ROBUST SIMULATION OF NONLINEAR ELECTRONIC CIRCUITS

A. Bemporad[†], *L. Giovanardi*[‡], *F.D. Torrisi*[†]

[†]Automatic Control Laboratory
Swiss Federal Institute of Technology
ETH Zentrum, ETL I 26
8092 Zurich, Switzerland
bemporad,torrisi@aut.ee.ethz.ch

[‡]Dip. di Sistemi e Informatica
UniversitaÁdegli Studi di Firenze
Via di S.Marta, 3
50139 Firenze, Italy
giovanardi@dsi.unifi.it

Abstract— *This paper proposes robust simulation of piecewise linear systems as a tool for the analysis of nonlinear electronic circuits. Rather than computing the evolution of a single trajectory, robust simulation computes the evolution from a set of initial conditions in the state space, for all forcing input signals within a given class. We describe here a tool to perform this analysis using mathematical programming. Among various applications, the tool allows to estimate the domain of attraction of equilibria, and to determine if some design speci£cations — expressed themselves in terms of reachability of subsets of the state-space — are met. A test of the tool on Chua's circuit is presented.*

I. Introduction

In the last decades several tools and packages have been developed for simulating nonlinear electronic circuits, based on research efforts in both modeling and numerical integration techniques. For a *particular* initial electrical condition of the circuit and a given input signal, simulation computes the expected behavior of voltages and currents, which allows to check if the circuit satis£es the design speci£cations.

On the other hand, in many situations is important to know if the speci£cations are met for a whole *set* of initial conditions and forcing input signals. For linear circuits, classical frequency domain analysis provides a full answer. For nonlinear circuits, such techniques are not directly applicable.

An approximate answer can be given by gridding the set of initial conditions and input signals, and by running a large number of simulations, although (i) in nonlinear circuits is not obvious how to sample such sets, (ii) some critical evolution might be overlooked, and (iii) the less the desired coarseness of results, the more the simulation effort.

As an alternative tool, in this paper we propose *robust simulation*: given a set of initial conditions and a class of input signals, we determine the time evolutions of *all* the resulting voltages and currents on the circuit. This type of analysis is also called *reachability analysis* in systems theory, and *formal veri£cation* in real-time and embedded systems engineering [1]. The main applications of robust simulation are the assessment of the satisfaction of design speci£cations, for instance guaranteeing that certain maximum current levels are never exceeded, and the approximation of the domain of attraction of certain DC equilibrium conditions.

Mainly based on a *piecewise linear* (PWL) description of the circuit, several techniques have been proposed for determining the DC solutions of nonlinear circuits [2]. PWL systems are in fact a natural way of modeling nonlinear circuits. More generally, PWL systems can model a large number of physical processes, from systems with static nonlinearities (e.g., nonlinear resistors), to nonlinear dynamics approximated with arbitrary accuracy via multiple linearizations at different operating points. PWL systems are also known in the control and software engineering literature as *hybrid systems*, where the term hybrid stems from the interaction between continuous dynamics and logic (i.e., switches). PWL systems have been also successfully applied for stability analysis of cellular neural networks [3].

This paper proposes the reachability analysis tool developed in [4], [5] and applied in [6] for veri£cation of safety requirements in process control systems, to perform robust simulation of PWL circuits. The tool is based on a discrete-time description of the circuit and relies on mathematical programming techniques.

II. PWL Systems

Switched systems [7], [8], [9] are de£ned by a partition of the state space into polyhedral regions, each one being associated to a different continuous dynamics:

$$\Sigma : \begin{cases} x(t+1) = A_i x(t) + B_i u(t) + f_i \\ \text{for} \quad x(t) \in \mathcal{C}_i \triangleq \{x : H_i x \le K_i\} \end{cases} \quad (1)$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$, $u \in \mathbb{R}^m$, $\{\mathcal{C}_i\}_{i=0}^{s-1}$ is a polyhedral partition of the sets of states $\mathcal{X}$, and f_i is a constant vector. For instance, a simple circuit composed of the series of a voltage source $u(t)$, a reactive component, and a piecewise linear resistor can be rewritten in the form (1) by writing one Kirchhoff's voltage law for each linear piece of the characteristics of the resistor.

A *trajectory* is the collection of vectors $\{x(0), \dots, x(t), \dots\}$ satisfying the difference equation (1). Without additional hypotheses on continuity of the piecewise af£ne state-update mapping, system (1) is not well posed in general, as the state-update function is twice (or more times) de£ned over common boundaries of sets $\mathcal{C}_i$ (the boundaries will be also referred to as *guardlines*). This is a technical issue which can be avoided as in [9].

In [10] the authors show that PWL systems are equivalent to the *mixed logical dynamical* (MLD) systems introduced in [4]. These are hybrid systems de£ned by the interaction of logic, £nite state machines, and linear discrete-time systems. The MLD form is based on the idea of transforming logic relations into mixed-integer linear inequalities [11], [12].

III. Problem Definition

A. Robust Simulation

Consider a PWL system (1), a polyhedral set of initial conditions $\mathcal{X}(0)$, a bounded polyhedral set of forcing input signals $\mathcal{U}$, and a time horizon T. *Robust simulation* amounts to £nding the *reach set* $\mathcal{X}(t, \mathcal{X}(0))$, namely the set of states reached at time t starting from any $x \in \mathcal{X}(0)$ and by applying any input $u(k) \in \mathcal{U}$, $k \le t-1$, or, in other words, all the possible evolutions at time t, for all $t \le T$. The term *robust simulation* was also adopted in [13], where simulation of entire set evolutions was proposed for stability and performance analysis of nonlinear control systems.

B. Reachability Analysis

Given a collection of disjoint polyhedral *target sets* $\mathcal{Z}_1, \mathcal{Z}_2, \dots, \mathcal{Z}_L$, *reachability analysis* amounts to determine (*i*) if $\mathcal{Z}_j$ is reachable from $\mathcal{X}(0)$ within $t \le T$ steps for some input sequence $\{u(0), \dots, u(t-1)\} \subseteq \mathcal{U}$; (*ii*) if yes, the subset of initial conditions $\mathcal{X}_{\mathcal{Z}_j}(0)$ of $\mathcal{X}(0)$ from which $\mathcal{Z}_j$ can be reached within T steps; (*iii*) for any $x_1 \in \mathcal{X}_{\mathcal{Z}_j}(0)$ and $x_2 \in \mathcal{Z}_j$, the input sequence $\{u(0), \dots, u(t-1)\} \subseteq \mathcal{U}$, $t \le T$, which drives x_1 to x_2.

Subsets of $\mathcal{X}(0)$ leading to none of the target sets $\mathcal{Z}_j$ will be labeled as *non-classi£able in T steps*. Typically, non-classi£able subsets shrink and eventually disappear for increasing T.

In a typical application $\mathcal{Z}_1, \dots, \mathcal{Z}_L$ represent sets of infeasible states leading to dangerous conditions (for instance, an excessive current in a component, or an excessive voltage at the terminals of a capacitor). In this case, the PWL model (1) is no longer valid, namely there is an empty intersection between the set $\mathcal{C}_i$ and the set of critical conditions. By letting $\mathcal{X}_{\text{inf}} \triangleq \mathbb{R}^n \setminus \bigcup_{i=0}^{s-1} \mathcal{C}_i$, we give the following

De£nition 1: The set $\mathcal{X}(0) \subseteq \mathbb{R}^n$ of initial conditions is said to belong to the *domain of infeasibility in T steps* $\mathcal{I}_T(0)$ if $\forall x(0) \in \mathcal{X}(0)$ there exists t, $0 \le t \le T$ such that $x(t) \in \mathcal{X}_{\text{inf}}$.

The problem is equivalent to *verify* that for all initial conditions and input the system behaves in a *safe*, that is, to what is called *formal veri£cation of safety* in the software engineering language.

C. Stability Characterization

Given a stable DC solution x_e of the PWL system (1), the *stability characterization* problem consists of estimating the domain of attraction of x_e, and can be recast as a reachability analysis problem. For simplicity of exposition, assume that the system is autonomous ($B_i = 0$ for all $i = 0, \dots, s-1$)[1], and that x_e belongs to the interior of one of the sets of the partition, say $\mathcal{C}_0$. Denote by $\mathcal{D}_\infty(0)(x_e) \subseteq \mathbb{R}^n$ the (unknown) domain of attraction of x_e. Given an (arbitrarily large) bounded set $\mathcal{X}(0)$ of initial conditions, we want to characterize $\mathcal{D}_\infty(0)(x_e) \bigcap \mathcal{X}(0)$. A necessary condition for x_e to be asymptotically stable is that the matrix A_0 associated with the region $\mathcal{C}_0$ is strictly Hurwitz. Under this assumption, we can compute an invariant set in $\mathcal{C}_0$. In particular, we compute the *maximum output admissible set* (MOAS) $\mathcal{X}_\infty(x_e) \subseteq \mathcal{C}_0$. $\mathcal{X}_\infty(x_e)$ is the largest invariant set contained in $\mathcal{C}_0$, which by [14, Th.4.1] is a polyhedron with a £nite number of facets, and is computed through a £nite number of linear programs (LPs) [14][2]. An estimate of the domain of attraction $\mathcal{D}_\infty(0)(x_e)$ is obtained by

[1]Robust stability questions in the presence of disturbances $u(t) \in \mathcal{U}$, where $\mathcal{U}$ is a given bounded set, can be similarly formulated.

[2]If the effect of perturbations $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$, where $\mathcal{U}$ is a given bounded set of disturbances and $B_0 \neq 0$, has to be taken into account $\mathcal{X}_\infty$ is the largest invariant set under disturbance excitation, and can be computed as proposed in [15].

solving a reachability analysis problem over T steps from $\mathcal{X}(0)$ to $\mathcal{Z}_1 = \mathcal{X}_\infty(x_e)$. This clearly only provides an inner approximation of $\mathcal{D}_\infty(0)(x_e)$, as initial conditions leading to trajectories that enter the region $\mathcal{X}_\infty(x_e)$ in a number of steps larger than T are ruled out. More precisely, we provide the following

De£nition 2: The set $\mathcal{X}(0) \subseteq \mathbb{R}^n$ of initial conditions is said to belong to the *domain of attraction in T steps* $\mathcal{D}_T(0)(x_e)$ if $\forall x(0) \in \mathcal{X}(0)$ the corresponding £nal state $x(T) \in \mathcal{X}_\infty(x_e)$.

Clearly, $\mathcal{D}_T(0) \subseteq \mathcal{D}_{T+1}(0) \subseteq \mathcal{D}_\infty(0)$, and $\mathcal{D}_T(0) \rightarrow \mathcal{D}_\infty(0)$ as $T \rightarrow \infty$. Given a set of initial conditions $\mathcal{X}(0)$, we aim at £nding subsets of $\mathcal{X}(0)$ which are safely asymptotically stable ($\mathcal{X}(0) \bigcap \mathcal{D}_T(0)$), and, eventually, subsets which lead to infeasibility in T steps ($\mathcal{X}(0) \bigcap \mathcal{I}_T(0)$). Subsets of $\mathcal{X}(0)$ leading to none of the two previous cases are again labeled as *non-classi£able in T steps*.

IV. Complexity

The undecidability of reachability analysis in the context of *formal veri£cation* of hybrid automata was proved in [16], [17]. However, the problem stated above is decidable, as we assume a £nite time horizon T and work in discrete time.The reason for focusing on £nite-time reachability is that the time-horizon T has a clear meaning, namely those states which are reachable in more than T steps are in practice unreachable. Nevertheless, the problem is $\mathcal{NP}$-hard. To see this, for simplicity consider that in (1) $f_i = 0$, for all $i = 0, \dots, s-1$, and the system is autonomous ($B_i = 0$ for all $i = 0, \dots, s-1$). Its evolution is

$$x(t) = A_{i(t-1)}A_{i(t-2)} \cdots A_{i(0)}x(0) \tag{2}$$

where $i(k) \in \{0, \dots, s-1\}$ is the index such that $H_{i(k)}x(k) \leq K_{i(k)}$, $k = 0, \dots, t-1$, is satis£ed. The previous questions of reachability can be answered once all the switching sequences $I(t) \triangleq \{i(0), \dots, i(t-1)\}$ leading to $\mathcal{Z}_1$, or $\mathcal{Z}_2$, ..., or $\mathcal{Z}_L$ from $\mathcal{X}(0)$ are known. In fact, it is enough to check that the reach set at each time $t \leq T$,

$$\mathcal{X}(t, \mathcal{X}(0)) \triangleq A_{i(t-1)}A_{i(t-2)} \cdots A_{i(0)}\mathcal{X}(0) \ ,$$

satis£es $\mathcal{X}(t, \mathcal{X}(0)) \cap \mathcal{Z}_j \neq \emptyset$ for all admissible switching sequences $I(T)$. However, the number of all possible switching sequences $I(T)$ is combinatorial with respect to T and s, and any enumeration method would be impractical. Below we recall the veri£cation algorithm proposed in [5] to avoid such an enumeration.

V. Verification Algorithm

In order to determine admissible switching sequences $I(t)$, we need to exploit the special structure of the PWL system (1). This allows an easy computation of the reach set, as long as the evolution remains within a single region $\mathcal{C}_i$. Whenever the reach set crosses a guardline and enters a new region $\mathcal{C}_j$, a new reach set computation based on the j-th linear dynamics is computed, as shown in Fig. 1.

Assume $\mathcal{X}(0) \subset \mathcal{C}_i$ is a convex polyhedral set. Computing the evolution $\mathcal{X}(T, \mathcal{X}(0))$ requires: (*i*) the reach set $\mathcal{X}(t, \mathcal{X}(0), \mathcal{C}_i)$, i.e. the set of evolutions at time t in $\mathcal{C}_i$ from $\mathcal{X}(0)$; (*ii*) crossing detection of the guardlines $\mathcal{P}_h \triangleq \mathcal{X}(t, \mathcal{X}(0), \mathcal{C}_i) \bigcap \mathcal{C}_h \neq \emptyset$, $\forall h = 0, \dots, i-1, i+1, \dots, s-1$; (*iii*) elimination of redundant constraints and approximation of the polyhedral representation of the new regions $\mathcal{P}_h$ (approximation is desirable, as the number of facets of $\mathcal{P}_h$ can grow linearly with time); (*iv*) detection of emptiness of $\mathcal{X}(t, \mathcal{P}_h, \mathcal{C}_i)$ — all the evolutions have crossed the guardlines — and detection of $\mathcal{X}(t, \mathcal{P}_h, \mathcal{C}_i) \subseteq \mathcal{Z}_j$, $j = 1, \dots, L$ (these will be referred to as *fathoming* conditions); (*v*) detection of $\mathcal{X}(t, \mathcal{P}_h, \mathcal{C}_i) \cap \mathcal{Z}_j \neq \emptyset$, $j = 1, \dots, L$ (reachability detection).

Note that in case of stability characterization tasks, the fathoming conditions become $\mathcal{X}(t, \mathcal{P}_h, \mathcal{C}_i) \subseteq \mathcal{X}_\infty$, and detection of full infeasibility $\mathcal{X}(t, \mathcal{P}_h, \mathcal{C}_i) \subseteq \mathcal{X}_{\text{inf}}$.

A. Reach Set Computation

Let the set of initial conditions be de£ned by the polyhedral representation $\mathcal{X}(0) \triangleq \{x : \ S_0 x \leq T_0\}$. The subset $S_i(t, \mathcal{X}(0))$ of $\mathcal{X}(0)$ whose evolution lies in $\mathcal{C}_i$ for t steps is given by

$$S_i(t, \mathcal{X}(0)) = \left\{ x \in \mathbb{R}^n \colon \begin{array}{rcl} S_0 x & \leq & T_0 \\ H_i A_i^k x & \leq & S_i - H_i \sum_{j=0}^{k-1} A_i^j f_i \\ & & k = 0, \dots, t \end{array} \right\} \tag{3}$$

As $S_i(t, \mathcal{X}(0))$ is a polyhedral set, the reach set $\mathcal{X}(t, \mathcal{X}(0), \mathcal{C}_i)$ is a polyhedral set as well. In the presence of input signals, $S_i(t, \mathcal{X}(0)) = \{x \in \mathbb{R}^n : S_0 x \leq T_0, H_i(A_i^k x + \sum_{j=0}^{k-1} A_i^j [B_i u(k-1-j) + f_i]) \leq S_i, \ k = 0, \dots, t\}$, is a polyhedron in the augmented space of tuples $(x, u(0), \dots, u(t-1))$. A compact explicit representation of the set $\mathcal{X}(t, \mathcal{X}(0), \mathcal{C}_i)$ (as inequalities over the £nal state $x(t)$) can be computed by a geometric projection procedure, for which ef£cient tools exist, e.g. [18], although the implicit representation (3) suf£ces for our purposes.

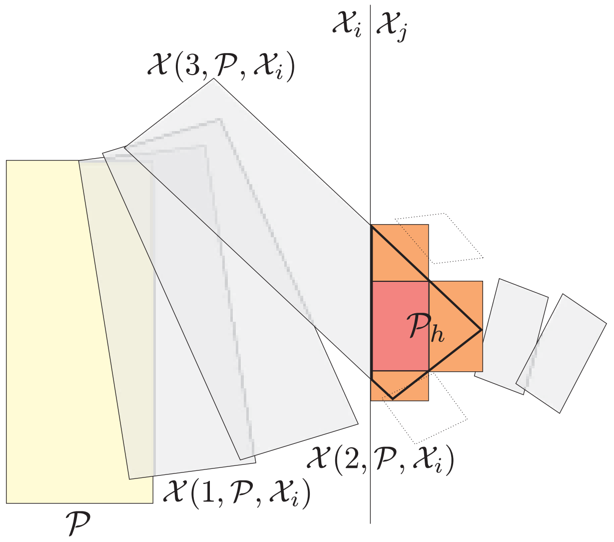

Fig. 1. Reach set evolution, guardline crossing, outer approximation of a new intersection

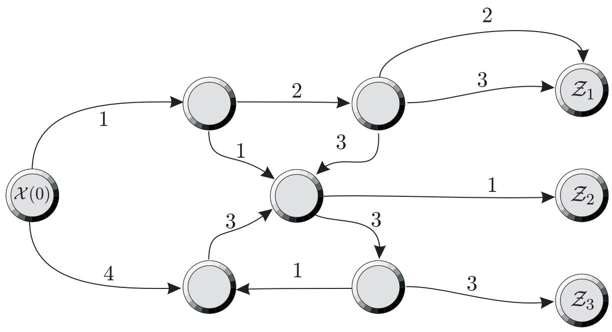

Fig. 2. Graph of evolution G

B. Guardline Crossing Detection

Switching detection amounts to £nding all possible new regions $\mathcal{C}_h$ entered by the reach set at the next time step, i.e. nonempty sets $\mathcal{P}_h \triangleq \mathcal{X}(t,\mathcal{X}(0),\mathcal{C}_i)\bigcap\mathcal{C}_h$, $h \neq i$. Rather than enumerating and checking nonemptiness for all $h = 0,\ldots,i-1, i+1,\ldots,s-1$, in [5] the authors exploit the equivalence between PWL systems and MLD models and solve the switching detection problem via branch and bound methods for mixed-integer linear programming.

C. Approximation of Intersection

The computation of the reach set proceeds in each region $\mathcal{C}_h$ from each new intersection $\mathcal{P}_h$. A new reach set computation is started from $\mathcal{P}_h$, unless $\mathcal{P}_h$ is contained in some larger subset of $\mathcal{C}_h$ which has already been explored. As in principle the number of facets of $\mathcal{P}_h$ grows linearly with time, the algorithm in [5] approximates $\mathcal{P}_h$ as the union of hyper-rectangles, as set inclusion between hyper-rectangles reduces to a simple comparison of the coordinates of the vertices, based on the iterative linear programming method [19].

D. Fathoming

In Sect. V-A we showed how to compute the evolution of the reach set $\mathcal{X}(t,\mathcal{P}_h,\mathcal{C}_i)$ inside a region $\mathcal{C}_i$. The computation is stopped once one of the fathoming conditions listed above happens, namely: (i) the set $\mathcal{X}(t,\mathcal{P}_h,\mathcal{C}_i)$ is empty, which means that the whole evolution has left region $\mathcal{C}_i$; (ii) $\mathcal{X}(t,\mathcal{P}_h,\mathcal{C}_i) \subseteq \mathcal{Z}_j, j = 1,\ldots,L$, i.e., the target set $\mathcal{Z}_j$ has been reached by all possible evolutions from $\mathcal{P}_h$; (iii) $t > T$. The fathoming conditions (i)–(ii) can be easily checked through linear programming.

E. Graph of Evolution

The result of the exploration algorithm detailed in the previous sections can be conveniently stored in a graph G (Fig. 2). The nodes of G represent sets from which a reach set evolution is computed, and an oriented arc of G connects two nodes if a transition exists between the two correspoding sets. Each arc has an associated weight which represents the time-steps needed for the transition. The graph has initially no arc, and nonempty initial set $\mathcal{X}(0)$ and $\mathcal{Z}_j, j = 1,\ldots,L$ as nodes. When a new intersection $\mathcal{X}(t,\mathcal{X}(0),\mathcal{C}_i)\bigcap\mathcal{C}_h$ is detected, it is approximated by a collection of hyper-rectangles, as described in Sect. V-C. Each hyper-rectangle becomes a new node in G, and is connected by a weighted arc from $\mathcal{X}(0)$. In addition, each hyper-rectangle is pushed on a stack of sets to be explored.

When the veri£cation algorithm terminates, the oriented paths on G from initial node $\mathcal{X}(0)$ to terminal nodes $\mathcal{Z}_j, j = 1,\ldots,L$ determine a superset of feasible switching sequences $I(t) = \{i(0),\ldots,i(t-1)\}$. In fact, because of the outer approximation of new intersections $\mathcal{P}_h$, not all switching sequences are feasible. Feasibility can be easily tested via linear programming.

VI. Application Example

As a simple application example we apply the robust simulation procedure described above to Chua's circuit, certainly one among the most thoroughly studied PWL electronic systems. We consider the follow-

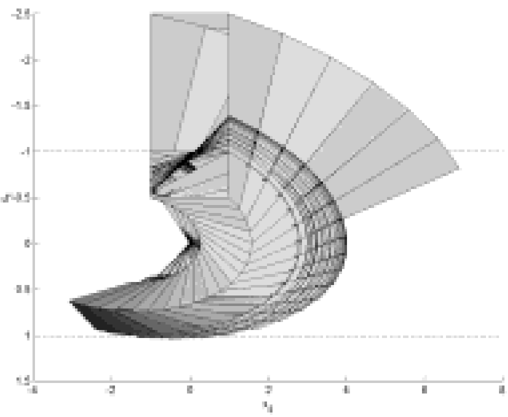

Fig. 3. Set evolution. Dashed lines represent guardlines, darkness level is proportional to evolution time

ing adimensional equations describing the system

$$\begin{cases} \dot{x}_1 &= \alpha(x_2 - x_1 - f(x_1)) \\ \dot{x}_2 &= x_1 - x_2 + x_3 \\ \dot{x}_3 &= -\beta x_2 \end{cases}$$

$$f(x_1) = bx_1 + \tfrac{1}{2}(a-b)[|x_1+1| - |x_1-1|].$$

The system is clearly PWL and has 3 equilibrium points, namely

$$x_0 = [0,0,0]^\top,\ x_{1,2} = \pm[-\xi, 0, \xi]^\top \quad \left(\xi = \tfrac{b-a}{b+1}\right).$$

For the parameter values $\alpha = 5,\ \beta = 100/7,\ a = -8/7,\ b = -5/7$, x_0 is unstable while $x_{1,2}$ are stable ($\xi = 3/2$). We obtain the corresponding discrete-time PWL form (1) by sampling the state-space equations in the three regions with $T_s = 0.25$ s. The choice of this sampling time ensures that the characteristics of the original continuous-time system are preserved.

Fig. 3 shows the x_1-x_2 view of a set evolution for which the initial set is a 2×2 square laying perpendicular to the x_3 axis and centered on x_1, and $T = 20$. The algorithm partitions the evolving set into a collection of (polyhedral) equivalence classes formed by continuous states giving rise to the same switching sequence. Indeed, we note that adjacent set trajectories have different darkness levels, corresponding to different evolution times.

REFERENCES

[1] R. Alur, T.A. Henzinger, and P.-H. Ho. Automatic symbolic veri£cation of embedded systems. *IEEE Trans. on Software Engineering*, 22:181–201, 1996.

[2] S. Pastore and A. Premoli. Polyhedral elements: a new algorithm for capturing all the equilibrium points of piecewise-linear circuits. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 40(2):124–132, February 1993.

[3] M. Di Marco, M. Forti, and A. Tesi. Oscillations in a class of third-order competitive cellular neural networks. In *Proc. 38th IEEE Conf. on Decision and Control*, pages 695–700, 1999.

[4] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[5] A. Bemporad, F.D. Torrisi, and M. Morari. Optimization-based veri£cation and stability characterization of piecewise af£ne and hybrid systems. In B. Krogh and N. Lynch, editors, *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science. Springer Verlag, 2000.

[6] A. Bemporad, F.D. Torrisi, and M. Morari. Veri£cation of mixed logical dynamical models — the batch evaporator process benchmark. Technical Report AUT00-04, Automatic Control Laboratory, ETH Zurich, Switzerland, February 2000. `http://control.ethz.ch/`.

[7] M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Automatic Control*, 43(4):475–482, April 1998.

[8] M. Johannson and A. Rantzer. Computation of piece-wise quadratic Lyapunov functions for hybrid systems. *IEEE Trans. Automatic Control*, 43(4):555–559, 1998.

[9] E. D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Automatic Control*, 26(2):346–358, April 1981.

[10] A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise af£ne and hybrid systems. *IEEE Trans. Automatic Control*, to appear. `http://control.ethz.ch/`.

[11] R. Raman and I. E. Grossmann. Relation between milp modeling and logical inference for chemical process synthesis. *Computers Chem. Engng.*, 15(2):73–84, 1991.

[12] M.L. Tyler and M. Morari. Propositional logic in control and monitoring problems. *Automatica*, 35(4):565–582, 1999.

[13] M. Kantner. Robust stability of piecewise linear discrete time systems. In *Proc. American Contr. Conf.*, pages 1241–1245, Evanston, IL, USA, 1997.

[14] E.G. Gilbert and K. Tin Tan. Linear systems with state and control constraints: the theory and applications of maximal output admissible sets. *IEEE Trans. Automatic Control*, 36(9):1008–1020, 1991.

[15] E.G. Gilbert and I. Kolmanovsky. Maximal output admissible sets for discrete-time systems with disturbance inputs. In *Proc. American Contr. Conf.*, pages 2000–2005, 1995.

[16] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.-H. Ho. Hybrid automata: an algorithmic approach to the speci£cation and veri£cation of hybrid systems. In A.P. Ravn R.L. Grossman, A. Nerode and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer Verlag, 1993.

[17] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration graphs: a class of decidable hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 179–208. Springer Verlag, 1993.

[18] K. Fukuda. *cdd/cdd+ Reference Manual*. Institute for operations Research ETH-Zentrum, ETH-Zentrum, CH-8092 Zurich, Switzerland, 0.61 (cdd) 0.75 (cdd+) edition, December 1997.

[19] A. Bemporad and F.D. Torrisi. Inner and outer approximation of polytopes using hyper-rectangles. Technical Report AUT00-02, Automatic Control Lab, ETH Zurich, 2000.