Overview of complexity and decidability results for three classes of elementary nonlinear systems*

Vincent D. Blondel† John N. Tsitsiklis‡

Abstract

It has become increasingly apparent this last decade that many problems in systems and control are NP-hard and, in some cases, undecidable. The inherent complexity of some of the most elementary problems in systems and control points to the necessity of using alternative approximate techniques to deal with problems that are unsolvable or intractable when exact solutions are sought.

We survey some of the decidability and complexity results available for three classes of discrete time nonlinear systems. In each case, we draw the line between the problems that are unsolvable, those that are NP-hard, and those for which polynomial time algorithms are known.

1 Introduction

We look at the decidability and the complexity of four particular control problem for three different classes of discrete time nonlinear systems. The first two problems that we consider are analysis problems, the other two are control design problems.

State Goes to the Origin
Input: A system $x_{t+1} = f(x_t)$, a state ξ.
Question: Does the initial state $x_0 = \xi$ eventually reach the origin when driven by $x_{t+1} = f(x_t)$?

Stability: All States Go to the Origin
Input: A system $x_{t+1} = f(x_t)$.

*This research was partly carried out while Blondel was visiting Tsitsiklis at MIT and was supported by the NATO under grant CRG-961115.

†Institute of Mathematics, University of Liège B37, B-4000 Liège, Belgium; Email: vblondel@ulg.ac.be

‡Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA; Email: jnt@mit.edu

Question: Do all initial states $x_0 = \xi$ eventually reach the origin when driven by $x_{t+1} = f(x_t)$?

State Can be Driven to the Origin
Input: A system $x_{t+1} = f(x_t, u_t)$, a state ξ.
Question: Does there exists some $k \geq 0$ and controls u_t, $i = 0, \ldots, k-1$ such that the system $x_{t+1} = f(x_t, u_t)$ drives $x_0 = \xi$ to the origin?

Null-controllability. All States Can be Driven to the Origin
Input: A system $x_{t+1} = f(x_t, u_t)$.
Question: Does there exists, associated to every state ξ, some $k \geq 0$ and controls u_t, $i = 0, \ldots, k-1$ such that the system $x_{t+1} = f(x_t, u_t)$ drives $x_0 = \xi$ to the origin?

Asymptotic versions of these definitions are obtained by requiring the sequences to converge to the origin rather than reaching it exactly. The results surveyed in this paper are stated in their non-asymptotic version, most of them remain valid when stated in the asymptotic case.

For linear systems all four questions are decidable and can be decided efficiently (see, e.g., [Sontag, 1990]). On the other hand, no such algorithms exist for general nonlinear systems. Stated at the general level of nonlinear systems, these questions are not interesting because they are far too difficult to solve. For example, as pointed in [Sontag, 1995], the null-controllability question for general nonlinear systems encompasses the problem of solving an arbitrary nonlinear equation. Indeed, for a given function g, consider the system $x_{t+1} = g(u_t)$. Then the system is null-controllable if and only if g has a zero and so the null-controllability question for nonlinear systems is at least as hard as deciding the existence of a zeros for an arbitrary nonlinear functions, which is a far too general problem. For nonlinear control problems to lead to interesting questions we need to constraint the type of nonlinear systems considered.

In the next sections we consider nonlinear systems of the following type: systems with a single nonlinearity, systems of the neural network type, and piecewise-linear systems. In many of these cases control questions become intractable even for systems that are apparently weakly nonlinear. An overview of the results surveyed in this contribution is given in a summarising table.

Before proceeding to the results, let us say a few words on the notions of decidability and computational complexity. When we say that a certain problem is *decidable* we mean that there is an algorithm which, upon input of the data associated to the problem, provides an answer after finitely many steps. The precise definition of *algorithm* is not critical here, it may be, for instance, a Turing machine, an unlimited register machine or any one of most of the other abstract computer models that are proposed in the literature. Most models proposed so far have been shown equivalent from the point of view of their computing capabilities.

When we say that a problem can be *decided in polynomial time*, or that it can be *decided efficiently*, we mean that there is a polynomial P and an algorithm which, upon input of any instance Σ of the problem, provides an answer after at most $P(size(\Sigma))$ computational steps. Again, the precise definition of the size of Σ, and the definition of what is meant by a computational step are not critical. The property of being decidable in polynomial time is robust across all reasonable definitions. The class P is the class of problems that can be decided in polynomial time. The class NP is a class of problems that includes all problems in P and includes a large number of problems of practical interest for which no polynomial time algorithms have yet been found. It is widely believed that $P \neq NP$. A problem is NP-hard if it is at least has hard as any problem in NP. A polynomial time algorithm for an NP-hard problem would immediately result in a polynomial time algorithm for all problems in NP. Finally, a problem is NP-complete if it is NP-hard and belongs to NP. For an introduction to computability, see [Davis, 1982] or [Hopcroft and Ullman, 1969]. For an introduction to computational complexity, see [Garey and Johnson, 1979] or the more recent reference [Papadimitriou, 1994].

This paper is partly based on a survey paper on computational complexity results for systems and control problems [Blondel and Tsitsiklis, 1997c]. A survey of complexity results for nonlinear systems is given in [Sontag, 1995]. See also [Tsitsiklis, 1994].

2 Systems with a single nonlinearity

Let us fix a scalar function $\nu : \mathbf{R} \mapsto \mathbf{R}$. We use the function ν to capture the nonlinearity in a system that has a single nonlinearity. Let $n \geq 1$, $A_0, A_1 \in \mathbf{R}^{n\times n}$, $c \in \mathbf{R}^n$, and consider the system

$$x_{t+1} = \left(A_0 + \nu(c^T x_t) A_1\right) x_t. \tag{1}$$

When ν is constant, the system (1) is linear and its stability can be decided easily. In Theorem 1 in [Blondel and Tsitsiklis, 1997a] the authors show that for most functions ν that are not constant, the stability of systems of the form (1) is NP-hard to decide.

Theorem 1: Let $\nu : \mathbf{R} \mapsto \mathbf{R}$ be a nonconstant scalar function such that

$$\lim_{x\to-\infty} \nu(x) \leq \nu(x) \leq \lim_{x\to+\infty} \nu(x)$$

for all $x \in \mathbf{R}$. Then, Stability of

$$x_{t+1} = \left(A_0 + \nu(c^T x_t) A_1\right) x_t$$

is NP-hard to decide.

Each particular choice of a nonconstant function ν leads to a particular class of nonlinear systems for which stability is NP-hard to decide. In particular, one of the classes is the class of systems that are linear on each side of a hyperplane that divides the state space in two.

Corollary: The problem of deciding, for given matrices A_+, A_- and vector c, whether the system

$$x_{t+1} = \begin{cases} A_+ x_t & \text{when} \quad c^T x_t \geq 0, \\ A_- x_t & \text{when} \quad c^T x_t < 0, \end{cases}$$

is stable, is NP-hard.

A control implication of this result is obtained for linear systems controlled by bang-bang controllers. A linear system $x_{t+1} = Ax_t + Bu_t$ controlled by a bang-bang controller of the type

$$u_k = \begin{cases} K_0 x_t & \text{when} \quad y_t \geq 0, \\ K_1 x_t & \text{when} \quad y_t < 0, \end{cases}$$

leads to a closed-loop system

$$x_k = \begin{cases} (A + BK_0) x_t & \text{when} \quad y_t \geq 0, \\ (A + BK_1) x_t & \text{when} \quad y_t < 0. \end{cases}$$

From Theorem 1 we see that the stability of such systems is NP-hard to decide.

It is not clear when the stability of the systems (1) is actually *decidable*. Except for the trivial case where ν is constant, and the systems are then linear, the authors are not aware of any function ν for which stability of (1) is decidable. For the simple case where ν is piecewise constant, the problem is related to the difficult open problem of deciding the stability of all possible sequences of products of finitely many matrices, see [Blondel and Tsitsiklis, 1997b] for more details.

One can easily adapt the definition (1) to include the possibility of a control action. Let us consider systems of the type

$$x_{t+1} = \left(A_0 + \nu(c^T x_t) A_1\right) x_t + Bu_t. \tag{2}$$

When ν is a constant function, these systems are linear and control questions can be decided easily. When ν is a nonconstant function that satisfies the hypothesis of Theorem 1, it is clear that null-controllability of (2) for $B = 0$ is equivalent to the stability of (1), and so NULL-CONTROLLABILITY of (2) is NP-hard to decide. One can in fact say more than that. We will see in Section 4 that, when ν is a function that has a finite range of cardinality greater or equal to two, then the system (2) becomes piecewise linear and null-controllability of the system is undecidable. The decidability of the case where the range of ν is infinite is open.

3 Systems of the neural network type

Let us fix a scalar function $\sigma : \mathbf{R} \mapsto \mathbf{R}$. Let $n \geq 1$, $A \in \mathbf{R}^{n\times n}$, and consider the system

$$x_{t+1} = \sigma(Ax_t) \tag{3}$$

where σ is defined componentwise, i.e.,

$$\sigma \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix} = \begin{pmatrix} \sigma(q_1) \\ \sigma(q_2) \\ \vdots \\ \sigma(q_n) \end{pmatrix}.$$

Systems of this type arise in a wide variety of situations. The dynamics of (3) depends heavily on the function σ. When σ is linear, the systems are linear and most dynamical properties are easy to check. When σ is the Heaviside function, the entries of the state vector take values in $\{0,1\}$ and the system becomes finite state after the first iteration. The dynamics of such a system, and in fact of any system (3) with a function σ that has finite range, can be modeled by a directed graph whose nodes correspond to the finite states of the system and with directed edges constructed from the matrix A. Dynamical properties for such systems are easy to decide.

Recurrent artificial neural networks are modeled by equations (3) where the function σ is the activation function used in the network (see [Sontag, 1993]). Activation functions that are common in the neural network literature are the *saturated linear function*

$$\sigma(x) = \begin{cases} 0 & \text{when} \quad x \leq 0 \\ x & \text{when} \quad 0 < x < 1 \\ 1 & \text{when} \quad x \geq 1 \end{cases}$$

the standard sigmoid $\sigma(x) = 1/(1+e^x)$, and the inverse trigonometric function $\sigma(x) = \arctan(x)$. All these functions are continuous and have a finite limit on both end of the real axis. These are features that are common in the context of artificial neural networks. Systems (3) with the saturated linear function also arises in the context of linear systems with saturation on the state, and in the analysis and design of fixed-point digital filters (see [Liu and Michel, 1994]for motivations and many references related to filter design). Finally, we also deal with the *cut function* $\sigma(x) = \max(0,x)$ which is probably the simplest piecewise linear function after the linear ones.

Although the difference between the systems (3) and linear systems looks minor when the function σ is weakly nonlinear (such as the cut function for example), the differences in the behavior is complete. In a work announced in [Siegelmann and Sontag, 1991] and completed in [Siegelmann and Sontag, 1995] it is shown that, when σ is the saturated linear function, systems of the type (3) are capable of simulating arbitrary Turing machines. In the simulation, the Turing machine is encoded in the matrix A and the tape content and machine configuration are encoded on some of the states of the system. The simulation of the machine is then obtained by simple iteration. Thus, as computational devices, linear saturated systems are as powerful as Turing machines. The problem of deciding if a given Turing machine halts on some particular tape configuration (the halting problem) is undecidable for Turing machines. Therefore, the problem of deciding if a given initial state of a saturated linear system eventually reaches a state that encodes a halting configuration, is also undecidable. One can show that this halting state can always be chosen to be the origin. And so one conclude (see [Sontag, 1995]for the sketch of a proof).

Theorem 2: State Goes to the Origin is undecidable for saturated linear systems.

By using a universal Turing machine one can in fact prove the stronger result that State Goes to the Origin is undecidable for some *particular* matrix A. There exists a particular matrix A (of size less than 1000×1000 and with integer entries) such that the problem of deciding if a given initial state $x_0 = \xi$ eventually hits the origin when driven by $x_{t+1} = \sigma(Ax_t)$, is undecidable.

The initial result by [Siegelmann and Sontag, 1995] has generated research activity in the direction of finding conditions on the function σ under which Turing machine simulation is possible by systems of the type (3). The fact that such simulations are possible is proved in a very elementary and simple way in [Hyotyniemu, 1997] in the case of the cut function. (Notice that the title of the reference [Hyotyniemu, 1997] involves the term "stability", but this term is actually used in a sense different than the usual notion of stability in systems theory).

In [Koiran, 1996], the author shows how to simulate Turing machines with systems of the type (3) and any function σ that eventually becomes constant on both ends of the real line and is twice differentiable with nonzero derivative on some open interval. The function $\sigma = \arctan$ and other classical function in the neural network literature do not satisfy these hypothesis. Conditions on σ under which systems (3) have Turing power are relaxed in [Kilian and Siegelmann, 1996] where the authors offer a sketch of a proof that Turing machines can be simulated by systems (3) with functions σ that belong to a class that encloses, among others, the functions just described and all the functions that are classically used in neural networks models. The function do not need to become ultimately constant but need to be monotone. Using an argument similar to that used for the case of the saturated linear function one then obtain:

Theorem 3: State Goes to the Origin is undecidable for systems of the type $x_{t+1} = \sigma(Ax_t)$ when σ is the saturated linear function, the cut function, the sigmoid function, the zeroing function and any function that belongs to the classes defined in [Koiran, 1996] and [Kilian and Siegelmann, 1996].

At this point we feel safe to conjecture that, STATE GOES TO THE ORIGIN is undecidable for any function σ that is not linear and that contains an open set in its codomain (the case where σ has finite range is trivially decidable).

From Theorem 3, undecidability of STATE CAN BE DRIVEN TO THE ORIGIN for the controlled system

$$x_{t+1} = \sigma(Ax_t + Bu_t) \tag{4}$$

is immediate to obtain. This result does however not have direct implications for the decidability of null-controllability (ALL STATES CAN BE DRIVEN TO THE ORIGIN) or for the decidability of stability (ALL STATES GO TO THE ORIGIN) of the systems (4) and (3). Despite various attempts and the fact that the undecidability of STABILITYfor saturated linear systems was conjectured in [Sontag, 1995], it is yet unclear whether there exists functions σ for which STABILITY is undecidable. And if one exists, it is not clear if one exists that is continuous. The computational complexity of this problem is also an open question. Although the stability of (3) is strongly suspected to be NP-hard for most function σ, this result was never proved. Let us finally notice that, since undecidability of STABILITY would imply undecidability of NULL-CONTROLLABILITY, the later problem is probably easier to prove undecidable.

4 Piecewise linear systems

Let a finite partition of $\mathbf{R}^n$ be given by $\mathbf{R}^n = H_1 \cup H_2 \cup \cdots \cup H_m$, and suppose that different linear systems are associated to each partition, i.e., the overall nonlinear system is given by

$$x_{t+1} = A_i x_t \qquad \text{when} \qquad x_t \in H_i. \tag{5}$$

When the partitions H_i are definable in terms of a finite number of linear equalities and inequalities, the systems (5) are the *piecewise linear* systems introduced in [Sontag, 1981] as a unifying model for interconnection between automata and linear systems (see [Sontag, 1996] for an updated overview of results available for this model).

Particular classes of piecewise linear systems are obtained from (1) when ν is piecewise constant and from (3) when σ is piecewise linear. Hence, STATE CAN BE DRIVEN TO THE ORIGIN and STATE GOES TO THE ORIGIN are both undecidable and NP-hard for piecewise linear systems. Undecidability of these questions is obtained by using the fact that Turing machines can be simulated by systems of the type (3). These simulations are performed in [Siegelmann and Sontag, 1995] with linear saturated systems of state dimension approximately equal to 1000. In [Koiran *et al.*, 1994], the authors show that similar simulations of Turing machines are possible by iteration of piecewise *affine* systems of state dimension two, or by piecewise linear systems of dimension three. Hence, State Goes to the Origin is undecidable for such systems.

As in the case of systems of the neural network type one can prove a stronger result by using a universal Turing machine. There exist a *particular* piecewise linear system with state dimension three (the system has approximately 800 partitions) such that the problem of deciding for this system if a given initial state $x_0 = \xi$ eventually hits the origin, is undecidable.

Theorem 4: There exist a particular piecewise linear system with state dimension three and with 800 partitions such that State Goes to the Origin is undecidable.

The systems (5) are similar to the *piecewise constant derivative* systems analyzed in [Asarin *et al.*, 1995] and for which analogous undecidability results are available. A piecewise constant derivative system is given by a finite partition $\mathbf{R}^n = H_1 \cup H_2 \cup \cdots \cup H_m$, and by slope vectors b_i for every region H_i of the partition. On any given region of the partition, the state $x(t)$ of the system has a fixed constant derivative,

$$\frac{d\,x(t)}{dt} = b_i \qquad \text{when} \qquad x \in H_i.$$

The trajectories of such systems are continuous broken lines, with breaking points occurring on the boundaries of the regions. In [Asarin *et al.*, 1995] the authors show that, for given states x_b and x_e, the problem of deciding whether x_b is reached by a trajectory starting from x_b, is decidable for systems of dimension two, but is undecidable for systems of dimension three or more.

Suppose now that we add a control to the system and define

$$x_{t+1} = A_i x_t + B u_t \qquad \text{when} \qquad x_t \in H_i. \tag{6}$$

As already explained, it follows from Theorem 4 that State can be Driven to the Origin is undecidable for such systems. This result is also obtained in

[Blondel and Tsitsiklis, 1997a] by using a different proof based on the undecidability of the Post correspondence problem.

Post's correspondence problem.

Instance: A set of pairs of words $\{(U_i, V_i) : i = 1, \ldots, n\}$ over a finite alphabet.

Question: Does there exist a non-empty sequence of indices $i_1, i_2, \ldots, i_k$ where $1 \leq i_j \leq n$, such that $U_{i_1} U_{i_2} \cdots U_{i_k} = V_{i_1} V_{i_2} \cdots V_{i_k}$?

Post's correspondence problem is trivially decidable for one letter alphabets. Furthermore, it is easy to see that the solvability of the problem does not depend on the size of the alphabet, as long as the alphabet contains more than one letter. Post proved that the correspondence problem for an alphabet with more than one letter is undecidable (for a proof of this classical result see [Hopcroft and Ullman, 1969]). In a recent contribution ([Matiyasevich and Sénizergues, 1996]) this result has been improved by showing that the problem remains undecidable in the case where there are only seven pairs of words. On the other hand, the problem is known to be decidable for two pairs of words. The limit between decidability/undecidability is somewhere between three and seven pairs.

There is an obvious trade-off in piecewise linear systems between the state space dimension n and the number of partitions m. When there is only one partition, or when the state dimension is equal to one, State Can be Driven to the Originand Null-controllability are easy to check. The proof technique used in [Blondel and Tsitsiklis, 1997a] is effective for obtaining bounds on n and m for which undecidability is attained. The next result is proved in [Blondel and Tsitsiklis, 1997d].

Theorem 5: Let n_p be any number of pairs of words for which Post's Correspondence Problem is undecidable. Let n be the state space dimension of a piecewise linear system defined on m partitions. If $n \geq 4$, $m \geq 2$ and $nm \geq 2 + 6n_p$, then, State can be driven to the Origin is undecidable.

As mentioned earlier we can take $n_p = 7$, and thus State can be driven to the Origin is undecidable when $nm \geq 44$. In particular, State can be driven to the Origin is undecidable for piecewise linear systems of state dimension 22 and with as few as 2 partitions.

Theorem 5 does not have direct implications for the problems Null-controllability and Stability for which we require certain properties to be shared by *all* states. Piecewise linear systems on two partitions are obtained as special cases of systems with a single nonlinearity. It is therefore clear that Stability and Null-controllability are NP-hard for piecewise linear systems. But that doesn't settle the issue of the decidability of these problems. We now consider these two problems in turn. The first one is undecidable but decidability of the second problem is an unsolved question. The next result is proved in [Blondel and Tsitsiklis, 1997d].

Theorem 6: Let n_p be any number of pairs of words for which Post's correspondence problem is undecidable. Let n be the state space dimension of a piecewise linear system defined on m partitions. If $n \geq 4$, $m \geq 2$ and $nm \geq 26 + 6n_p$, then, Null-controllability, is undecidable.

We finally turn our attention to the decidability of Stability of piecewise linear systems. Consider the particular class of piecewise linear systems in which the partition consists of two regions separated by a hyperplane. The system is

$$x_{t+1} = \begin{cases} A_1 x_t & \text{when} \quad c^T x_t \geq 0 \\ A_2 x_t & \text{when} \quad c^T x_t < 0 \end{cases} \tag{7}$$

Deciding stability of nonlinear systems as simple as (7) is already a nontrivial task. We know that the problem is NP-hard but we do not know if it is decidable. The decidability of this problem is, as we now argue, intimately related to the problem of determining if all possible sequences of products of two given matrices are stable. Let us illustrate this with an example. We build a piecewise linear system with state vector (v_t, y_t, z_t), where v_t and y_t are scalars and z_t is a vector in $\mathbf{R}^n$. The system consists of two linear systems, each of which is enabled in one of two halfspaces, as determined by the sign of y_t

$$\begin{pmatrix} v_{t+1} \\ y_{t+1} \\ z_{t+1} \end{pmatrix} = \begin{pmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & 0 & A_+ \end{pmatrix} \begin{pmatrix} v_t \\ y_t \\ z_t \end{pmatrix} \text{ when } y_t \geq 0,$$

and

$$\begin{pmatrix} v_{t+1} \\ y_{t+1} \\ z_{t+1} \end{pmatrix} = \begin{pmatrix} 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \\ 0 & 0 & A_- \end{pmatrix} \begin{pmatrix} v_t \\ y_t \\ z_t \end{pmatrix} \text{ when } y_t < 0.$$

Let us now look at the evolution of an initial state vector (v_0, y_0, z_0). Suppose that $v_0 = 1$ in which case we have $v_t = 2^{-t}$ for all t. Suppose in addition, that y_0 can take any value in $[-1, 1]$. Then, it is easily seen that y_1 can take any value in $[-1/2, 1/2]$, no matter what was the sign of y_0. Continuing inductively, we see that y_t can take any value in $[-2^{-t}, 2^{-t}]$, can have either sign, and this is independent of the signs of y_s for $s < t$. This shows that every possible sign sequence can be generated by suitable choice of y_0. Hence, the dynamics of the state subvector z_t are of the form $z_{t+1} = A_t z_t$, where A_t is an arbitrary matrix from $\{A_-, A_+\}$. We conclude that the state vector converges to zero, for all possible initial states, if and only if all sequences of products of the matrices A_- and A_+ (taken in an arbitrary order) converge to zero. Thus, a decision algorithm for STABILITY of piecewise linear systems would lead to a test for the stability of all possible sequences of products of two matrices.

5 Summary

AUTONOMOUS SYSTEMS		STABILITY	STATE GOES TO ORIGIN
$x_{t+1} = \left(A_0 + \nu(c^T x_t) A_1\right) x_t$	Complexity	NP-hard for nonconstant ν	?
	Decidability	?	?
$x_{t+1} = \sigma(A x_t)$	Complexity	?	?
	Decidability	Conjectured undecidable	Undecidable for most σ
$x_{t+1} = A_i x_t$ $(x_t \in H_i)$	Complexity	NP-hard	?
	Decidability	?	Undecidable

CONTROLLED SYSTEMS		NULL-CONTROLLABILITY	STATE DRIVEN TO ORIGIN
$x_{t+1} = \left(A_0 + \nu(c^T x_t) A_1\right) x_t + B u_t$	Complexity	NP-hard	?
	Decidability	Undecidable for ν with finite range	Undecidable for ν with finite range
$x_{t+1} = \sigma(A x_t + B u_t)$	Complexity	?	?
	Decidability	?	Undecidable for most σ
$x_{t+1} = A_i x_t + B u_t \ (x_t \in H_i)$	Complexity	NP-hard	?
	Decidability	Undecidable	Undecidable

References

[Asarin *et al.*, 1995] Asarin, A., O. Maler and A. Pnueli (1995). Reachability analysis of dynamical systems having piecewise-constant derivatives, *Theoretical Computer Science*, **138**, 35–66.

[Blondel and Tsitsiklis, 1997a] Blondel, V. D. and J. N. Tsitsiklis (1997). Complexity of elementary hybrid systems, *Proc. of the 4th European Control Conference*, Brussels.

[Blondel and Tsitsiklis, 1997b] Blondel, V. D. and J. N. Tsitsiklis (1997). When is a pair of matrices mortal?, *Information Processing Letters*, **63**, 283-286.

[Blondel and Tsitsiklis, 1997c] Blondel, V. D. and J. N. Tsitsiklis (1997). Survey of complexity results for systems and control problems, (in preparation).

[Blondel and Tsitsiklis, 1997d] Blondel, V. D. and J. N. Tsitsiklis (1997). Decidability limits for low-dimensional piecewise linear systems, (submitted).

[Davis, 1982] Davis, M. (1982). *Computability and Unsolvability*, New York, Dover.

[Garey and Johnson, 1979] Garey, M. R. and D. S. Johnson (1979). *Computers and Intractability : A Guide to the Theory of NP-completeness*, Freeman and Co., New York.

[Hopcroft and Ullman, 1969] Hopcroft, J. E. and J. D. Ullman (1969). *Formal languages and their relation to automata*, Addison-Wesley.

[Hyotyniemu, 1997] Hyotyniemu, H. (1997). On unsolvability of nonlinear system stability, Proc. ECC conference, to appear.

[Kilian and Siegelmann, 1996] Kilian, J. and H. Siegelmann (1996). The dynamic universality of sigmoidal neural networks, *Information and Computation*, **128**, 48-56.

[Koiran, 1996] Koiran, P. (1996). A family of universal recurrent networks, *Theor. Comp. Sciences*, **168**, 473-480.

[Koiran *et al.*, 1994] Koiran, P., M. Cosnard and M. Garzon (1994). Computability properties of low-dimensional dynamical systems, *Theoretical Computer Science*, **132**, 113-128.

[Liu and Michel, 1994] Liu, D. and A. Michel (1994). Dynamical systems with saturation nonlinearities: analysis and design, Springer-Verlag, London, 1994.

[Matiyasevich and Sénizergues, 1996] Matiyasevich, Y. and G. Sénizergues (1996). Decision problem for semi-Thue systems with a few rules, preprint.

[Papadimitriou, 1994] Papadimitriou, C. H. (1994). *Computational complexity*, Addison-Wesley, Reading.

[Siegelmann and Sontag, 1991] Siegelmann, H. T. and E. D. Sontag (1991). Turing computability with neural nets, *Applied Mathematics Letters*, **4** , 77-80.

[Siegelmann and Sontag, 1995] Siegelmann, H. and E. Sontag (1995). On the computational power of neural nets, *J. Comp. Syst. Sci.*, 132–150.

[Sontag, 1981] Sontag, E. (1981). Nonlinear regulation: the piecewise linear approach, *IEEE Trans. Automat. Control*, **26**, 346–358.

[Sontag, 1990] Sontag, E. (1990). *Mathematical control theory*, Springer, New York.

[Sontag, 1993] Sontag, E. (1993). *Neural networks for control* in Essays on Control: Perspectives in the Theory and its Applications (H.L. Trentelman and J.C. Willems, eds.), Birkhauser, Boston, pp. 339-380.

[Sontag, 1995] Sontag, E. (1995). From linear to nonlinear: some complexity comparisons, *Proc. IEEE Conference Decision and Control*, New Orleans, 2916–2920.

[Sontag, 1996] Sontag, E. (1996). Interconnected automata and linear systems: A theoretical framework in discrete-time, in *Hybrid Systems III: Verification and Control* (R. Alur, T. Henzinger, and E.D. Sontag, eds.), Springer, 436–448.

[Tsitsiklis, 1994] Tsitsiklis, J. N. (1994). Complexity theoretic aspects of problems in control theory, *Transactions of the eleventh Army*, ARO Report.