REACHABILITY ANALYSIS USING PROXIMITY BASED AUTOMATA ⋆

Jim Kapinski, Klaus Schmidt, and Bruce H. Krogh

Carnegie Mellon University
Dept. of Electrical and Computer Engineering
Pittsburgh, PA 15213 USA
{jpk3|klaussch|krogh}@andrew.cmu.edu

Abstract: This paper introduces *proximity based automata* (PBA) to perform reachability analysis for discrete-time, continuous-state systems. The notion of proximity of states in the continuous valued state space is used to evaluate the response of the system efficiently. A PBA is constructed using ellipsoidal techniques and tools for solving optimization problems constrained by linear matrix inequalities. Several empirical studies demonstrate how trade-offs between conservativeness and computational complexity can be exploited by varying parameters in the procedure for constructing the PBA.

Keywords: reachability, hybrid systems, verification, dynamic systems

1. INTRODUCTION

There has been much interest in recent years in developing methods for formally verifying hybrid systems (Alur et al., 1996; Dang and Maler, 1998; Silva et al., 2002). Typically, the goal is to verify that a hybrid system satisfies some safety property for an entire range of initial conditions and system parameters. The most computationally expensive aspect of verifying hybrid systems is the computation of the set of reachable states for a system due to variations in operating conditions characterized by system parameters, initial conditions and ranges of inputs (controls and disturbances). Many different types of reachability have been studied (Chutinan and Krogh, 2003; Dang and Maler, 1998; Henzinger et al., 1997; Jönsson, 2002; Mitchell et al., to appear), but these techniques only provide methods for estimating the set of reachable states in the continuous-valued state space and there is no systematic way of exploring multiple paths in the reachable space due to inputs.

This work extends the concept of systematic simulation introduced in Kapinski et al. (2003) by representing the set of reachable states of the infinite state system with a finite state machine, called a *proximity based automaton* (PBA). The definition of a PBA is based on the notion that trajectories from regions that are near each other in the state space will evolve in a similar manner. We use ellipsoidal methods to perform reachability analysis for dynamic systems with bounded inputs (Kurzhanski and Vályi, 1997). These methods provide an efficient way of approximating reachable sets using semidefinite programming techniques (Boyd et al., 1994). The notion of proximity is exploited by merging PBA states that fulfill specified nearness conditions. The PBA can also be used to perform refinement of the

⋆ This research was supported in part by Ford, the US Defense Advance Projects Research Agency (DARPA) contract nos. F33615-00-C-1701 and F33615-02-C-4029, US Army Research Office (ARO) contract no. DAAD19-01-1-0485, the US National Science Foundation (NSF) contract no. CCR-0121547, and the Institute of Control Engineering and Automation at the University of Erlangen-Nuremberg.

reachable set approximation and the construction easily extends to hybrid systems. Further, we improve on systematic simulation by introducing a computationally efficient grid based method for testing proximity of states in the PBA.

Some researchers have investigated reachability of hybrid systems with discrete-valued inputs (Cury et al., 1998; Raisch and O'Young, 1998; Moor et al., 2004). We consider systems where the continuous set of available control inputs is covered by a finite collection of sets. The information contained in the PBA is used to perform refinement of the representation if necessary. We exploit the trade-off between conservativeness and computational complexity by varying parameters of the reachability analysis.

2. THE REACHABILITY PROBLEM

A discrete-time dynamic system (DDS) consists of a state space $X \subseteq \mathbb{R}^n$, a set of initial states $X_0 \subseteq X$, an input set $U \subseteq \mathbb{R}^m$, which covers control inputs as well as disturbance inputs, and an update equation $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. In this paper, we assume the dynamics of the DDS are linear; that is, there are matrices A and B such that $f(x, u) = Ax + Bu$.

Definition 2.1. A sequence $(x_0, u_0, x_1, u_1, \ldots)$ is a *run* of a DDS $\mathcal{H}$ if $x_0 \in X_0$ and for all $k \geq 0$, $x_{k+1} = f(x_k, u_k)$ and $u_k \in U$. $\mathcal{R}_\mathcal{H}$ denotes the set of all runs of $\mathcal{H}$, and $\mathcal{R}_\mathcal{H}(k)$ denotes the set of all runs of $\mathcal{H}$ of length less than or equal to k.

Definition 2.2. Given a DDS $\mathcal{H}$, a proximity based automaton (PBA) for $\mathcal{H}$ is a tuple, $P = (Q, \Sigma, E, Q_0, \lambda, \rho_X, \rho_U)$, where

- Q - a finite set of states;
- Σ - a finite set of symbols;
- $E \subseteq Q \times Q$ - a set of transitions;
- $Q_0 \subseteq Q$ - a set of initial states;
- $\lambda : E \to 2^\Sigma$ - a labelling function that assigns a set of input symbols to each transition;
- $\rho_X : Q \to C_{\mathbb{R}^n}$ - a function that assigns a region in $\mathbb{R}^n$ to each state in P. $C_{\mathbb{R}^n}$ denotes the set of compact, connected subsets of $\mathbb{R}^n$;
- $\rho_U : \Sigma \to C_{\mathbb{R}^m}$ - a function that assigns a region in $\mathbb{R}^m$ to each symbol in Σ,

and $\forall (q, q') = e \in E, \sigma \in \lambda(e), u \in \rho_U(\sigma)$ and $x \in \rho_X(q), \exists x' \in \rho_X(q')$ such that $f(x, u) = x'$.

Remark 2.1. Note that Definition 2.2 can easily be extended to piecewise-linear discrete time systems with a finite set of discrete modes $\mathcal{I}$ and a partition $\mathcal{X} = \{X_i\}_{i \in \mathcal{I}}$ of the continuous state space $\mathbb{R}^n$, where for each mode the dynamics are $f_i(x, u) = A_i x + B_i u$. Then the function $\rho_X : Q \to C_{\mathbb{R}^n} \times \mathcal{I}$ assigns a region in $\mathbb{R}^n$ and a mode in $\mathcal{I}$ to each state in Q.

Definition 2.3. The *language of the PBA* P, denoted $\mathcal{L}_P$, is the set of all input sequences $\pi = u_0 u_1 \cdots u_k$ such that:

$\exists \omega = q_0 q_1 \cdots q_{k+1} \ni \forall 0 \leq i \leq k, \sigma \in \lambda(q_i, q_{i+1})$ with $u_i \in \rho_u(\sigma)$.

The language of a PBA P will be a subset of U^*. Also, we define U^k as the set of all input sequences of length less than or equal to k.

Definition 2.4. Let P be a PBA as in Definition 2.2 and let $\mathcal{H}$ be a DDS. P is said to be a conservative estimate of the reachset $\mathcal{R}_\mathcal{H}(k)$ if $\mathcal{R}_\mathcal{H}(k) \subseteq \bigcup_{q \in Q} \rho_X(q)$.

Lemma 2.1. If the language of the PBA P contains U^k, then the PBA is a conservative estimate of the reachset for up until time k.

Proofs appear in Kapinski (2004).

Definition 2.5. An *invariant PBA* is one whose language is U^*.

Example 2.1. To illustrate the concept of a PBA, consider the servo system shown in Figure 1, which has first-order plant dynamics and a first-order reference signal filter, where u is an input reference signal, x_1 is the position, x_2 is the filtered version of u, and $e = x_2 - x_1$ is the error. The system matrices for the sampled version of the system, with $g = 10$, $\tau = 0.1$, and a sample period of 0.1, are

$$A = \begin{bmatrix} 0.368 & 0.368 \\ 0.000 & 0.368 \end{bmatrix} \qquad B = \begin{bmatrix} 0.264 \\ 0.632 \end{bmatrix}.$$

The set of initial conditions is given by a circle, centered at zero, with $\|x\|^2 \leq \frac{1}{10}$.

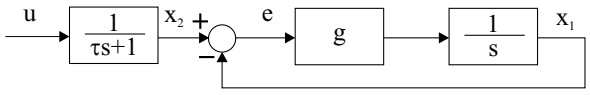


Fig. 1. A simple servo system.

Suppose we wish to verify that the servo system error does not exceed 1, that is, we want to guarantee $|x_1 - x_2| \leq 1$, for any input sequence $u_k \in [0.0, 1.0]$. Figure 2 shows the regions associated with an invariant PBA for System 2.1, with $\Sigma = \{1, 2, 3\}$, $\rho_U(1) = [0\ \frac{1}{3}]$, $\rho_U(2) = [\frac{1}{3}\ \frac{2}{3}]$, and $\rho_U(3) = [\frac{2}{3}\ 1]$. Other techniques can be used to analyze example 2.1, but the purpose of the example is to illustrate our computational method.

This example shows the general safety problems we want to address with the PBA in this paper. Given a region, $Fail \subset \mathbb{R}^n$, a DDS $\mathcal{H}$ is *k-step safe with respect to Fail* if $\mathcal{R}_\mathcal{H}(k) \cap Fail = \emptyset$ and *safe wrt. Fail* if $\mathcal{R}_\mathcal{H} \cap Fail = \emptyset$.

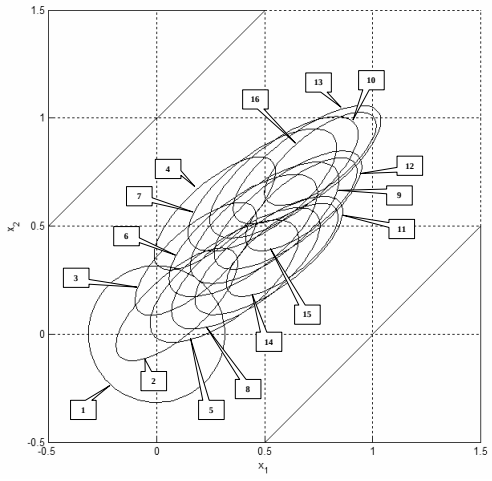


Fig. 2. The set of regions for the PBA of System 2.1. Each region is labelled with the corresponding state of the PBA.

3. REACHABILITY ANALYSIS USING PBA'S

The PBA reachability procedure (PRP), shown in Figure 3, constructs a PBA for a DDS. The procedure propagates regions forward using the reachability operator $Next$. If the reachability estimate resulting from $Next$ satisfies a nearness criterion for a region that is already associated with the PBA, the procedure merges the two regions; if the reachability estimate does not satisfy the nearness criterion for any region associated with P, a new state is created and associated with the new region. The result is that at each iteration the PBA represents all of the possible behaviors of the underlying system until time k. The procedure terminates if either there are no more states to explore or after K iterations, where K represents the termination time step.

The procedure requires a dynamical system description, a set of initial regions S, which includes the set of initial conditions X_0, a set of input symbols Σ, and a function $\rho_U : \Sigma \rightarrow C_{\mathbb{R}^m}$.

The PRP uses three functions: $Jointest$, $Join$, and $Next$. $Jointest(N, P)$ is a function that determines when two regions will be merged. It returns a state within PBA P that has a region associated with it that satisfies some nearness criterion to region N if such a state in P exists, otherwise it returns the empty set. In our experiments, we partition the space and merge regions when their center points are in the same element of the partition.

$Join(N, N')$ is a function that returns a region containing N and N'. Our implementation uses ellipsoidal representations; $Join(N, N')$ is an LMI that solves for the minimum volume ellipsoid that contains ellipsoids N and N' (Boyd et al., 1994).

```
/* Create PBA P^temp */
For each N ∈ S
  Insert q_N into Q^temp
  ρ_X^temp(q_N) := N
Q_0^temp := Q^temp
Q_W^temp := Q^temp
Loop K times
  Copy machine P^temp to P
  Q_W := ∅
  For each q ∈ Q_W^temp
    For each σ ∈ Σ
      /* Compute new region */
      N' := Next(ρ_X^temp(q), σ)
      q̂ := Jointest(N', P)
      If q̂ ≠ ∅
        /* New region should join with
           existing region */
        If N' ⊈ ρ_X(q̂)
          Ñ := Join(N', ρ_X(q̂))
          ρ_X(q̂) := Ñ
          Insert q̂ into Q_W
        Insert (q, q̂) into E
        λ(q, q̂) := λ(q, q̂) ∪ {σ}
      Else
        /* Create new state which is
           associated with new region */
        Insert q_N' into Q
        Insert q_N' into Q_W
        ρ_X(q_N') := N'
        Insert (q, q_N') into E
        λ(q, q_N') := σ
```

Fig. 3. PBA Reachability Procedure

$Next(N, \sigma)$ returns a region N', which contains the set of states that are reachable from region N in one time increment given some input in $\rho_U(\sigma)$, i.e. $f(N, \rho_U(\sigma)) \subseteq N'$. We compute the minimum volume ellipsoid containing $f(N, \rho_U(\sigma))$ using the ellipsoidal technique developed by Kurzhanski and Vályi (1997).

The PRP is initiated with a set of regions S. The set of initial conditions X_0 should be included in S, but additional regions may be added. It is often possible to add regions to S so that greater connectivity in the PBA array results from the PRP. We call this process *seeding the reachset*. This is beneficial because it can help reduce the amount of computation time spent on the PRP, and in some cases, it results in the PRP terminating with the final PBA being invariant. We have found that ellipsoids that are invariants given one input or given one mode are good seeding regions. (For a survey of methods for computing invariant regions, see Blanchini (1999).)

Lemma 3.1. For a DDS $\mathcal{H}$ and a given region $Fail \subset \mathbb{R}^n$, if $X_0 \subseteq Inv$ for some $Inv \subset \mathbb{R}^n$, $f(Inv, U) \subseteq Inv$, and $Inv \cap Fail = \emptyset$, then $\mathcal{H}$ is safe with respect to $Fail$.

If a region Inv, as described in lemma 3.1, can be identified, then the DDS is safe, and no further analysis needs to be performed.

The PRP generates a PBA whose language is guaranteed to contain U^K. Due to the merging of regions during the PRP procedure, which creates cycles in the finite state representation of the system, the PBA will possibly contain other strings, some of which are of infinite length.

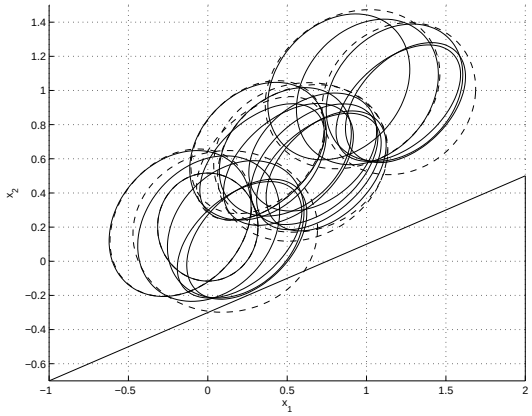


Fig. 4. Example of the reachset estimate for two different merging conditions. The 3D example is projected into the first two dimensions. The dashed lines correspond to a coarser merging condition grid and the solid lines correspond to a finer merging condition grid.

Proposition 3.1. If the PRP terminates with $Q_W^K = \emptyset$, then the PBA created is an invariant PBA.

In the case where $Jointest(N, P)$ is always empty, merging is never performed, and the complexity of the PRP is $O(n^3|\Sigma|^K)$, where the n^3 term is due to the linear algebra computations necessary for the propagation of ellipsoids (Kurzhanski and Vályi, 1997; Golub and Loan, 1996). If $Jointest(N, P)$ always returns a region in P for N to merge with, the complexity of the PRP is $O(n^{6.5}|\Sigma|K)$, where the $n^{6.5}$ term is due to the LMIs used to compute bounding ellipsoids (Vandenberghe et al., 1998). The main challenge in using the PRP method is to choose the $Jointest$ function such that a sufficient amount of merging occurs so as not to incur the exponential complexity, but not so much merging occurs so as to cause the approximation to become overly conservative.

4. EXPERIMENTS

In using the PRP, choosing an appropriate $Jointest$ function is essential for achieving a proper balance between computation time and conservativeness. For our PRP we use an implementation of $Jointest$ based on a partition of the state space. Each ellipsoid in the PBA is associated with the partition element containing the center point of the ellipsoid. Two ellipsoids are merged if their center points lie in the same partition element. It is expected that computation time will decrease as the merging condition is made more liberal, but conservativeness will increase. In our case, the merging condition being more liberal corresponds to the size of the partition elements being larger.

We would like to trade off conservativeness and computation time depending on the problem. Figure 4 illustrates this for an example that has three state variables and three inputs. (Details of this example are available from the authors.) The diagonal line in the figure indicates the failure region. The union of the solid ellipsoids represent the reachset estimate when the PRP is used with a grid size of 0.2. The union of the dashed ellipsoids represents the reachset found with a grid size of 0.4. In both cases, the PRP was performed for $K = 4$. Figure 4 shows that the reachset estimate using the grid size of 0.4 is too conservative, and the reachset estimate using the grid size of 0.2 is sufficient for the 4-step verification problem.

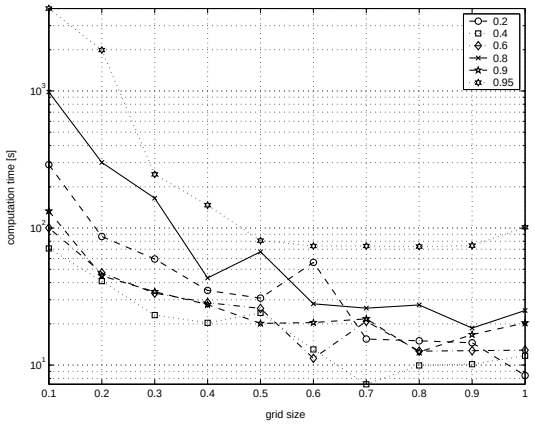
Several experiments were performed to determine the effect of merging condition, system stability, and state space dimension on computation time and conservativeness using the PRP. Our procedure was implemented in MATLAB, and all experiments were performed on an 800 MHz, Pentium III machine with 120MB of RAM running Windows ME.

The first experiment investigates how the computation time changes for different grid sizes and spectral radii of the system A matrix. We used randomly generated A matrices with spectral radii of 0.2, 0.4, 0.6, 0.8, 0.9, and 0.95 and used grid sizes between 0.1 and 1 with a granularity of 0.1. Figure 5 shows results for this experiment.

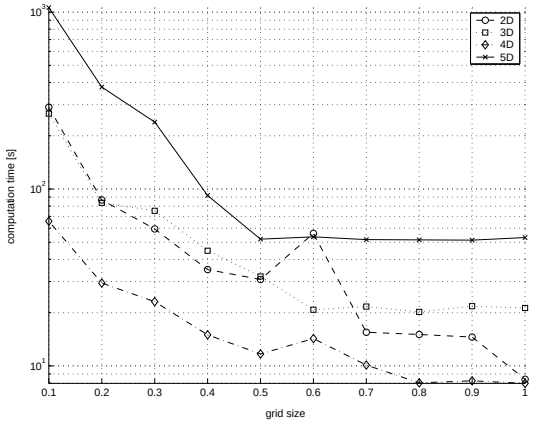
It can be seen that the computation time decreases as grid size increases and as spectral radius decreases. Note that, as the figures show particular examples, the computational effort is not monotone with respect to the joining condition, but that the general trend is what we expect, that is, that the computation time decreases as the merging condition becomes more liberal.

In the next experiment, system dynamics with spectral radii of 0.4 and system dimensions varying from 2 to 5 were randomly created. Figure 5 shows that the computation time for performing the PRP decreases as the grid size increases. Also, the increase in computation time in going from the 2D system to the 5D system is less than one order-of-magnitude.

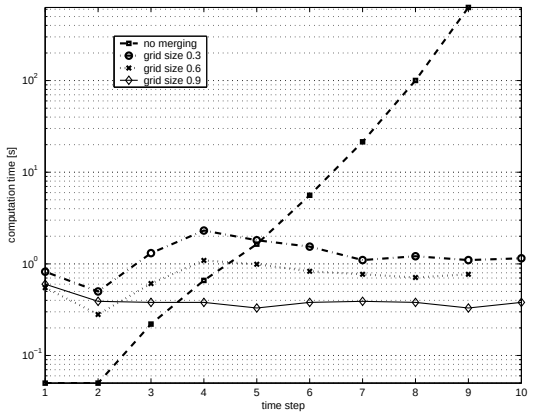
The last experiment compares the computation time per time step for the case where no merg-



(a)



(b)



(c)

Fig. 5. Computation times for randomly generated examples with (a) various spectral radii (b) various dimensions as a function of the merging condition (c) various merging conditions and number of iterations of the PRP.

ing takes place with the case with merging for different grid sizes for a two-dimensional system with a spectral radius of 0.4. Figure 5 suggests an exponential growth of the computation time if we do not merge any ellipsoids and it also shows that the computation time per time step remains at a nearly constant level until the exploration terminates in 9 and 10 time steps for grid size 0.6 and 0.3, 0.9, respectively.

5. REFINEMENT

Section 4 showed that the PRP could achieve less conservative approximations of the reachable set if the merging condition was conservative enough. Refining the PBA accomplishes the same task without repeating the entire reachability process. The refinement procedure consists of taking regions that are the result of a merging operation and lead to the failing region and *unmerging* them, that is, taking regions that result from a merging operation and breaking them into the two regions that were merged. In this way, the PBA that results from the PRP can be refined so that the conservativeness introduced by the merging operation is reduced.

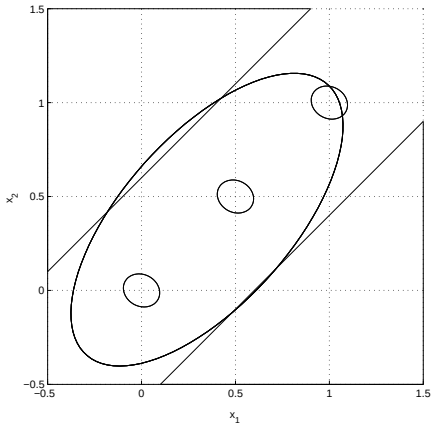
Consider again example 2.1. Suppose that we wish to show that $|x_1 - x_2| \leq .424$ for all input sequences. The reachset was first seeded with three regions, and the PRP was performed with a merging condition grid size of 1. The ellipsoids in Figure 6 (a) illustrate the result of the first two steps of the reachability procedure. Note that the regions associated with the PBA intersect the failing region. Since this intersection may be due to the conservativeness introduced from the merging operation, a refinement procedure was performed on the PBA.

Figure 6 (b) shows the final result of the refinement procedure. Note that the failing ellipsoid from the previous figure is removed and replaced with the one step reachable sets from the failing region, the initial condition set, and the seeding regions. This final collection of regions satisfies the safety condition.

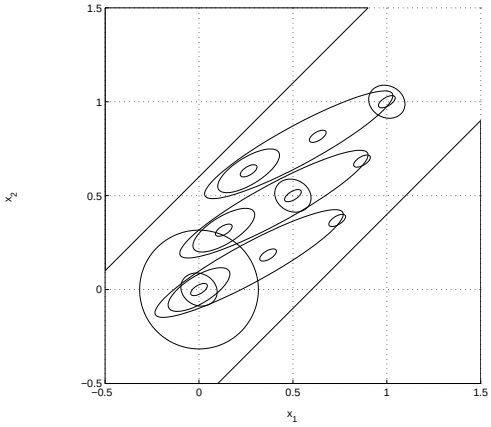
6. CONCLUSIONS

This paper presents a method for constructing a finite-state structure, the proximity based automaton (PBA), to guide reachability analysis for discrete-time continuous systems. The PBA is a discrete state, input driven representation of the system dynamics. Empirical studies confirm the efficiency that can be achieved with the PBA and illustrate the tradeoffs to be made between conservativeness and computation time.

We are currently developing other applications of the PBA, including the use of the PBA representation to perform safe input sequence synthesis (Schmidt et al., 2004). The PBA can also be extended to hybrid systems. This makes it possible to compute reachability for nonlinear systems by approximating them with a piecewise affine switched mode system. The approximation error, based on the Lipschitz constant, can then be compensated for by adding a disturbance to the system dynamics, similar to the method proposed by Asarin et al. (2003). Then the PRP can be



(a)



(b)

Fig. 6. Result of (a) the PRP with a liberal merging condition and (b) the final step in the refinement procedure for the servo system example. Note that the reachset no longer intersects $Fail$.

applied to the system to compute a conservative approximation of the reachable set.

REFERENCES

R. Alur, T.A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. *IEEE Trans. on Software Engineering*, 22(3):181–201, Mar 1996.

E. Asarin, T. Dang, and A. Girard. Reachability analysis of nonlinear systems using conservative approximations. In *Hybrid Systems: Computation and Control 2003*, pages 20–35. Springer-Verlag, 2003.

F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 1999.

S. Boyd, L.E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *SIAM Studies in Applied Mathematics*. SIAM, 1994.

A. Chutinan and B. Krogh. Computational techniques for hybrid system verification. *IEEE Transactions on Automatic Control*, 48(1):64–75, Jan. 2003.

J.E.R. Cury, B.H. Krogh, and T. Niinomi. Synthesis of supervisory controllers for hybrid systems based on approximating automata. *IEEE Transaction on Automatic Control*, 43(4):564–569, April 1998. a preliminary version of this paper also appeared in *Proc. 34th IEEE CDC*, pp. 1461-1466.

T. Dang and O. Maler. Reachability via face lifting. In *Hybrid Systems: Computation and Control 1998*, pages 96–109. Springer-Verlag, 1998.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Press, Baltimore, 1996.

T. A. Henzinger, P. Ho, and H. Wong-Toi. Hytech:a model checker for hybrid systems. In *Software Tool for Technology Transfer 1*, pages 110–122. Springer-Verlag, 1997.

U. Jönsson. Robustness of trajectories with finite time extent. *Automatica*, 38:1485–1497, 2002.

J. Kapinski. *Control System Verification and Synthesis Using Proximity Based Automata*. PhD thesis, Carnegie Mellon University, 2004.

J. Kapinski, O. Maler, O. Stursberg, and B. H. Krogh. On systematic simulation of open continuous systems. In *Hybrid Systems: Computation and Control, 6th International Workshop, HSCC'03*, pages 283–297. Springer-Verlag, 2003.

A. B. Kurzhanski and I. Vályi. *Ellipsoidal Calculus for Estimation and Control*. Birkhä user, Boston, 1997.

I. Mitchell, A. Bayen, and C. Tomlin. Computing reachable sets for continuous dynamic games using level set methods. *IEEE Trans. on Automatic Control*, to appear.

T. Moor, J. Raisch, and S.D. O'Young. Discrete supervisory control of hybrid systems based on l-complete approximations. *Journal of Discrete Event Dynamic Systems*, pages 83–107, 2004.

J. Raisch and S. D. O'Young. Discrete approximation and supervisory control of continuous systems. *IEEE Transactions on Automatic Control*, 43(4):569–573, April 1998.

K. Schmidt, J. Kapinski, and B. H. Krogh. Control input synthesis for hybrid systems using informed search. In *Computer Aided Control System Design Conference (to appear)*, 2004.

B.I. Silva, O. Stursberg B.H. Krogh, and S. Engell. An assessment of the current status of algorithmic approaches to the verification of hybrid systems. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 3, pages 2867–2874, 2002.

L. Vandenberghe, S. Boyd, and W. Shao-Po. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, April 1998.