Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Randomized Algorithm: A viability computation

B. Djeridane * **E. Cruck** ** **J. Lygeros** ***

* *ETH-Zurich, Automatic Control Laboratory,8092 Zurich, Switzerland,djeridane@control.ee.ethz.ch*
** *ETH-Zurich, Automatic Control Laboratory,8092 Zurich, Switzerland,cruck@control.ee.ethz.ch*
*** *ETH-Zurich, Automatic Control Laboratory,8092 Zurich, Switzerland,lygeros@control.ee.ethz.ch*

Abstract: We deal with the problem of computing maximal viability sets for nonlinear continuous or hybrid systems. Our main objective is to beat the *curse of dimensionality*, that is, we wan to avoid the exponential growth of required computational resource with respect to the dimension of the system. We propose a randomized approach for viability computation: we avoid griding the state-space, and use random extraction of points instead. This algorithm was implemented successfully to linear and nonlinear examples. We provide comparison of our results with results of other method.

Keywords: Viability, Randomized technique.

1. INTRODUCTION

Because of their importance in applications ranging from engineering to biology and economics, questions of reachability, viability and invariance of sets have been studied extensively in the dynamic and control literature. More recently, reachability computation has gained attention in the context of safety problems, such as air traffic management Livadas et al. [2000] and flight control Lygeros et al. [1999].

The characterization of viability concepts can be formulated using optimal control or game theory problems Lygeros [2004], and the solution can be characterized using variants of the Hamilton-Jacobi-Bellman equations. Efficient algorithms developed to solve such PDE Mitchell et al. [2001] can then be used to solve the reachability problem numerically. In theory, these numerical tools are appropriate for systems of any state dimension. However the computational cost of viability analysis in higher than four dimensions is no practical option. The main reason is the exponential increase in computing time and resource requirements which clearly limits the use of these tools. Some efforts have been done to extend the use of continuous system reachability tools to six dimensions, thus making them applicable to a number of interesting case studies in the area of aeronautics Kitsios and Lygeros [2005], but we are still limited in the dimension size of the problem to be treated.

Another approach to compute a viability set is based on nonsmooth analysis and viability theory Aubin [1991]. The basic idea is to compute directly the viability kernel without solving a Hamilton-Jacobi-Belman's equation. The development of computational tools to support the numerous viability theory concepts is an ongoing effort Cardaliaguet et al. [1999]. The numerical tools based on set valued analysis used for viability computations come with theoretical proofs of convergence and numerical accuracy, but they also suffer of the *curse of dimensionality*.

It is well known that, frequently, the complexity for computing the viability kernel is very high even for low state space dimensions. Hence, it appears natural to seek approximate method involving suitable discretization to facilitate computer work. The computationally efficient full discretization of state space has been a challenge for researchers for many decades. To overcome this difficulty, for example Chow and Tsitsiklis [1988] propose a multi-grid method adapted to a class of discrete time, continuous state, discounted, infinite horizon dynamic programming problems to improve the computational complexity for this class of system. Another method called cell-mapping uses the basic idea of considering the state space not as continuous but rather as a collection of large number of state cells with each cell being taken as a state entity Hu and Chiu [1986a,b]. This method has been successfully applied to optimal control problem Bursal and Hsu [1989] by representing all the admissible controls and their duration application as finite set. Then, the process of extracting optimal control results from the family of controlled mappings becomes a matter of systematic search.

All these methods suffer from an exponential computational complexity. Recently, new idea have emerged that could find solution at "most of time" for particular problem with "high confidence" that the candidate solution is the true solution. Randomized algorithms are gaining popularity among control theory community Ariola et al. [2003], and have been applied successfully to compute a reachable set using neural networks Djeridane et al. [2007] and to system identification of ARMA Model Vidyasagar and Karandikar [2002]. Another application is the identification of a piecewise affine system presented in Prandini [2004].

In this paper, we present the initial steps of an approach motivated by learning theory Vidyasagar [1998] that aims to beat the curse of dimensionality for viability kernel computation by generating points randomly instead of gridding over the whole state-space. Once we have all sample points, we start by ordering our sample points according to lexicographical scheme and afterwards all our operations are based on this new representation of our points. With this scheme we took the advantage

to be able to use efficient algorithms for searching inside a set, such as list algorithm.

The paper is organized into sections as follows: Section 2 deals with a problem statement and provides some background material on the viability theory. Section 3 presents our randomized algorithm used to compute the viability kernel. In section 4 we show many results where our method has been applied successfully from linear to nonlinear system and finally we conclude our paper with conclusion 5 and some directions for our future research.

2. PROBLEM DESCRIPTION

Consider a continuous time control system,

$$\dot{x} = f(x,u) \tag{1}$$

with $x \in \mathbb{R}^n$, $u \in U \subseteq \mathbb{R}^m$ and $f(\cdot,\cdot) : \mathbb{R}^n \times U \to \mathbb{R}^n$. We assume that U is compact convex and that f is bounded and Lipschitz continuous with respect to the first variable and continuous with respect to the second variable. Under these assumptions, for any $t \in \mathbb{R}^+$, $x \in \mathbb{R}^n$ and time-measurable $u : \mathbb{R}^+ \to U$, system (1) admits a unique solution. We denote this solution by $x(\cdot;x,u(\cdot))$.

Given a closed set of states $K \subseteq \mathbb{R}^n$, the viability set we would like to compute is the largest set of states $x \in K$ for which there exists a control input $u(\cdot)$ that keeps the solution $x(\cdot;x,u(\cdot))$ in K. This can be rewritten as

$$Viab(K) = \{x \in \mathbb{R}^n | \, \exists u(\cdot) \in U, \forall \tau \in \mathbb{R}^+ \; x(\tau;t,x,u(.))\}.$$

In the literature, this set has been characterized indirectly, using optimal control Lygeros [2004], and directly, using nonsmooth analysis tools Aubin [1991]. Here we adopt the latter approach. Following Cardaliaguet et al. [1999] we reformulate the dynamics (1) as

$$\dot{x}(t) \in F(x(t)),$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is the set-valued map defined by

$$\forall x \in \mathbb{R}^n, \quad F(x) := \{f(x,u), u \in U\}.$$

It has been proved in Cardaliaguet et al. [1999] that $Viab(K)$ can be computed as the limit of the non-increasing sequence of closed sets defined by

Algorithm 1 Computation of the viability kernel: Partial discretization

$K_\varepsilon^0 := K$
$K_\varepsilon^{p+1} := \{x \in K_\varepsilon^p | \; [x + \varepsilon F_\varepsilon(x)] \cap K_\varepsilon^p \neq \emptyset\}$

where F_ε is an approximation of F which satisfies the following properties:

(1) F_ε is upper semi-continuous with convex compact nonempty values.
(2) $Graph(F_\varepsilon) \subset Graph(F) + \phi(\varepsilon)B$ where $lim_{\varepsilon \to 0^+} \phi(\varepsilon) = 0^+$
(3) $\forall x \in X, \cup_{\|x-y\| \leq M\varepsilon} F(y) \subset F_\varepsilon(x)$

where B is unit ball in $\mathbb{R}^n$.

Remark 1. If the set-valued map F is bounded and Lipschitz in K, namely if

$$\exists M \geq 0, \quad \forall x \in K, \forall y \in F(x), \qquad \|y\| \leq M,$$

and if

$$\exists l > 0, \quad \forall (x_1,x_2) \in K^2, \quad F(x_1) \subset F(x_2) + l\|x_1 - x_2\|B,$$

then a natural choice of approximation is

$$F_\varepsilon(x) := F(x) + \varepsilon MlB.$$

Let us set

$$\widetilde{Viab}_\varepsilon(K) = \cap_{p=0}^{\infty} K_\varepsilon^p.$$

We know from Cardaliaguet et al. [1999] that

$$\lim_{\varepsilon \to 0^+} \widetilde{Viab}_\varepsilon(K) = Viab(K)$$

and that

$$\forall \varepsilon > 0, \quad \widetilde{Viab}_\varepsilon(K) \subset Viab(K) + M\varepsilon B.$$

Therefore, in order to get a good approximation of $Viab(K)$, it is enough to compute a good approximation of $\widetilde{Viab}_\varepsilon(K)$ for ε small enough.

A natural idea if to use discrete approximations of the sets K_ε^p. For this purpose, let us introduce a finite state space discretization K_h, where $h > 0$ is defined below, and let us associate with K_h a finite approximation $F_{\varepsilon,h} : K_h \to K_h$ of F_ε. Then the following algorithm terminates in finite time.

Algorithm 2 Finite computation for the approximation of the viability kernel

$K_{\varepsilon,h}^0 := K_h$
$K_{\varepsilon,h}^{p+1} := \{z_h \in K_{\varepsilon,h}^p : \; [z_h + \varepsilon F_{\varepsilon,h}(z_h)] \cap K_{\varepsilon,h}^p \neq \emptyset\}$

Let $\hat{p}$ be such that

$$\forall p \geq \hat{p}, \quad K_{\varepsilon,h}^p = K_{\varepsilon,h}^{\hat{p}}.$$

We set

$$\widehat{Viab}_{\varepsilon,h}(K) = K_{\varepsilon,h}^{\hat{p}} + hB.$$

Then if the fully discrete approximation $F_{\varepsilon,h}$ has good properties, $\widehat{Viab}_{\varepsilon,h}(K)$ is a good approximation of $Viab(K)$.

In Cardaliaguet et al. [1999], the choice of $F_{\varepsilon,h}$ is based on the projection of F_ε on a fixed regular grid, which yields an exponential growth of the size of the finite representation of the state-space as a function of the dimension of the state. In the next section, we present a choice for $F_{\varepsilon,h}$ based on randomized techniques. Then the size of the finite state space can be fixed arbitrarily. The price is a good approximation *in average*.

3. RANDOMIZED APPROACH

3.1 Viability kernel approximation

Instead of using a regular griding of the state space we approximate the set K using a finite number of points generated randomly. Let $\{x^i\}_{i \leq N}$ denote these points. And we define the *state discretization index h*. As usual with a numerical explicit scheme, the space and time discretization steps are linked up. Therefore the time-step ε is determined after the random sampling of K in order to ensure consistency of the approximation.

The procedure for computing an approximation of the viability kernel is relatively simple once you can check if a point is *locally viable* in a set K, that is, if it has a successor in K. The procedures for performing this test are provided below. The overall viability kernel approximation procedure is summarized in algorithm 3.

We should stress that when using the local viability test described below, this algorithm provides an upper approximation

Algorithm 3 Computation of the viability kernel

Generate randomly N points x^i over K
Compute the state discretization index h
Initially set $K^0 = \{x^i : i \in \leq N\}$
Initially set $p = 0$
repeat
  for all $i = 1$ to N **do**
    if x^i is locally viable in K^p **then**
      Keep x^i inside K^p
    else $\{x^i$ is not locally viable in $K^p\}$
      Remove x^i from K^p
    end if
  end for
  M number of points x^i removed from K^p
  $N = card(K^p)$
  $p = p + 1$
until $N = 0$ or $M = 0$

of the viability kernel *with a certain confidence*. Indeed, our approximation method guarantees that a point the successor of which in inside the real kernel is not eliminated only if our initial random sampling is *good enough*. Therefore, the randomization of the methods loses the guarantee of over-approximation of Cardaliaguet et al. [1999].

3.2 Local viability test

Computation of the successors of x^i From Cardaliaguet et al. [1999], the assumptions that must be satisfied by the finite approximation $F_{\varepsilon,h}$ when using a regular griding are the following

(1) $Graph(F_{\varepsilon,h}) \subset Graph(F_\varepsilon) + \psi(\varepsilon,h)B$ where $lim_{\varepsilon\to 0^+, \frac{\varepsilon}{h}\to 0} \frac{\psi(\varepsilon,h)}{h} = 0$
(2) $\forall x \in X, \cup_{\|x^i - y\| \leq h}[F_\varepsilon(y) + hB] \cap X_h \subset F_{\varepsilon,h}(x^i)$

where X_h denotes the grid. However, it is clear from the proof of convergence that the second condition is only sufficient to ensure that $\widehat{Viab}_{\varepsilon,h}(K)$ is an over-approximation of $\widetilde{Viab}_\varepsilon(K)$. It leads to the following test for local viability of a point x^i in a set $K^p_{\varepsilon,h}$:

$$[x^i + \varepsilon F_\varepsilon(x^i) + (\varepsilon l + 2)hB] \cap K^p_{\varepsilon,h} \neq \emptyset,$$

in which l denotes the Lipschitz constant of the dynamics F. This condition can be rewritten as

$$\exists z \in F_\varepsilon(x^i) \text{ such that } \exists k \leq N, \quad ||x^i + \varepsilon z - x^k|| \leq (\varepsilon l + 2)h.$$

Now, we recall that $F(x) = \bigcup_{u\in U} f(x,u)$. Because of the convexity of U and the continuity of f, we can assume that there exists $U_\varepsilon \supset U$ such that

$$\forall x, \qquad F_\varepsilon(x) = \bigcup_{u\in U_\varepsilon} f(x,u).$$

In order to perform the local viability test, we use a finite approximation $U_{\varepsilon,h}$ of U_ε. Note that the number of points in $U_{\varepsilon,h}$ is linked to h. Then the condition for local viability can be written

$$\exists u \in U_{\varepsilon,h}, \exists k \leq N, \quad ||x^i + \varepsilon f(x^i,u) - x^k|| \leq (\varepsilon l + 2)h. \quad (2)$$

In the current implementation of the algorithm, we use a dichotomy method to determine $U_{\varepsilon,h}$.

Membership test In order to test condition (2) for a given value of u, a simple idea is to compute the distance between this point x^i and all the points which constitute the set $K^p_{\varepsilon,h}$. However this naive approach needs a huge amount of computing resources if the number of sample points is large. In order to handle this problem, we propose a technique based on lexicographical ordering of the sample points and the use of the infinity norm. The principle is explained in Figure 1 for the two-dimensional case.

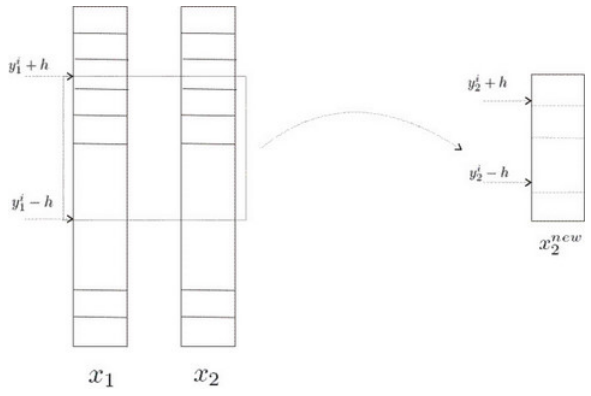

Fig. 1. Diagram showing how we check that x^i belong to a Set

Using the *sort* function of Matlab©, this is performed has follows: First sort x_1 in descent order

$$x_1 = sort(x_1, 'descent')$$

Then compute the successor y^j of x^j, afterwards check if

$$[y_1^i - (\varepsilon l + 2)h, y_2^i + (\varepsilon l + 2)h] \cap x_1 = S_1$$

if S_1 is nonempty set with $S_1 = [x_1^{\alpha_1}, x_1^{\alpha_2}, \cdots, x_1^{\alpha_{card(S_1)}}]$, then we set x_2^{new} as sub-vector of x_2, where those elements are defined by

$$x_2[\alpha_1 : \alpha_{card(S_1)}]$$

And now, we do the same procedure for x_2^{new} to check if it's belong to the set K or not. This procedure is formulated in the following algorithm

Algorithm 4 Membership test of x^i to K^p

Compute the successor y^i of the point x^i
if $[y_1^i - (\varepsilon l + 2)h, y_1^i + (\varepsilon l + 2)h] \in x_1$ **then**
  if $[y_2^i - (\varepsilon l + 2)h, y_2^i + (\varepsilon l + 2)h] \in x_2^{new}$ **then**
    $\cdots$
    if $[y_n^i - (\varepsilon l + 2)h, y_n^i + (\varepsilon l + 2)h] \in x_n^{new}$ **then**
      x^i belongs to K^p
    else {otherwise}
      x^i does not belong to K^p
    end if
    $\cdots$
    x^i does not belong to K^p
  end if
  x^i does not belong to K^p
end if

Sorting is highly desirable for searching in large quantities of information, because it's greatly improves the efficiency of searching. For this reason we are sorting our data. There are many sorting algorithm which could be used like: Heapsort, Quicksort Radix Sort. In practice Quicksort is often the best choice because it is remarkably efficient on the average running time Cormen et al. [1990].

4. RESULTS

To demonstrate the efficiency of our method, we have applied it to linear and nonlinear system in 2D and 3D, and we compared our results to results obtained using other computation methods. We have performed all the computations on a Pentium 4, 3.2 GHz processor running on WindowsXP.

4.1 Linear system

We have double and triple integrator examples as test benches for the computation methods.

2D example Consider the linear system defined by:

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= u\end{aligned}$$

subject to state constraints defined by

$$(x_1,x_2) \in K = [-1,1]\times[-1,1]$$

and control constraint $u \in [-1,1]$.

We have used the following parameters: h=$\frac{1}{\sqrt{(N)}}$, $\varepsilon = \sqrt{(h)}$ to compute approximate viability kernel using our randomized algorithm.

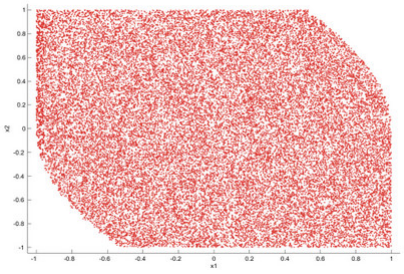

Fig. 2. Viability set of 2D example computed with randomized algorithm

3D example Consider the following linear system

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= x_3\\ \dot{x}_3 &= u\end{aligned}$$

subject to state constraints

$$(x_1,x_2,x_3) \in K = [-1,1]\times[-1,1]\times[-1,1]$$

and control constraint $u \in [-1,1]$. We have used the same parameters as previously.

We summarize the results in the following table in which Method 1 is Level Set Toolbox, Method 2 is our randomized algorithm and we have compared them to "exact" solution which is also computed using level set toolbox but with higher accuracy. We have performed computation using the two methods as mentioned for all our example and we have obtained the following results:

	2D	3D	4D	5D
Method 1 (5%)	549	103823	1874161	2476099
Method 2 (5%)	22000	65000	120000	350000

From this table it's clear that our method is outperforming regarding to other method, because ours is growing polynomially

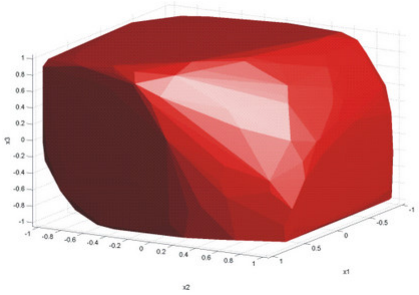

Fig. 3. Viability set of 3D example computed with randomized algorithm

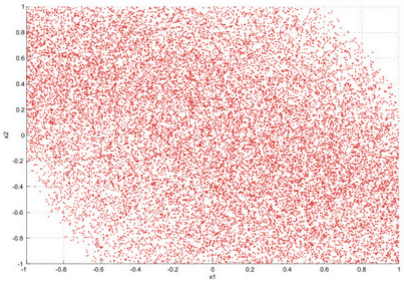

Fig. 4. Projection of a viability set of 3D example on the plane (x_1,x_2)

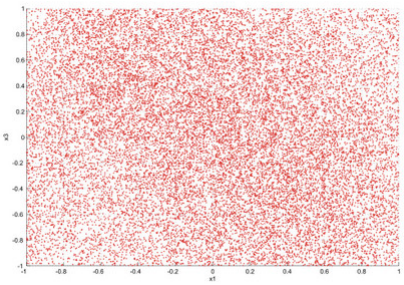

Fig. 5. Projection of a viability set of 3D example on the plane (x_1,x_3)

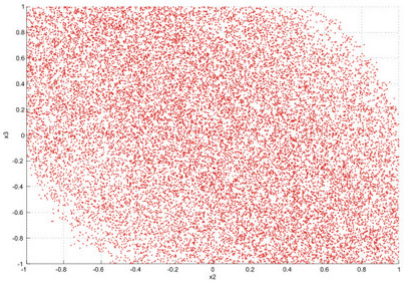

Fig. 6. Projection of a viability set of 3D example on the plane (x_2,x_3)

respect the dimensions of the state space, which imply that our method is independent from the state-space.

However, with the approach proposed in this paper, we are loosing on the accuracy of our approximation and this the price for random extracting samples from the state space, but the

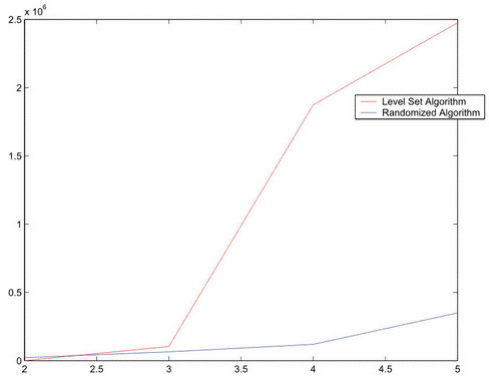

Fig. 7. Growth of the samples respect to the dimension of the space

accuracy could be improved it by doing refinement principle (Multi-grid approximation) to get better approximation. We should note, that we have done sketch implementation of the algorithm (2)and could significantly improved, for example we could use an intelligent strategy of search through the state-space, but with our approach we still have big advantage over simple gridding of the state-space.

4.2 Nonlinear system

To proof the applicability of our method to nonlinear system. Let us consider the following system

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned}$$

subject to constraint $K = [-1,1] \times [-1,1]$ and $U = [-1,1]$. We have compute an approximation of the viability kernel using 22000 points, and the results is shown in the figure 8

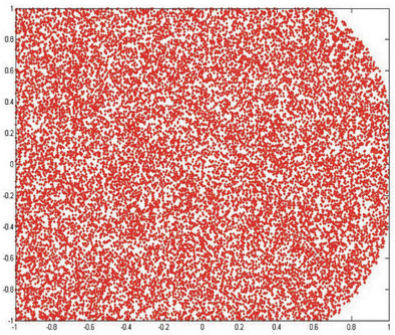

Fig. 8. Viability set of 2D nonlinear example computed with randomized algorithm

5. CONCLUSION

The goal of this paper was to propose an approximation method that circumvents the curse of dimensionality encountered in viability computations. We have used our algorithm on a set of different examples with good results. In current work we are trying to extended this trial solution to more complicated system. A direct advantage of our method is that the size of the state space to be explored can be fixed in advance as the number of points taken randomly in the initial set. We are currently working on a multi-step refinement process to obtain fine approximation with a limited number of points. The hope is that the computation required to achieve a certain accuracy will grow polynomially with the dimension of the system.

The real advantages introduced by our randomized method have to be balanced with the loss of the guarantee of obtaining a meaningful approximation. The probability that the output of our algorithm is a *bad* approximation of the real viability kernel is strictly positive. We are working on a convergence proof in order to provide a statistical guarantee of convergence in a confidence interval. The proof is rather challenging since it requires mixing elements from randomized techniques together with non-smooth analysis.

Another direction for future work is the improvement of the implementation of our algorithm. In the current implementation, the most time-consuming aspect is the test of membership of a successor point to the approximation since we need to find the closest point in a finite, but large, set of points. This is not surprising since this test is the core of the direct computation approach (as opposed to the level-set methods). In future work, we shall focus on selecting appropriate algorithms for sorting and parsing lists; tree search algorithms or graph search algorithms could improve the performances of the membership test.

ACKNOWLEDGEMENTS

Work supported by the European Commission under the HYCON Network of Excellence, IST-511368 and the HYGEIA project,NEST-4995.

REFERENCES

M. Ariola, C.T. Abdallah, and V. Koltchinskii. Applications of statistical-learning control in system and control. In *IFAC Workshop*, Villa Erba, Cernobbio-Como, Agust 29-31 2003. on Adaptation and Learning in Control and Signal Processing.

J. P. Aubin. *Viability theory*. Bikhauser, Boston, 1991.

F.H. Bursal and C.S. Hsu. Application of a cell-mapping method to optimal control problems. *International Journal of Control*, (5):1505–1522, 1989.

P. Cardaliaguet, M. Quincampoix, and P. Saint-Pierre. Stochastic and differential games: Theory and numerical methods. In *Annals of the International Society of Dynamic Games*. Birkaser, 1999.

C. Chow and J.N. Tsitsiklis. An optimal multigrid algorithm for continuous state discrete time stochastic control. In *Conference on Decision and Control*, pages 1908–1912, Austin, Texas, December 1988.

T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. McGraw-Hill, 1990.

B. Djeridane, E. Cruk, and J. Lygeros. A learning theory approach: to the computation of reachable sets. In *European Control Conference*, Kos, July 2007.

C.S. Hu and H.M. Chiu. A cell mapping method for nonlinear deterministic and stochastic systems -part i: The method of analysis. *Trans. ASME J. Appl. Mech.*, 53(3):702–710, 1986a.

C.S. Hu and H.M. Chiu. A cell mapping method for nonlinear deterministic and stochastic systems -part ii: Examples of application. *Trans. ASME J. Appl. Mech.*, 53(3):695–701, 1986b.

I. Kitsios and J. Lygeros. Aerodynamic envelope computation for safe landing of the hl-20 personnel launch vehicule using hybrid control. In *Mediterranean Conference on Control and Automation*, Limassol, June 2005.

C. Livadas, J. Lygeros, and N. Lynch. High-level modeling and analysis of the traffic alert and collision avoidance system (tcas). *Proceeding of the IEEE*, 88(7):926–948, July 2000.

J. Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40:917–927, 2004.

J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35:349–370, 1999.

I. Mitchell, A.M. Bayen, and C. Tomlin. Validating a hamilton jacobi approximation to hybrid reachable sets. In M. DiBenedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid systems: Computation and Control*, pages 418–432. Springer-Verlag, 2001.

M. Prandini. Priecewise affine systems identification: a learning theoretical approach. In *Conference on Decision and Control*, pages 3844–3849, Bahamas, December 2004.

M. Vidyasagar. Statistical learning theory and randomized algorithms for control. *IEEE Control Systems*, pages 69–85, December 1998.

M. Vidyasagar and R.L. Karandikar. A learning theory approach to system identification. In *Technical committe on robust control*, Portugal, July 2002. Workshop of IFAC.