Proceedings of the American Control Conference
Anchorage, AK May 8-10, 2002

How Hard Is It to Control Switched Systems?

Magnus Egerstedt and Vincent D. Blondel

magnus@ece.gatech.edu
Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

blondel@inma.ucl.ac.be
Department of Mathematical Engineering
Université Catholique de Louvain
B-1348 Louvain-la-Neuve, Belgium

Abstract

We show that the problem of deciding if there exists a control that drives a switched control system between two given states is undecidable. We furthermore investigate what happens if we search for a control that achieves this in a given number of steps, or with a given number of switches. These problems are shown to be respectively NP-complete and NP-hard. The results follow as a consequence of recent complexity results on matrix mortality.

1 Introduction

In this paper, we investigate the computational complexity of controllability questions associated with switched systems. The systems we consider are of the form

$$x_{t+1} = \begin{cases} A_1 x_t + b_1 u_t & \text{if } c^T x_t \geq 0 \\ A_2 x_t + b_2 u_t & \text{if } c^T x_t < 0, \end{cases} \tag{1}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}$, and A_1, A_2, b_1, b_2, c are matrices and vectors of compatible dimensions. It should be noted, already at this point, that the switch conditions are completely characterized by the current state of the system, since the switches take place on the subspace $\{x \in \mathbb{R}^n \mid c^T x = 0\}$.

Such systems have received considerable attention in the hybrid systems literature (see for example [3, 8, 15]). Previous results concerning the computational complexity of such systems, and of related hybrid systems, are surveyed in [2]. (See also [7] for recent results on the controllability and [10] for an accessible survey on the stability of such switched systems.) In [4], the continuous time version of this type of system was studied, and bounds on the minimum energy controller, that drives the system between given points, were derived. The reason why this type of system is relevant is that it models a number of situations where the task is to control a device whose dynamics changes at given parts of the state space. This is the case, for instance, when legged locomotive robotic systems are controlled, where each step-cycle consists of a swing and a stance phase, or when autonomous helicopters make transitions between different flight modes [5]. Another scenario where this problem needs to be solved is in rough terrain path planning applications for mobile robots. The idea is to plan a path, optimal with respect to some given cost functional, in such a way that routing through uneven or slippery environments is penalized, as suggested in [8]. The relevance of the results reported in this paper are to be understood in light of these potential applications.

The paper is organized as follows: In Section 2 we, for the sake of easy reference, very briefly introduce the concepts of undecidability and NP-hardness. In Section 3, we define mortality questions for matrices and show that the problem of deciding whether there exists a control that drives the system (1) from a given state to the origin is undecidable. We then shift focus slightly and ask two different questions. The first one asks whether it is possible to drive the initial state to the origin *in k steps*, which will turn out to be a NP-complete problem. In [14], it was shown that this problem belongs to the class of NP, using the results in [13] for establishing a correspondence between piecewise linear systems and the polynomial hierarchy in logic. However, the NP-completeness result presented in this paper is constructive, and it is based on recent results on matrix mortality [1]. The second question concerns the possibility of going between these states while switching at most k times between the different dynamical regimes. In a number of applications such as mode scheduling for autonomous aerial vehicles, it is desirable to keep the number of switches between different modes to a minimum. The number of switches needed to drive the system between states is thus a natural measure of cost in a number of hybrid control applications, as pointed out in [3, 5, 6]. The problem of finding the minimal number of switches will be found to be NP-hard. (Note that the occurrence of a switch depends solely on the state of the system, which is different from the situation studied in [2].) In a final Section 4, we show that the problem of determining if a given hyperplane can be reached from a given initial state is undecidable as well.

2 Computational Complexity

For the sake of clarity of the presentation, we very briefly introduce the concepts of undecidability and NP-hardness (see for example [11] for an accessible introduction to the subject.) Throughout the paper, we focus our attention on decision problems, i.e., problems where the desired output can be interpreted as a "yes" or a "no". An undecidable problem is a decision problem for which there is no algorithm, defined on a Turing machine or on an equally expressive mode of computation, that always halts with the right answer.

There are many decidable problems of practical interest for which no polynomial time algorithm is known. A decision problem is said to belong to the class NP (nondeterministic polynomial time), which includes all problems of polynomial complexity, if every instance of the problem with a positive answer can be verified in polynomial time. The hardest of all such problems in NP, in the sense that every problem in NP can be reduced to any such problem in polynomial time, are called NP-complete. Any problem in NP can thus be reduced to any of the NP-complete problems in polynomial time. Finally, a problem is said to be NP-hard if it is at least as hard as the NP-complete problems.

3 Controllability to the Origin

In this section we investigate whether or not we can decide if there exists a control that drives the system (1) between desired initial and final states. This question is relevant to any attempt to do path planning for switched systems, as indicated in the introduction.

Theorem 3.1 *The problem of determining for a given system (1) and initial state x_0, if there exists a control that drives x_0 to the origin (state controllability), is undecidable. The problem of determining if there exists a control that drives all initial states to the origin (global controllability) is also undecidable.*

Before we prove this theorem, we recall some notions associated with matrix products. A finite set of real matrices Σ is said to be *mortal* if there exists a finite product of matrices in Σ that is equal to the zero matrix. The set is said to be *length-k-mortal* (for some positive integer k) if the zero matrix can be expressed as a product of length k of matrices in the set. We also say that Σ is *change-k-mortal* if the zero matrix can be expressed as a product of matrices in the set in which there are k changes of matrices. For example, there are 2 changes in the product

$$A_0A_0A_1A_0$$

and 5 in the product

$$A_0A_0A_1A_0A_0A_0A_1A_1A_0A_1.$$

It is quite clear that length-k-mortality is decidable for all possible k since it suffices to compute all products of the given length and to check the presence of the zero matrix. There is no such simple procedure for checking that matrices are change-k-mortal and it is unknown if this problem is decidable. On the other hand, mortality (with no length or change constraints) is known to be undecidable, even for the case of two matrices only, as stated in the following proposition:

Proposition 3.1 (Matrix Mortality) *Mortality of two integer matrices is undecidable.*

The first undecidability proof of mortality for integer matrices is due to Paterson [12]. The proof for two matrices is due to Blondel and Tsitsiklis [1]. With the help of Proposition 3.1 we can now prove Theorem 3.1.

Proof of Theorem 3.1:

Consider the matrix-to-vector bijective mapping $Vec(\cdot) : \mathbb{R}^{n\times n} \to \mathbb{R}^{n^2}$ that arranges the entries of an $n \times n$ matrix in a n^2-vector by taking the rows of the matrix one by one, i.e. if $A = [a_{ij}]$ then

$$Vec(A) = (a_{11}, a_{12}, \cdots, a_{1n}, a_{21}, \cdots, a_{nn})^T.$$

Thus, the entry (i, j) of the matrix A is mapped to the entry $n(i-1)+j$ of the vector $Vec(A)$.

Now, let A and B be $n \times n$ matrices. If we use $A \otimes B$ to denote the Kronecker (tensor) product,

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & & \vdots \\ a_{n1}B & \cdots & a_{nn}B \end{pmatrix},$$

then it is easy to verify that

$$(AB) \otimes I_n = (A \otimes I_n)(B \otimes I_n)$$

and that

$$Vec(AB) = (A \otimes I_n)Vec(B),$$

where I_n is the identity matrix of size n.

If A and B are mortal, i.e. there exists a finite length product S over $\{A, B\}$ such that $S = 0$, e.g.

$$S = ABB \cdots AB = 0,$$

then $(S \otimes I_n = 0)$ as well. The first tensor equality now gives us that $(A \otimes I_n)$ and $(B \otimes I_n)$ are mortal as well.

From the second equality it follows that if there exists a product P of the matrices $(A \otimes I_n)$ and $(B \otimes I_n)$ for which $P\ Vec(A) = 0$, then the matrices A and B are mortal. Combining these two observations we conclude that the following three conditions are equivalent:

1. The matrices A and B are mortal;
2. There exists a product P of the matrices $(A \otimes I_n)$ and $(B \otimes I_n)$ for which $P\ Vec(A) = 0$;
3. The matrices $(A \otimes I_n)$ and $(B \otimes I_n)$ are mortal.

We now transform these conditions into controllability questions for a particular switched system. Consider the switched system

$$x_{t+1} = \begin{cases} A_1 x_t + b_1 u_t & \text{if } c^T x_t \geq 0 \\ A_2 x_t + b_2 u_t & \text{if } c^T x_t < 0, \end{cases}$$

with

$$A_1 = \left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & A \otimes I_n \end{array}\right) \quad A_2 = \left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & B \otimes I_n \end{array}\right)$$

$$b_1 = \begin{pmatrix} 1 \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix} \qquad b_2 = \begin{pmatrix} 1 \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$c = \begin{pmatrix} 1 \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix} \qquad x_0 = \left(\frac{0}{Vec(A)}\right).$$

By constructing the system matrices in this way we directly control which system to use, i.e.

$$x_{t+1} = A_1 x_t + b_1 u_t \text{ if } u_{t-1} \geq 0$$

and

$$x_{t+1} = A_2 x_t + b_2 u_t \text{ if } u_{t-1} < 0.$$

For this system, controllability of x_0 to the origin is equivalent to the existence of a product P of the matrices $(A \otimes I_n)$ and $(B \otimes I_n)$ for which $P\ Vec(A) = 0$, and global controllability is equivalent to mortality of the matrices $(A \otimes I_n)$ and $(B \otimes I_n)$. If we could design an algorithm for one of these problems, we could then also design one for checking mortality of arbitrary matrices. Since this problem is known to be undecidable, the proof follows. ∎

So far we have asked whether there exists a control that drives our piecewise linear control system between boundary states. Consider instead what happens if we ask for the control that drives a state to the origin in a given finite number of steps or with a given finite number of dynamics changes. In terms of the encoding given in the proof of Theorem 3.1, these conditions constrain the type of matrix products we consider. We give below a small adaptation of the statement of Theorem 2 in [1].

Proposition 3.2 (Constrained Matrix Mortality) *The problem of determining if a given pair of Boolean matrices (i.e., matrices with entries in* $\{0,1\}$*) is length-k-mortal is NP-complete. The problem of determining if a given pair of Boolean matrices is change-k-mortal is NP-hard.*

A proof that length-k-mortality is NP-complete is given in [1]. Here we prove that change-k-mortality is NP-hard. The proof given in [1] is by reduction from the NP-complete satisfiability problem SAT, which is the decision problem that investigates if a given a collection of q Boolean expressions (clauses) over p Boolean variables is satisfiable.

Proof of Proposition 3.2:

Starting from an instance of SAT, with p variables and q clauses, the authors in [1] construct two Boolean square matrices of size $(p+1)(q+1)$ that are mortal if and only if the instance of SAT is satisfiable. The construction is such that the matrices are mortal if and only if they are mortal of length $(p+1)(p+3)$.

This result can be directly applied to the change-k-mortality problem, since the previous result implies that if the matrices are mortal then they are also change-k-mortal with $k \leq (p+1)(p+3)$. But if they are change-k-mortal, with $k \leq K$ then they are clearly change-K-mortal as well. Furthermore, if they are change-k-mortal with $k = (p+1)(p+3)$ then they are obviously length-k-mortal as well. From this it follows that they are mortal of length $(p+1)(p+3)$ if and only if they are change-k-mortal with $k = (p+1)(p+3)$.

From this it follows that change-k-mortality is NP-hard, and the proposition follows. ■

Now, combining Proposition 3.2 with the construction given in the proof of Theorem 3.1, we obtain:

Corollary 3.1 *The problem of determining, for a given system (1) and initial state x_0, if there exists a control that drives x_0 to the origin in at most k steps is NP-complete. The problem of determining if there exists a control that drives x_0 to the origin with at most k switches between $c^Tx \geq 0$ and $c^Tx < 0$ is NP-hard.*

This corollary follows if we use the same systems matrices as in the proof of Theorem 3.1. To reach the origin in k steps is thus equivalent to the length-k-mortality of the system matrices. Furthermore, controllability to the origin while crossing the switching surface k times is equivalent to the change-k-mortality of the system matrices.

4 Controllability to a Hyperplane

Theorem 3.1 and Corollary 3.1 are fairly discouraging from a path planning point of view. But, one could ask what would happen if we relax the demand that we must reach a final point. Instead we could be content with reaching a subspace.

However, this question is undecidable as well, as we will see in the following Theorem:

Theorem 4.1 *The problem of determining whether there exist a control u_t that drives the system in Equation 1 between x_0 and a given subspace $d^Tx = 0$ is undecidable.*

In order to prove this result we consider Post's correspondence problem which is one of the classical, undecidable problems. (See for example [9].) Let $\{0,1\}$ be our alphabet, a word is a concatenation of finitely many symbols taken from $\{0,1\}$, e.g. 011011, or the empty word. Now, consider finitely many pairs of such words $(x_1, y_1), \ldots, (x_N, y_N)$. We wish to decide if there exists $m \geq 1$ and a sequence $(i_1, \ldots i_m)$ of integers ($m < \infty$) in the range $1, \ldots, N$ such that $x_{i_1}x_{i_2}\cdots x_{i_m} = y_{i_1}y_{i_2}\cdots y_{i_m}$.

If we let $l(x)$ map a word x to N as $l(x) = 10^{|x|}$, where $|x|$ is the number of symbols in the word ($|\emptyset| = 0$). Then for each pair of words (x_i, y_i) we can construct the matrix

$$M_i = \begin{pmatrix} l(x_i) & 0 & 0 \\ 0 & l(y_i) & 0 \\ x_i & y_i & 1 \end{pmatrix}.$$

Post's correspondence problem is then equivalent to finding a combination of $M_i, \ldots, M_N$ such that

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} M_{i_1} M_{i_2} M_{i_3} \cdots M_{i_m} \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} = 0,$$

which is a quite standard coding of Post's correspondence problem. But even for $N = 7$ this problem is known to be undecidable.

Following the development in [1], we let

$$\hat{A}_1 = \mathrm{diag}(M_1, \ldots, M_N)$$

be the block-diagonal matrix with blocks $M_1, \ldots, M_N$, and let

$$\hat{A}_2 = \left(\begin{array}{c|c} 0 & I_{3(N-1)} \\ \hline I_3 & 0 \end{array} \right).$$

We see that

$$\begin{aligned} \hat{A}_2^{k-1} \hat{A}_1 \hat{A}_2^{N-(k-1)} &= \hat{A}_2^{k-1} \hat{A}_1 \hat{A}_2^{-(k-1)} \\ &= \mathrm{diag}(M_k, \ldots, M_N, M_1, \ldots, M_{k-1}) \\ &= W_k. \end{aligned}$$

Any sequence $S = \hat{A}_2^{t_1} \hat{A}_1^{a_1} \hat{A}_2^{t_2} \cdots \hat{A}_2^{t_q} \hat{A}_1^{a_q} \hat{A}_2^{t_{q+1}}$, where we without loss of generality let $0 \leq t_i \leq N-1$, $i = 1, \ldots, q+1$, can be written as $S = W_{p_1}^{a_1} \cdots W_{p_q}^{a_q} \hat{A}_2^{t_*}$, for some $t_* \geq 0$ and $1 \leq p_i \leq N$, $i = 1, \ldots, q$.

Hence the Post's correspondence problem has a solution if and only if

$$\begin{pmatrix} 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix} S \begin{pmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 0$$

has one.

Proof of Theorem 4.1: We now, in a manner similar to that in the proof of Theorem 3.1, let our system matrices in Equation 1 be

$$A_1 = \left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & \hat{A}_1 \end{array} \right) \quad A_2 = \left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & \hat{A}_2 \end{array} \right)$$

$$b_1 = \begin{pmatrix} 1 \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix} \qquad b_2 = \begin{pmatrix} 1 \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$c = \begin{pmatrix} 1 \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Hence if the problem of driving from $x_0 = (0,1,-1,0,0,\ldots,0)^T$ to $d^Tx = 0$, with

$$d = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}^T$$

would be decidable then Post's correspondence problem would be decidable as well, which concludes the proof. ∎

Acknowledgments

The authors wish to express their gratitude to Roger Brockett for bringing them both to the Division of Engineering and Applied Sciences at Harvard University at the same time, which was instrumental for initiating this work.

References

[1] V.D. Blondel and J.N. Tsitsiklis. When is a Pair of Matrices Mortal? *Information Processing Letters*, Vol. 63, pp. 283–286, 1997.

[2] V.D. Blondel and J.N. Tsitsiklis. A Survey of Computational Complexity Results in Systems and Control. *Automatica*, Vol. 36, pp. 1249–1274, 2000.

[3] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A Unified Framework for Hybrid Control: Model and Optimal Control Theory. *IEEE Transactions on Automatic Control*, Vol. 43, No. 1, pp. 31–45, Jan. 1998.

[4] M. Egerstedt, P. Ögren, O. Shakernia, and J. Lygeros. Toward Optimal Control of Switched Linear Systems. *IEEE Conference on Decision and Control*, Sydney, Australia, Dec. 2000.

[5] M. Egerstedt, J. Koo, F. Hoffmann, and S. Sastry. Path Planning and Flight Controller Scheduling for an Autonomous Helicopter. *LNCS 1569: Hybrid Systems: Computation and Control*, pp. 91–102, Springer-Verlag, The Netherlands, Mar. 1999.

[6] E. Frazzoli, M.A. Dahleh, and E. Feron. A Hybrid Control Architecture for Aggressive Maneuvering of Autonomous Helicopters. In the Proceedings of the *IEEE Conference on Decision and Control*, Phoenix, AZ, Dec. 1999.

[7] L. Gurvits. Stabilities and Controllabilities of Switching Systems, preprint.

[8] S. Hedlund and A. Rantzer. Optimal Control of Hybrid Systems. *Proc. of CDC*, Phoenix, AZ, Dec. 1999.

[9] J.E. Hopcroft and J.D. Ullman. *Formal Languages and Their Relation to Automata*. Addison-Wesley, Reading, MA, 1969.

[10] D. Liberzon and A.S. Morse. Basic Problems in Stability and Design of Switched Systems. *IEEE Control Systems Magazine*, Vol. 19, No. 5, pp. 59-70, Oct. 1999.

[11] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.

[12] M. Paterson. Unsolvability in 3×3 matrices, *Studies in Applied Mathematics*, 49, 105-107, 1970.

[13] E. Sontag. Real Addition and the Polynomial Hierarchy. *Information Processing Letters*, Vol. 20, pp. 115-120, 1985.

[14] E. Sontag. Interconnected Automata and Linear Systems: A Theoretical Framework in Discrete-Time. *Hybrid Systems III: Verification and Control* (R.Alur, T.Henzinger, and E.D.Sontag, eds.), pp. 436-448, Springer, NY, 1996.

[15] C. Tomlin, J. Lygeros, and S. Sastry. A Game Theoretic Approach to Controller Design for Hybrid Systems. *Proceedings of the IEEE*, Vol. 88, No. 7, July 2000.