(t) \in K\}.$$

The related constructs of constrained reachability analysis are a popular technique for formal safety verification (for example Mitchell, 2007). They provide a method of simulating all possible trajectories of a dynamic system under all admissible inputs. Essentially, they are concerned with determining if any trajectories of the system (1) that begin in a set of initial conditions I can reach a set of terminal states T.

---

[3] This is not the most general context in which viability theory can be developed. Aubin (1991) allows the constraint set $\mathcal{U}$ to depend on the state x.