A Summary of Recent Progress on Efficient Parametric Approximations of Viability and Discriminating Kernels*

Ian M. Mitchell[†]

Department of Computer Science
University of British Columbia
mitchell@cs.ubc.ca
http://www.cs.ubc.ca/~mitchell

Abstract

Viability and discriminating kernels are powerful constructs for analyzing system safety through model checking, but until recently the only computational algorithms available were nonparametric grid-based approaches which, although accurate, scaled exponentially with the dimension of the system's state space. In contrast, several polynomial complexity reachability algorithms have been developed using various parametric set representations. In a recent series of papers, two of these parametric approaches—based on ellipsoids and support vectors—have been adapted to approximate viability and discriminating kernels in the discrete, continuous and sampled data models of time. This paper briefly summarizes these algorithms and compares their key features with one another and with a representative nonparametric approach.

1 Introduction

Viability theory [1] provides an alternative approach to reachability for model checking safety. For a system with trajectories $x(\cdot) : t \to \mathbb{R}^{d_x}$ and input signal $u(\cdot) : t \to \mathcal{U} \subset \mathbb{R}^{d_u}$, the *viability kernel* for constraint $\mathcal{K} \subset \mathbb{R}^{d_x}$ over horizon $T \in [0, \infty)$ can be written

$$\mathsf{Viab}\left([0,T],\mathcal{K}\right) \triangleq \{x_0 \in \mathbb{R}^{d_x} \mid \exists u(\cdot), \forall t \in [0,T], x(t) \in \mathcal{K}\}, \tag{1}$$

In words, the viability kernel is the set of initial conditions for which there exist control signals which will keep the resulting trajectories inside the constraint set over the desired horizon. The dual construct is the *invariance kernel*

$$\mathsf{Inv}\left([0,T],\mathcal{K}\right) \triangleq \{x_0 \in \mathbb{R}^{d_x} \mid \forall u(\cdot), \forall t \in [0,T], x(t) \in \mathcal{K}\}. \tag{2}$$

which is the set of initial conditions for which all control signals will lead to trajectories which remain within the constraint set over the horizon. If we allow for a second input signal $v(\cdot) : t \to \mathcal{V} \subset R^{d_v}$ acting adversarially to the first, we can define the *discriminating kernel*

$$\mathsf{Disc}\left([0,T],\mathcal{K}\right) \triangleq \{x_0 \in \mathbb{R}^{d_x} \mid \exists u(\cdot), \forall v(\cdot), \forall t \in [0,T], x(t) \in \mathcal{K}\}. \tag{3}$$

which is the set of initial conditions for which there exist input signals $u(\cdot)$ which will keep the resulting trajectories within the constraint despite the action of any input signals $v(\cdot)$.

[*] Except where otherwise noted, contents ©2015 by the authors and released under a Creative Commons Attribution 4.0 International License.

[†] Research supported by the National Science and Engineering Council of Canada (NSERC) Discovery Grant #298211.

For model checking purposes we typically choose the safe states of the system as the constraint set, in which case $u(\cdot)$ in (1) or (3) corresponds to a control input seeking to keep the system safe. The viability kernel then represents the set of states which can be kept safe, and is sometimes called the control invariant set of $\mathcal{K}$, although we will stick with the viability terminology to avoid potential confusion with the regular invariant set (2). In this context the discriminating kernel's other input $v(\cdot)$ is often called the disturbance. It allows for a worst case analysis in the face of bounded model error and so the discriminating kernel could be considered a robust version of the viability kernel.

Algorithms for (backward) reach tubes can be used to find these kernels. We use the terms *reach sets* and *reach tubes* to denote reachability at a single instant of time and over an interval of time respectively. Following [15] we define several possible such sets and tubes depending on what the input is doing:

$$\mathsf{Reach}_{+}\left(t, \mathcal{S}\right) \triangleq \{x_0 \mid \exists u(\cdot), x(t) \in \mathcal{S}\}, \tag{4}$$

$$\mathsf{Reach}_{+}\left([t_0, t_f], \mathcal{S}\right) \triangleq \{x_0 \mid \exists u(\cdot), \forall t \in [t_0, t_f], x(t) \in \mathcal{S}\}, \tag{5}$$

$$\mathsf{Reach}_{-}\left(t, \mathcal{S}\right) \triangleq \{x_0 \mid \forall u(\cdot), x(t) \in \mathcal{S}\}, \tag{6}$$

$$\mathsf{Reach}_{-}\left([t_0, t_f], \mathcal{S}\right) \triangleq \{x_0 \mid \forall u(\cdot), \forall t \in [t_0, t_f], x(t) \in \mathcal{S}\}. \tag{7}$$

We call the first two *maximum* because the selection of input makes the set or tube as large as possible, while we call the second two *minimum*. Assuming that we can complement sets, the two reach tubes can be used to find the viability and invariance kernels:

$$\mathsf{Viab}\left([0, T], \mathcal{K}\right) = \left(\mathsf{Reach}_{-}\left([0, T], \mathcal{K}^{\complement}\right)\right)^{\complement}, \tag{8}$$

$$\mathsf{Inv}\left([0, T], \mathcal{K}\right) = \left(\mathsf{Reach}_{+}\left([0, T], \mathcal{K}^{\complement}\right)\right)^{\complement}. \tag{9}$$

It is the maximum reach set (4) and tube (5) which are computed by most published algorithms, taking advantage of the fact that the tube (5) is just the union over time of the sets (4) at the individual times, and the sets can be efficiently approximated using parametric set representations. Unfortunately, it is shown in [15] that the minimal tube (7) is not the union of the corresponding sets (6). Consequently, although the relationship (9) may point toward a convenient algorithm for approximating the invariance kernel through efficient maximal reach set calculations, it is not so easy to repurpose minimal reach sets to approximate viability or discriminating kernels.

Instead, the viability and discriminating kernels have traditionally been approximated by approaches such as [19, 4]. Through the relationship to reachability, it is also easy to modify level set algorithms based on the Hamilton-Jacobi (HJ) partial differential equation (PDE) such as [16] to approximate these kernels. These schemes are all nonparametric grid-based approaches.[1] While they can accurately represent complex kernel shapes for systems with nonlinear dynamics, the discretization of the state space with a grid means that these schemes generally require computational time and space exponential in the system's state space dimension.

[1] Also called "Eulerian" in some of the citations. This terminology is based on the connection of level set algorithms to so-called Eulerian front capturing schemes in computational fluid dynamics (CFD). In the CFD community, a scheme is called Eulerian if it computes on a grid which does not move with the underlying flow of the fluid. The corresponding principle in viability would be a representation of the kernel which does not move with the dynamics, such as the grids with fixed node locations (although sometimes with adaptive resolution) used in [19, 4, 16].

Our goal in this paper is to briefly summarize recently developed formulations that approximate these kernels using reachability algorithms based on parametric set representations—ellipsoids and support vectors—which scale polynomially with state space dimension.[2] Table 1 in section 5 is the key contribution of this work, in that it summarizes the features of these parametric algorithms and compares them to the level set algorithm (as a representative of the traditional nonparametric approaches). The intervening sections of the paper discuss models of time, the kernel approximation algorithms in abstract, and a few details of the ellipsoidal and support vector algorithms.

2 Models of Time

Up until now we have purposefully avoided discussion of the model of time. Our justification is that generic algorithms for these kernels can be defined for at least three different treatments of time in the system dynamics. In the discussion below, we write the system dynamics for the more general case of two adversarial inputs used in the discriminating kernel, but they are easily modified for the simpler single input cases.

Discrete time (DT): System dynamics are given by $x(t+1) = f(x(t), u(t), v(t))$, and we assume full and instantaneous state feedback so that the input $u(t)$ may be chosen with knowledge of $x(t)$. For the two input case where the constraint is the safety set, any advantage of choosing last is given to the disturbance $v(t)$, which can choose its value knowing both $x(t)$ and $u(t)$.

Continuous time (CT): System dynamics are given by $\frac{d}{dt}x(t) = \dot{x}(t) = f(x(t), u(t), v(t))$, and we similarly assume full and instantaneous state feedback so that input $u(t)$ can be chosen with knowledge of $x(t)$. The CT case typically requires regularity assumptions on f and the inputs in order to ensure that solutions are well-posed. Common and sufficient assumptions are continuity of f with respect to its parameters (and Lipschitz continuity in x) as well as measurability of the inputs; however, weaker conditions have been described (for example, see [1, 2]).

Treatment of priority in the two input case is particularly delicate: the input at a disadvantage because it plays first can still instantaneously determine the choice of the second input because that choice has an immediate effect on the dynamics and is thereby known to the first input through state feedback. Forcing the advantaged player to use a “non-anticipative strategy” is the typical approach to rigorously address the mathematical niceties [5]. Further discussion of this knowledge pattern issue can be found in [16], but from an engineering perspective giving the disturbance $v(t)$ the non-anticipative strategy advantage during the analysis is simply ensuring a result which is technically a conservative bound on any physically realizable disturbance, and is in practice indistinguishable from a feedback disturbance except for artificially constructed examples.

A much larger threat to the validity of a CT analysis arises from the assumption that the inputs are measurable functions of time, because the optimal signals often turn out to be discontinuous; for example, in systems where scalar control inputs appear in an affine fashion

[2] Also called “Lagrangian” in some of the citations to distinguish from the Eulerian schemes. In CFD a scheme is called Lagrangian if it computes with a representation that moves with the underlying fluid flow. The corresponding principle in viability would be a representation of the kernel which moves with the dynamics, which the ellipsoidal representation does explicitly (through the motion of the center of the ellipsoid) and the support vector representation does implicitly.

(eg: $\dot{x} = f_1(x)+f_2(x)u$), the optimal feedback $u(x(t))$ will jump from one end of the input range to the other at potentially unpredictable locations in the state space (so-called "bang-bang" control). The impossibility of implementing such controls perfectly throws into doubt whether the resulting kernels could be maintained in practice, and hence motivated development of algorithms for the final model of time discussed next.

Sampled data (SD): The most common design pattern for cyber-physical systems involves a CT plant (the physical system) controlled by a DT controller (the cyber system). For analysis purposes, it is common to adopt either DT or CT for both; however, a fully DT analysis may miss safety violating behaviours of the plant which occur between time samples, while a fully CT analysis may require the controller to generate impossible to implement signals (as discussed above) to ensure safety. An SD analysis restricts the controller to generate feedback signals which can be implemented—in our case, piecewise constant signals through a zero-order hold on the DT output from the controller—while still taking into account the CT evolution of the plant between samples. When a disturbance input is present, it generally makes sense to give it the full flexibility of a measurable CT signal to ensure that the resulting analysis is robust to the uncertainty modeled by this input.

3 Abstract Algorithms

The abstract DT algorithm is a simple recursion [14]

$$\begin{aligned} \mathcal{K}_0 &= \mathcal{K}, \\ \mathcal{K}_{n+1} &= \mathcal{K}_0 \cap \mathsf{Reach}_+ \left(1, \mathcal{K}_n\right). \end{aligned} \tag{10}$$

This recursion is similar to the viability kernel algorithm from [19], except that we use a general (maximal) reach set for the dynamics instead of the differential inclusion itself. While the obvious recursion would intersect with $\mathcal{K}_n$ at each step rather than $\mathcal{K}_0$, it is shown in [14] that the latter is sufficient, so we use $\mathcal{K}_0$ because it is less affected by computational error than $\mathcal{K}_n$.

Like many algorithms for CT reach tubes, the CT viability algorithm is essentially the same as the DT algorithm except that the sets involved need to be appropriately modified to take into account the system dynamics between computational timesteps. To ensure that their result is appropriately conservative, the reach tube algorithms dilate (or "bloat") their approximations, whereas for conservativeness in approximating the viability kernel it is necessary to erode. Following [14] if M is a bound on the dynamics then a sufficient erosion is given by

$$\mathcal{K}_\downarrow = \{x \in \mathcal{K} \mid \operatorname{dist}(x, \mathcal{K}^{\complement}) \geq \rho M\}$$

where $\rho > 0$ is the algorithm's computational timestep. The recursion is then given by

$$\begin{aligned} \mathcal{K}_0 &= \mathcal{K}_\downarrow, \\ \mathcal{K}_{n+1} &= \mathcal{K}_0 \cap \mathsf{Reach}_+ \left(\rho, \mathcal{K}_n\right). \end{aligned} \tag{11}$$

The extension of both the DT and CT algorithms to discriminating kernels in [9] is conceptually straightforward, although the presence of the adversarial inputs results in a complicated notation.

Because the shape of these kernels depends on the CT dynamics of the plant, the abstract SD algorithm builds on the abstract CT algorithm but must ensure that the control signal remains constant between sample times. In order to do so, the CT algorithm is run in an augmented state space and with augmented dynamics:

$$\tilde{x} \triangleq \begin{bmatrix} x \\ u \end{bmatrix} \qquad \tilde{f}(\tilde{x}) \triangleq \begin{bmatrix} f(x,u) \\ 0 \end{bmatrix}$$

In the augmented state space the states representing the input u are held constant because the corresponding dynamics are zero. The first version of the algorithm [18] uses a recursion built on invariance kernels in the augmented state space (where the input for the invariance kernel is the disturbance, if any) plus tensor products and projections to move between the original and augmented state spaces. A newer algorithm [17] mixes invariance kernels and minimal reach sets to achieve improved accuracy and in some cases improved efficiency.

4 Technical Details

The abstract algorithms themselves are straightforward recursions; what makes them interesting is that they can be efficiently implemented or approximated using parametric set representations. At this point we restrict the dynamics to linear models

$$x(t+1) = \mathrm{A}x(t) + \mathrm{B}u(t) \qquad \text{or} \qquad \dot{x}(t) = \mathrm{A}x(t) + \mathrm{B}u(t).$$

Support functions provide a mechanism for parametric outer approximations of compact, convex sets, and have been used very successfully for reach sets and tubes of systems with such linear dynamics [6, 13]. When using viability and discriminating kernels for safety analysis, we generally require an underapproximation; fortunately, support vectors [12] provide a dual and inner approximation of compact, convex sets. It is possible to implement the key reachability and intersection steps of the DT algorithm (10) exactly using convex optimization to find the support vectors and thereby construct a polytopic underapproximation of the viability kernel [14] (as well as a free overapproximation based on the corresponding support functions). While highly accurate and scalable (see section 5), the algorithm has three shortcomings: it has not yet been adapted to continuous time, adaptation to discriminating kernels appears to require the solution of nonconvex optimizations, and the support vectors generated by the optimizations may become ill-posed if intermediate time viability kernels (including the original constraint set) do not have a smooth boundary.

As a class of parametric representations, ellipsoids might at first glance appear to be poorly suited for approximating the viability or discriminating kernels because even for linear dynamics they are not closed under the key operations of reachability and intersection needed in (10) or (11) (unlike support functions / vectors). However, it is possible to construct inner approximations of the necessary reach sets [10] and intersections [3], and thereby implement both the DT and CT algorithms [8]. For the extension to the SD algorithm, the required projection operation can be implemented exactly and (an underapproximation of) the tensor product can be determined through another convex optimization [17, 18].

5 Results

The ellipsoidal viability kernel algorithm is demonstrated on a four dimensional CT flight envelope protection problem and a seven dimensional DT anesthesia model in [8]. In [14] the accuracy and efficiency of the DT support vector and ellipsoidal viability kernel algorithms

	Level Set	Ellipsoidal	Support Vector
Dynamics	nonlinear	linear	linear
Time	CT / SD	CT / DT / SD	DT
Complexity	$\mathcal{O}(n^d)$	$\mathcal{O}(kd^3)$	$\mathcal{O}(kd^2)$
Control input	optimal (CT) sampled (SD)	optimal	optimal
Control synthesis	✓	✓	–
Discriminating kernel	optimal	optimal	–
Accuracy	excellent	fair	good
Inner guarantee	–	✓	✓
Outer approx	–	?	free

Table 1: Properties of the algorithms. Time models are continuous (CT), discrete (DT) or sampled data (SD). Complexity parameters are dimension ($d = d_x$ for CT or DT, $d = d_x + d_u$ for SD), grid resolution per dimension (n) and number of ellipses / support vectors (k). The complexities given for the ellipsoidal and support vector schemes are empirical estimates.

are compared on a double integrator and a chain of integrators, and then the support vector algorithm is further demonstrated on a (different) six dimensional anesthesia model and a twenty dimensional heat equation. The CT discriminating kernel algorithm is demonstrated on a twelve dimensional quadrotor model in [9]. For the SD algorithm, only toy examples are considered in [18], but the robustness capabilities of the discriminating kernel are used in [17] to rigorously analyze a *nonlinear* three dimensional quadrotor model.

A qualitative summary of the various algorithms for approximating the viability and discriminating kernels is given in table 1. The "level set" column refers to formulations based on the HJ PDE (for example, see [16, 18]), and is included here as a representative of nonparametric grid-based approaches. The grid is assumed to contain $\mathcal{O}(n)$ nodes per dimension, and hence the computational cost is at least $\mathcal{O}(n^d)$. The viability and discriminating kernel algorithms described in [19, 4] can provide a guarantee of underapproximation but require greater resolution to achieve the same accuracy as the HJ PDE formulation.

Figure 1 shows the scalability of the algorithms on the chain of integrators example from [14]. The curve labelled "polytope" corresponds to another algorithm from [14], but is representative of the exponential scalability to be expected from the nonparametric algorithms. The complexity of the ellipsoidal and support vector algorithms is known to be polynomial in state space dimension, but the cubic and quadratic complexities shown in the table are experimental estimates. The support vector algorithm's computational cost is driven almost entirely by a large convex optimization; from the figure it is clearly superlinear but may be subquadratic in practice. The size of an ellipsoid's representation is quadratic in dimension and the intersection operation requires a convex optimization over the parameters in this representation, so it is not surprising that the ellipsoidal algorithm's computational cost appears to be superquadratic.

Although not discussed here, the inner ellipsoids' representations can be used as a form of Lyapunov function to synthesize control signals that will keep a system within the viability or discriminating kernel over the analysis horizon [9]. The solution of the HJ PDE approximated by the level set method can also be used as a Lyapunov function. No similar construct is

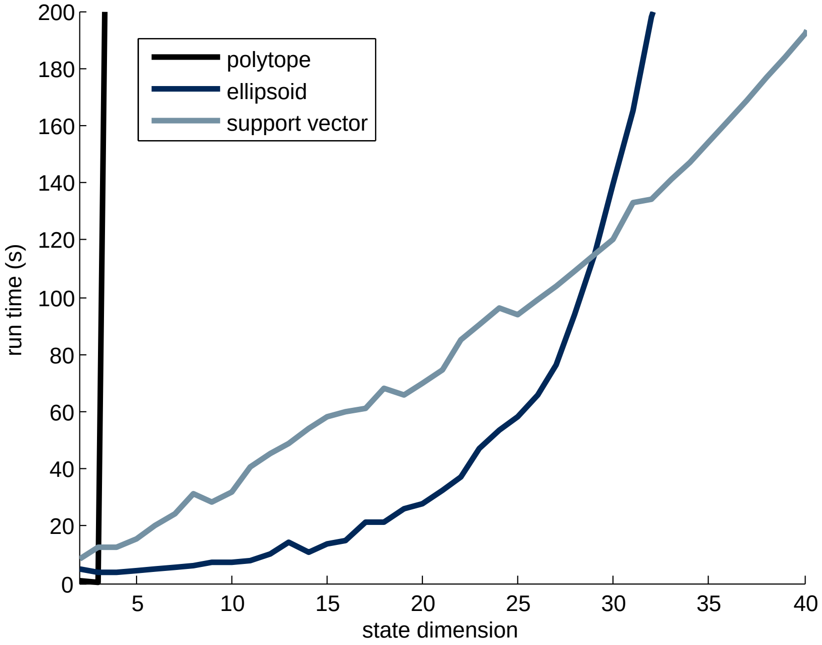

Figure 1: Comparison of DT viability kernel algorithm run times for a chain of integrators. Figure reproduced from [14].

currently available for the support vector approach.

6 Conclusions and Recommendations for Further Readings

The key components of the DT and CT viability algorithms were already present in [8], but we recommend [14] for a more complete treatment. The discriminating kernel case can be found in [9]. For the SD case we recommend using the most recent algorithm described in [17] because it is faster and more accurate; however, it may be difficult to digest this paper without first reading [18].

The support vector and ellipsoidal algorithms each have their own benefits. The support vector algorithm tends to be much more accurate because its representation is closed under the key operations in the algorithm, and it appears to be more efficient experimentally. The ellipsoidal algorithm is still polynomial complexity, supports all three models of time, can be used to synthesize safe controls, and can be made robust to uncertainty through discriminating kernels. Despite its reduced accuracy, we are currently working most actively with the ellipsoidal algorithms because of these latter capabilities; however, we encourage efforts to extend the support vector approach in a similar manner.

Acknowledgements. The author would like to thank his co-authors Shahab Kaynama (who developed the ellipsoidal algorithm) and John N. Maidens (who worked with Shahab to develop the support vector algorithm). Practical implementation would have been impossible without the Ellipsoidal Toolbox [11] and CVX [7] software packages. The author has also had productive discussions on these topics with Alexandre Bayen, Thao Dang, Goran Freshe, Bruce Krogh, Alexander B. Kurzhanski, Alex A. Kurzhanskiy, Oded Maler and Patrick Saint-Pierre. The thesis of Colas Le Guernic [12] heavily influenced our approach to the support vector algorithm.

References

[1] Jean-Pierre Aubin, Alexandre M. Bayen, and Patrick Saint-Pierre. *Viability Theory: New Directions*. Systems & Control: Foundations & Applications. Springer, 2011. doi:10.1007/978-3-642-16684-6.

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

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[4] P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Set-valued numerical analysis for optimal control and differential games. In M. Bardi, T. E. S. Raghavan, and T. Parthasarathy, editors, *Stochastic and Differential Games: Theory and Numerical Methods*, volume 4 of *Annals of International Society of Dynamic Games*, pages 177–247. Birkhäuser, 1999.

[5] L. C. Evans and P. E. Souganidis. Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. *Indiana University Mathematics Journal*, 33(5):773–797, 1984.

[6] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler. SpaceEx: Scalable verification of hybrid systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Proceedings of the International Conference on Computer Aided Verification*, number 6806 in Lecture Notes in Computer Science, pages 379–395. Springer, 2011. doi:10.1007/978-3-642-22110-1_30.

[7] Michael C. Grant and Stephen P. Boyd. Graph implementations for nonsmooth convex programs. In Vincent D. Blondel, Stephen P. Boyd, and Hidenori Kimura, editors, *Recent Advances in Learning and Control*, volume 371 of *Lecture Notes in Control and Information Sciences*, pages 95–110. Springer, 2008. doi:10.1007/978-1-84800-155-8_7.

[8] Shahab Kaynama, John Maidens, Meeko Oishi, Ian M. Mitchell, and Guy A. Dumont. Computing the viability kernel using maximal reachable sets. In *Hybrid Systems: Computation and Control (HSCC)*, pages 55–64, Beijing, China, 2012. doi:10.1145/2185632.2185644.

[9] Shahab Kaynama, Ian M. Mitchell, Meeko M. K. Oishi, and Guy A. Dumont. Scalable safety-preserving robust control synthesis for continuous-time linear systems. *IEEE Transactions on Automatic Control*, Early Access, 2015. doi:10.1109/TAC.2015.2411872.

[10] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis: Internal approximation. *Systems and Control Letters*, 41:201–211, 2000.

[11] Alex A. Kurzhanskiy and Pravin Varaiya. Ellipsoidal toolbox. Technical Report UCB/EECS-2006-46, Department of Electrical Engineering and Computer Science, University of California, Berkeley, May 2006. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-46.html.

[12] Colas Le Guernic. *Reachability Analysis of Hybrid Systems with Linear Continuous Dynamics*. PhD thesis, Université Joseph Fourier (Grenoble I), 2009.

[13] Colas Le Guernic and Antoine Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250 – 262, 2010. doi:10.1016/j.nahs.2009.03.002.

[14] John N. Maidens, Shahab Kaynama, Ian M. Mitchell, Meeko M. K. Oishi, and Guy A. Dumont. Lagrangian methods for approximating the viability kernel in high-dimensional systems. *Automatica*, 49(7):2017–2029, July 2013. doi:10.1016/j.automatica.2013.03.020.

[15] Ian M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In Alberto Bemporad, Antonio Bicchi, and Giorgio Buttazzo, editors, *Hybrid Systems: Computation and Control (HSCC)*, number 4416 in Lecture Notes in Computer Science, pages 428–443. Springer Verlag, 2007. doi:10.1007/978-3-540-71493-4_34.

[16] Ian M. Mitchell, Alexandre M. Bayen, and Claire J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005. doi:10.1109/TAC.2005.851439.

[17] Ian M. Mitchell and Shahab Kaynama. An improved algorithm for robust safety analysis of sampled data systems. In *Hybrid Systems: Computation and Control (HSCC)*, pages 21–30, 2015. doi:10.1145/2728606.2728619.

[18] Ian M. Mitchell, Shahab Kaynama, Mo Chen, and Meeko Oishi. Safety preserving control synthesis for sampled data systems. *Nonlinear Analysis: Hybrid Systems*, 10:63–82, 2013. doi:10.1016/j.nahs.2013.04.003.

[19] Patrick Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics and Optimization*, 29:187–209, 1994. doi:10.1007/BF01204182.