From Linear to Nonlinear: Some Complexity Comparisons

Eduardo Sontag
Dept. of Mathematics
Rutgers University
New Brunswick, NJ 08903

Abstract

This paper deals with the computational complexity, and in some cases undecidability, of several problems in nonlinear control. The objective is to compare the theoretical difficulty of solving such problems to the corresponding problems for linear systems. In particular, the problem of null-controllability for systems with saturations (of a "neural network" type) is mentioned, as well as problems regarding piecewise linear (hybrid) systems. A comparison of accessibility, which can be checked fairly simply by Lie-algebraic methods, and controllability, which is at least NP-hard for bilinear systems, is carried out. Finally, some remarks are given on analog computation in this context.

1. Introduction

It is obvious that many control problems are in general easier to solve for linear systems than for arbitrary, not necessarily linear, ones. An interesting and worthy area of research deals with the attempt to make mathematically precise the increases in difficulty that may arise when passing to the nonlinear case.

By obtaining such precise statements, one gains an understanding of which analysis and/or design problems may be expected to be intractable. For instance, even for apparently mildly nonlinear systems it becomes impossible to check if a state ever reaches the origin. More interestingly perhaps, one also can then explain in what sense some variants of problems are easier than others for nonlinear systems. An example of this later aspect is given by comparing the characterization of the accessibility property (being able to reach a set of states with full degree of freedom) to the characterization of controllability; for some classes of nonlinear systems, these two properties (which coincide for linear systems) diverge in complexity: one can be checked as easily as for linear sytems, but the other becomes intractable.

In this talk, I will briefly touch upon a number of known results on comparing complexity of some control problems for linear and nonlinear systems, mentioning in passing a few open problems/conjectures. I pick simple controllability-type problems to illustrate the issues that arise. Due to space limitations, the discussion will be somewhat informal, but precise bibliographical references are provided to the source of the results, and all details can be found in the respective papers.

*Supported in part by US Air Force Grant AFOSR-94-0293

2. Null-Controllability, Saturated Linear Systems

Assume given a discrete-time control system

$$x(t+1) = P(x(t), u(t)) \qquad (1)$$

with states $x(t)$ evolving in a space $\mathbb{X}$ which contains a special element to be called "0" (typically, of course, $\mathbb{X} = \mathbb{R}^n$ and 0 is the origin). As usual, a state $\xi \in \mathbb{X}$ is *null-controllable* is there is some some nonnegative integer k and some input sequence $u(0), \ldots, u(k-1)$ so that, solving (1) with initial state $x(0) = \xi$, and this particular, input sequence, the solution satisfies $x(k) = 0$. One of the first questions one asks in control problems is if there is some algorithm so that, given a system (1) and a state $x \in \mathbb{X}$, one can determine if this state null-controllable. For linear systems, answering this question is trivial (see below).

Stated in this generality, the null-controllability question is not very interesting. First of all, one may play "logic tricks" with the statement and make the problem too difficult simply by making too abstract the precise meaning of the words "given a system and a state" and allowing an arbitrary recursive-function specification (see for instance the proposed solution in [1] to Arnold's famous open question regarding the analogous problem of asymptotic stability). Second, and more relevant to some of the arguments to be given later in this paper, the problem is not too interesting as posed because it subsumes the general problem of solving an arbitrary nonlinear equation $\phi = 0$: given a map ϕ, just look at the system $x(t+1) = \phi(u(t))$; then ϕ has a zero if and only if 0 (or any other given element of $\mathbb{X}$) is null-controllable. The literature on computational complexity provides an abundant source of examples of classes of maps ϕ (and consequently, classes of systems (1)) for which such problems are hard to solve. (It is often trivial to encode an NP-complete problem such as 3-SAT into problems involving zeroes of functions, for example.)

Far more challenging is to study this property for a class of systems which appears in the control literature and for which the characterization of null-controllability has been of interest, especially if this class superficially seems to be "almost" linear.

One such class, which appears often in many different contexts ranging from actuator saturation to the study of hybrid systems, is given by systems of the following form:

$$x(t+1) \;=\; \text{sat}\;(Ax(t) + Bu(t)) \qquad (2)$$

where the state space $\mathbb{X} = [-1,1]^n$, for some positive integer n, the controls $u(t)$ take values in a space $\mathbb{R}^m$, the matrices A and B have sizes $n \times n$ and $n \times m$ respectively, and sat (z), for a vector z, is the clipped linearity or saturation of each component, that is, the ith component of sat (z) is z_i, the ith component of z, if $|z_i| < 1$, and equals the sign of z_i otherwise. (One could take $\mathbb{X} = \mathbb{R}^n$ instead of $[-1,1]^n$, but nothing essential would change, since after one step all state coordinates have magnitude at most one anyway because of the saturation.) See the recent book [3] for an exposition of many control-theoretic questions for systems of the form (2); see also [16].

Before proceeding further with this class, note for purposes of comparison that if there would be no saturation, we would be studying the standard class of linear systems

$$x(t+1) = Ax(t) + Bu(t) \tag{3}$$

(now with $\mathbb{X} = \mathbb{R}^n$), and for linear systems one can determine null-controllability of a state ξ in a computationally simple manner. Indeed, ξ is null-controllable for (3) if and only if the null-controllability property is verified with $k = n$ (this is a standard elementary fact; see for instance Lemma 3.2.8 in [15]); thus a state ξ is null-controllable if and only if $A^n\xi$ is in the the reachability space of (3), that is, the span of the columns of $B, AB, \ldots, A^{n-1}B$. This property can in turn be checked by Gaussian elimination, so it can be checked in a number of algebraic operations that is polynomial in n and m ("strong polynomial time"). Alternatively, we may ask the question of null-controllability in a bit-computational (Turing-machine) model, assuming that the entries of the matrices A and B, as well as the coordinates of the state ξ, and all rational (as opposed to arbitrary real) numbers, and are each given by specifying pairs of integers in a binary basis, Then the fact is that null-controllability of a state ξ for the system (3) can be checked in a number of elementary Turning-machine steps which is polynomial in the size of the input data, that is, the total number of bits needed to specify A, B, ξ. Thus, the problem is in the class "P" of polynomial-time computable problems. (From now on, I use the Turing machine model, to stay close to classical computational complexity.)

Thus it is natural to ask if adding a saturation can change matters in a fundamental way. The answer is yes. In fact, the change is as big as it could be:

Theorem. *For saturated linear systems* (2)*, the null-controllability question is recursively unsolvable.*

In other words, there is no possible computer program which, when given A, B, ξ with rational entries, can answer after a finite amount of time "yes" if the state ξ is null-controllable for the corresponding system, and "no" otherwise. (In particular, there is no possible characterization in terms of rank conditions, such as was available for linear systems, nor any characterization in terms of checking higher-order algebraic conditions in terms of polynomials constructible from the entries of the matrices and vector in question.) The proof of this fact relies upon the work on simulation of Turing machines by devices such as (2), started in [5] and completed in [6] and [8]. From that work it follows that there exists a certain matrix A (with n approximately equal to 1000 in the construction given in [8], and most entries being 0, 1, or certain small rational numbers) for which there is no possible algorithm that can answer the following question: "Given ξ, is there any integer k so that the first coordinate of the solution of

$$x(t+1) = \text{sat}\ (Ax(t))\ ,\ \ x(0) = \xi \tag{4}$$

has $x_1(k) = 1$?" (Of course, (4) is a particular case of (2), when $B = 0$.) Moreover, the matrix A is built in such a manner that the above property is impossible to check even if ξ is restricted to be a vector with the property that the solution of (4) has $x_1(t) \in \{0,1\}$ for all $t = 0, 1, \ldots$. It is easy to convert the problem "is $x_1(k) = 1$ for some k?" to "is $x(k) = 0$ for some k?" simply by changing each coordinate update equation $x_i(t+1) = \text{sat}\ (\ldots)$ to $x_i(t+1) = \text{sat}\ (\ldots - \alpha x_1(t))$, where α is a positive integer bigger than the possible maximum magnitude of the expression "...". While $x_1(t) = 0$ nothing changes, but if x_1 ever attains the value 1 then the next state is $x = 0$. So the null-controllability question is also undecidable, even in the case in which the system is this one particular system of dimension about 1000 (which in the proof corresponds to a simulation of a universal Turing machine, with the initial condition ξ corresponding to the program for such a machine). This negative result shows that adding a saturation has changed the problem dramatically from the linear case.

One may of course ask about related problems such as observability. For instance, given a system (2) and a linear output map $y = Cx$, one may ask for the decidability of the problem, for a given state ξ: "is ξ indistinguishable from 0?" Again this is essentially trivial for linear systems (just check if ξ is in the kernel of the Kalman observability matrix), but the problem becomes undecidable for saturated systems (take $Cx := x_1$ and use the above construction; as $Cx(t) = x_1(t)$ is always zero or one, distinguishability from zero is equivalent to determining if it is ever one).

It is not yet clear, however, if all reasonable problems for (2) are undecidable. I am willing to conjecture, however, that stabilization problems will be hard. Observe that for linear systems there are low-complexity tests (e.g. Routh-Hurwitz) for asymptotic stability. I believe that it will be impossible to find a test for saturated systems (with no "cheating": data is given as a rational matrix A) and will risk the following:

Conjecture. *To determine the global asymptotic stability of the trivial solution* $x = 0$ *of* (4) *is an undecidable problem.*

3. Piecewise-Linear Systems

The paper [10] proposed the study of general discrete-time interconnections of linear systems and automata, a type of "hybrid" systems that would integrate linear systems theory and some areas of computer science. These "piecewise linear systems" are defined by update equations (1) in which the state space $\mathbb{X}$ and the control-value space are both Euclidean spaces, and the mapping P is given by a finite number of linear functions, each defined on some polyhedron.

Specifically, the idea in that paper was to propose a logic-based approach based on the "language" of "piecewise linear algebra" developed in [11], in such a manner that finite-horizon analysis and design problems can be translated into decision problems for the corresponding language, and more precisely solved by means of quantifier elimination procedures. By "finite-horizon" I mean a bounded-time question such as "is there a control which brings the state ξ into 0 in 78 steps?" in contrast to the unbounded-time problem of null-controllability ("is there a number of steps k and a control so that …?"). Making the number of steps a variable in the problem —that is, in logical terms, "quantifying over" that variable— often makes problems undecidable, as remarked in the previous section for systems (2), which are a very special case of general piecewise linear systems. On the other hand, finite-horizon problems are in principle decidable, as shown in [11, 10]. Thus it is of interest to try to study the computational complexity of such problems. Unfortunately, there is a rather negative result in that regard. To explain this result, given in [12], it is necessary to review some basic terminology from logic. I'll try to do so in very intuitive terms.

Given a fixed piecewise-linear system, a problem like:

"Is there a control u on the interval $[0, 78]$ which steers the state $(-2, 1, 5)$ to $(0, 0, 0)$?"

is said to be a purely-existential problem, or a "$\exists$" problem, for the language of piecewise linear algebra, because it is possible to write a logical formula of the type "there exists u so that $\Phi(u)$" which is true if and only if the property holds (and Φ does not involve any free variables besides u; Φ is simply the set of equations that state that the composition of the dynamics 78 times, using this control, land the state at zero when starting from $(-2, 1, 5)$). Other (still finite-horizon) problems in control cannot be put in $\exists$ form. For instance, to ask if the whole system is controllable to zero in 78 steps would require a formula of $\forall\exists$ type, namely a formula that reads "for all x there exists u such that $\Phi(x, u)$" whose truth is equivalent to null-controllability (and now Φ has the initial state x represented by a variable as well as the control). Another variant appears in design problems. For instance, given a parametric form for a closed-loop controller, say $P(\lambda)$, asking that some value of the parameter result in a feedback law which controls each state to zero in 78 steps would be given by an $\exists\forall$ formula ("there is some parameter λ so that, for each initial state state x, $\Phi(x, u)$"). Even more alternations of quantifiers might appear. For example, in the context of "control Lyapunov functions" one might ask there is a value for a parameter λ so that a scalar "energy" function $V_\lambda(x)$ decreases along suitable trajectories, giving rise to a $\exists\forall\exists$ formula ("there is a λ so that, for each x, there is some u so that either $x = 0$ or $V_\lambda(P(x, u)) < V_\lambda(x)$"). In the same manner, one can define of course $\exists\forall\ldots\exists$ types of problems, for all finite sequences of quantifiers.

Roughly stated, the "polynomial hierarchy" in logic and computer science is obtained in this same way when the basic quantifier-free formulas Φ are propositional formulas, and the variables over which one quantifies are Boolean-valued. Problems are in the class NP (non-deterministic polynomial time) if they can be described by just $\exists$ formulas, and in P (polynomial time) if they can be described with no quantifiers at all. It is widely believed, and one of the most important open problems in theoretical computer science to prove, that the various levels are very different in complexity. Thus, not only should P be different from NP, but problems whose definition requires $\forall\exists$ should be much harder to solve than those in NP, and so forth going "up" along the hierarchy. The result in [12] was that problems in any given level, such as for instance $\exists\forall\exists$, for piecewise linear systems are of the exact same complexity as problems in the corresponding level of the polynomial hierarchy. Thus one has a complete understanding of complexity for such problems modulo the same understanding for the classical hierarchy. Unfortunately, this means that, not only are $\exists$ problems in general NP-hard but more alternation of quantifiers make the problems harder. (On the positive side, advances in understanding of the polynomial hierarchy then allow better understanding of piecewise linear systems.)

4. Continuous-Time Bilinear Systems

When asking about the complexity of controllability questions for continuous time systems one makes contact with a rich area of study. I will now sketch some of the ideas that arise in that context.

One of the fundamental problems for continuous-time systems

$$\dot{x} = f(x, u) \tag{5}$$

is that of finding necessary and sufficient conditions for deciding when a such a system is (locally or globally) controllable. The ultimate goal is to have some type of generalization of the classical Kalman controllability rank condition. An early success of this line of research was achieved with the characterization of the *accessibility property*: there is a Lie-algebraic rank condition for deciding if it is possible to reach a full-dimension set from a given initial state. When this accessibility rank condition does not hold, all trajectories must remain in a lower-dimensional submanifold of the state space. It is known that local controllability can also be *in principle* checked in terms of linear relations between Lie brackets of the vector fields defining the system ([17]), and recent research has succeeded in isolating a number of necessary as well as a number of sufficient explicit conditions for controllability. The literature regarding this question is very large; see for instance [18] and the references there. No complete characterization is yet available, however. In [13], a result was presented which shows that the problem is intrinsically hard from a computational point of view. It is shown that the controllability problem is NP-hard for a class of systems, bilinear systems, for which the accessibility property can be checked in polynomial time. In this sense, accessibility, which for linear systems is equivalent to controllability, is no more complex to check, in a precise quantifiable sense, for nonlinear systems. On the other hand, controllability is harder, also in a precise mathematical measure of difficulty.

To summarize some of the results in [13] and [14] (this latter reference contains some results which were not in the journal paper), consider the class of "polynomial" systems (5), defined as follows. The state-space $\mathbb{X}$ is a nonsingular n-dimensional algebraic subset of some space $\mathbb{R}^N$ (that is, a set with well-defined tangent space at each point, and described by polynomial equations, e.g. the state space used $SO(3)$ used to described orientations of rigid bodies, or Euclidean space $\mathbb{R}^N$ itself), the controls $u(t)$ take values on an Euclidean space $\mathbb{R}^m$, and f is polynomial. An explicit representation of $\mathbb{X}$ by equations is assumed given as part of the data, that is, a set of l polynomials with rational coefficients $\phi_i(x_1, \ldots, x_N)$, $i = 1, \ldots, l$ such that the Jacobian of $(\phi_1, \ldots, \phi_l)'$ has constant rank $N - n$. Let $R(x)$ be the set of states reachable from x. The system Σ is *accessible* if the interior of the set $R(x)$ is nonempty, for each $x \in \mathbb{X}$. It is *controllable* if $R(x) = \mathbb{X}$ for all x.

We remarked in [14]:

Proposition. *Accessibility is decidable for the class of polynomial systems.*

Of course, this begs the question of more accurately characterizing the computational complexity of the problem. In a recent preprint, Gabrielov ([2]), extending work of Risler, showed that it is enough to consider brackets of order at most

$$(n-1)\left(2^{3n-5}3d^3\right)^{2^{n-2}}$$

where d is a bound on the degree of the polynomials defining the system. This helps in bounding the complexity, but the precise characterization seems to be still unknown. The following conjeture was in [14], and, as far as I am aware, is still open:

Conjecture. *Controllability is undecidable for the class of polynomial systems, but the class of all controllable polynomial systems is recursively enumerable.*

It is not difficult to prove the undecidability of restricted versions of this problem, such as asking if an output ever achieves a desired value, by means of an encoding of nonlinear diophantine equations; this is related to papers in the logic literature dealing with the nondecidability of theories involving certain transcendental functions which can be obtained from solving differential equations, but in the above manner the problem seems harder.

To state a somewhat more interesting result, take next the subclass of *bilinear subsystems*, that is, polynomial systems with equations describable as

$$\dot{x} \;=\; \left(A + \sum_{i=1}^{m} u_i G_i\right) x + Bu \;,$$

in terms of rational matrices of appropriate sizes, as before evolving on an algebraic submanifold $\mathbb{X}$. In [13] the following result is shown:

Theorem. *For the class of bilinear subsystems, accessibility can be decided in polynomial time but controllability is NP-hard.*

The proof in [13] used systems with at least two control channels and with unbounded controls. Subsequently, the result was improved so as to show that the problem is hard even for single-input (and bounded control) systems by Kawski in [4].

5. Analog Computation

Finally, I wish to make some comments on the issue of analog computation. The complexity and undecidability results surveyed above are all stated in terms of classical models of computation, that is, Turing machine (serial and digital computing) models. It is a priori conceivable that newer and conceptually different models (neural networks, cellular automata, analog computers of various types, or quantum machines) may be powerful and might allow the solution of analysis and synthesis questions for nonlinear systems, overcoming the current limitations. In the papers [7, 9], however, it is shown that a reasonable model of analog computing (evolving in discrete time, but a similar result is possible in continuous time) is still subject to strong limitations. Roughly speaking, it is shown there that general dynamical systems of the type (1) (assuming that the right-hand side is Lipschitz continuous, which does *not* allow inclusion of Blum-Shub-Smale machines), while far more powerful in principle than Turing machines (because of the possible use of unbounded precision real numbers in computations) cannot compute in polynomial time problems in NP unless the complete polynomial hierarchy collapses to the class $\forall\exists$, a highly unlikely fact according to current knowledge in theoretical computer science. It is interesting, in the context of this paper, that the most general such analog computers can be simulated by saturated systems (2), now using real matrices A and B (with rational matrices one obtains just classical digital computation). This is yet another indication of the richness of that model.

References

[1] da-Costa, N.C.A., and F.A. Doria, "On Arnold's Hilbert symposium problems," in *Computational logic and proof theory (Brno, 1993), Lecture Notes in Comput. Sci., 713*, Springer, Berlin, 1993, pp. 152–158.

[2] Gabrielov, A., "Multiplicities of zeroes of polynomials on trajectories of polynomial vector fields and bounds on degree of nonholonomy," Preprint, Mathematical Sciences Institute, Cornell University, Sept 1994.

[3] Liu, D., and A.N. Michel, *Dynamical Systems with Saturation Nonlinearities: Analysis and Design*, Springer-Verlag, London, New York, 1994.

[4] Kawski, M., "The complexity of deciding controllability," in *Robust Control of Linear Systems and Nonlinear Control* (M.A. Kaashoek, J.H. van Schuppen, and A.C.M. Ran, eds,.) Birkhäuser, Cambridge, MA, 1990.

[5] Siegelmann, H.T., and E.D. Sontag, "Turing computability with neural nets," *Appl. Math. Lett.* **4(6)**(1991): 77-80.

[6] Siegelmann, H.T., and E.D. Sontag, "On the computational power of neural nets," in *Proc. Fifth ACM Workshop on Computational Learning Theory*, Pittsburgh, July 1992, pp. 440-449.

[7] Siegelmann, H.T., and E.D. Sontag, "Some results on computing with "neural nets," *Proc. IEEE Conf. Decision and Control, Tucson, Dec. 1992*, IEEE Publications, 1992, pp. 1476-1481.

[8] Siegelmann, H.T., and E.D. Sontag, "On the computational power of neural nets," *J. Comp. Syst. Sci.* (1995), to appear.

[9] Siegelmann, H.T., and E.D. Sontag, "Analog computation, neural networks, and circuits," *Theor. Comp. Sci.* **131**(1994): 331-360.

[10] Sontag, E.D., "Nonlinear regulation: The piecewise linear approach," *IEEE Trans. Autom. Control* **AC-26**(1981): 346-358.

[11] Sontag, E.D., "Remarks on piecewise-linear algebra," *Pacific J.Math.*, **98**(1982): 183-201.

[12] Sontag, E.D., "Real addition and the polynomial hierarchy," *Inform. Proc. Letters* **20**(1985): 115-120.

[13] Sontag, E.D., "Controllability is harder to decide than accessibility," *SIAM J. Control and Opt.*, **26** (1988): 1106-1118.

[14] Sontag, E.D., "Some complexity questions regarding controllability," *Proc. IEEE Conf. Decision and Control, Austin, Dec. 1988*, pp. 1326-1329.

[15] Sontag, E.D., *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, Springer, New York, 1990.

[16] Sontag, E.D., "Systems combining linearity and saturations, and relations to "neural nets," in *Proc. Nonlinear Control Systems Design Symp., Bordeaux, June 1992* (M. Fliess, Ed.), IFAC Publications, pp. 242-247.

[17] Sussmann, H.J., "Lie brackets, real analyticity, and geometric control," in *Differential Geometric Control Theory* (R.W.Brockett, R.S.Millman, and H.J.Sussmann, eds.), Birkhauser, Boston, 1983.

[18] Sussmann,H.J., "A general theorem on local controllability," *SIAM J. Control and Opt.*, **25**(1987): 158-194.