

# Deterministic and probabilistic algorithms for stabilizing discrete-time switched linear systems $\star$

Atreyee Kundu<sup>a</sup>, Niranjan Balachandran<sup>b</sup>, Debasish Chatterjee<sup>a</sup>

<sup>a</sup>*Systems & Control Engineering, Indian Institute of Technology Bombay, Powai, Mumbai - 400076, India*

<sup>b</sup>*Department of Mathematics, Indian Institute of Technology Bombay, Powai, Mumbai - 400076, India*

---

## Abstract

In this article we study algorithmic synthesis of the class of stabilizing switching signals for discrete-time switched linear systems proposed in [12]. A weighted digraph is associated in a natural way to a switched system, and the switching signal is expressed as an infinite walk on this weighted digraph. We employ graph-theoretic tools and discuss different algorithms for designing walks whose corresponding switching signals satisfy the stabilizing switching conditions proposed in [12]. We also address the issue of how likely/generic it is for a family of systems to admit stabilizing switching signals, and under mild assumptions give sufficient conditions for the same. Our solutions have both deterministic and probabilistic flavours.

**Key words:** Switched systems; Digraph; Azuma's inequality; Algorithmic synthesis.

---

## 1 Introduction

A *switched system* comprises of two components — a family of systems and a switching signal. The *switching signal* selects an active subsystem from the family at every instant of time [18, §1.1.2], and the family of systems may contain nonlinear dynamics, systems with delays, etc. Switching signals are broadly classified as time-dependent (depends only on time), state-dependent (depends on state as well), and with memory (also depends on the history of active subsystems) [19].

In this article we stay within the confines of discrete-time switched linear systems with purely time-dependent switching signals. Stability of such systems has been studied extensively by researchers over the past few decades, see e.g., [19,23,9,18] for detailed surveys. This study can be broadly classified into two categories — stability under *arbitrary switching* and stability under *constrained switching*. In the former category, conditions on the family of systems are identified such that the resulting switched system is

asymptotically stable under all admissible switching signals [6,16,15,14,3,1]; in the latter category, given a family of systems, conditions on the switching signals are identified such that the resulting switched system is asymptotically stable [12,27,19].

Our focus in this article is on stability of discrete-time switched linear systems under constrained switching [12,13]. Stability conditions in this direction primarily rely on the idea of *slow switching* vis-a-vis (*average*) *dwell time* switching. These results were originally developed in the context of continuous-time switched systems, but they can be readily extended to the discrete-time setting [10], [18, Chapter 3], with the (average) dwell time expressed in terms of the number of time steps [19]. This classical theory pertains to switched systems in which all subsystems are stable. In case there are unstable systems, slow switching alone is not sufficient to guarantee stability of the switched system, and additional conditions are required to ensure that the switched system does not spend too much time on the unstable subsystems; see e.g., [19]. Departing from this set of results, in [27] the authors study global exponential stability of a discrete-time switched linear system in which not all subsystems are Schur stable (or possibly no subsystem is Schur stable) but the unstable subsystems form a stable combination. The characterization of the stabilizing switching signal in this new setting involves a modified definition of average dwell time [18, Chapter 3], [27], and the rule of activating the Schur stable subsystems (if any) arbitrarily but activating the unstable subsystems depending

---

$\star$  This paper was not presented at any IFAC meeting. Corresponding author Atreyee Kundu. Tel. +91-22-25764892. Fax +91-22-25720057. N. Balachandran was supported in part by the grant 12IRCCSG016 from IRCC, IIT Bombay. D. Chatterjee was supported in part by the grant 12IRCCSG005 from IRCC, IIT Bombay.

Email addresses: atreyee@sc.iitb.ac.in (Atreyee Kundu), niranj@math.iitb.ac.in (Niranjan Balachandran), dchatter@iitb.ac.in (Debasish Chatterjee).

on a prespecified ratio.

In [12] the authors present a general class of switching signals for global asymptotic stability of switched linear systems. Our conditions involve *only* certain asymptotic properties of the switching signal and do not involve nor imply point-wise bounds on the number of switches — unlike in the case of average dwell time switching. Consequently, there is plenty of flexibility insofar as the transient behaviour of the switching signals are concerned. Although this is not the first instance when unstable systems in the family are considered (e.g., [27]), to the best of our knowledge, this is the first instance when unstable systems in the family are considered *and* the stabilizing conditions depend only on asymptotic behaviour of the switching signal. On the one hand, our conditions require the presence of at least one asymptotically stable system in the family and is therefore “conservative” compared to the ones in the literature that accommodate families with all unstable systems (e.g., [27]). On the other hand, we do not require the unstable systems in the family to form an asymptotically stable combination. The conditions proposed in [12] are also numerically easier to verify than verifying semidefinite programming based conditions (e.g., [16]) on the given family of systems.

In this article we address the important aspect of existence and algorithmic synthesis of the class of stabilizing switching signals proposed in [12]. Given a family of systems, the aspect of algorithmic synthesis of a class of stabilizing switching signals is important for obvious reasons and is not new in the switched systems literature, see e.g., [11], where randomized algorithms to synthesize stabilizing state-dependent switching rules for multimodal systems are presented, and [21], where the authors propose optimization-based methods to verify the average dwell time property. In [12] the authors provide a partial solution towards existence and algorithmic synthesis of the class of stabilizing switching signals under consideration. Here we extend the proposed techniques further and provide an array of results.

The overarching contributions of this article are twofold, the first of which addresses the following pair of questions:

- (a) Given a family of systems, possibly containing unstable dynamics, under what condition does there exist a switching signal that satisfies the conditions proposed in [12, Theorem 1]?
- (b) If there exists such a switching signal, then how to detect/design it algorithmically?
- o Towards answering (a) and (b), we employ graph-theoretic terminology and arguments in our analysis. A weighted digraph is associated to a family of subsystems and the admissible transitions between them, and a switching signal is represented by an infinite walk on the above digraph, both in a natural way; see §2.2 for precise details. We are interested in the class of infinite walks corresponding to the class of stabilizing switching signals proposed in [12]. We shall henceforth freely switch between system-theoretic and the corresponding graph-theoretic terminology in the above sense.

- o We propose sufficient conditions for the desired infinite walk to admit what we call a closed contractive walk, the latter necessarily of finite length, and also discuss algorithmic detection/design of such walks. A closed contractive walk gives rise to an infinite walk corresponding to a stabilizing switching signal in a natural way, as we shall see momentarily.
- o Given a weighted digraph, we provide two necessary and sufficient conditions for the existence of the aforementioned closed contractive walk in terms of a contractive circuit and a contractive cycle. For a contractive circuit we discuss an algorithm based on a linear program to detect such a circuit, and if it exists, we construct it algorithmically. For a contractive cycle we show that it is equivalent to detecting/designing a cycle of “negative weight” and employ standard algorithms for the same. See §3 for details.

The second major contribution of this article is towards posing and answering a more ambitious question:

- (c) What class of switched systems admits the class of switching signals that satisfy the conditions proposed in [12, Theorem 1]?
- o This question necessarily entails a shift in paradigm: while (a) and (b) dealt with conditions for a “given” switched system, (c) seeks to identify a suitable “class of switched systems”. At the level of generality that (c) pertains to, there are two natural candidate apparatuses to turn to for providing its answers — Baire category theory and probability. We turn to the latter, and under mild conditions provide a randomized algorithmic mechanism to identify a class of switched systems that satisfy the conditions proposed in [12, Theorem 1].
- o Our contributions in this direction, presented in §4, may be viewed from the following four perspectives:
  - ◊ Firstly, our algorithm detects cycles on (the underlying digraph of) switched systems that are “typical” with high probability, and under mild conditions we guarantee that such cycles are contractive.
  - ◊ Secondly, the deterministic algorithms that are employed for answers to (a) and (b) above may not be applicable to switched systems whose underlying digraphs are large, especially if their sizes are so large that not all the weights can be kept in memory at once. For such large digraphs our algorithm provides probabilistic guarantees in the spirit of randomized algorithms for detection and design of contractive cycles.
  - ◊ Thirdly, and in cue with the preceding point, our algorithm is of an “online” nature in the following way: starting with a rough probabilistic description of the underlying weighted digraph, (i.e., without knowledge of the precise values of the weights,) we explore the digraph and synthesize a cycle during this exploration that is contractive with high probability. On the other hand, the traditional algorithms for detecting cycles require complete knowledge of the digraph and the vertex and edge weights a priori.
  - ◊ Fourthly, if the constituent subsystems of a switched

systems are prone to evolve over time in a manner that is not precisely known but certain statistical estimates of the nature of evolution are available, our algorithm applies and constructs a contractive cycle with uniform probabilistic guarantees over all such evolutions. See §4 for details.

The remainder of this article is organized as follows: In §2 we briefly recall the class of stabilizing switching signals proposed in [12] and formulate the problems under consideration. Our main results appear in §3 and §4. We provide numerical examples in §5. We conclude in §6 with a brief discussion on future directions. The proofs of our main results are provided in a consolidated fashion in §7.

**Notation.**  $\mathbb{N} = \{1, 2, \dots\}$  is the set of natural numbers,  $\mathbb{N}_0 = \{0\} \cup \mathbb{N}$ , and  $\mathbb{R}$  is the set of real numbers. For a finite set  $M$ ,  $|M|$  denotes its cardinality and  $M \sqcup N$  denotes the disjoint union of  $M$  with another finite set  $N$ . For a digraph  $G(V, E)$  such that  $V := V' \sqcup V''$ ,  $d^+(v)$  denotes the outdegree of a vertex  $v \in V$ ,  $N_{V'}^+ := \{u \in V' \mid (v, u) \in E\}$  denotes the set of outneighbours of a vertex  $v$  in  $V'$ , and  $d_{V'}^+(v) := |N_{V'}^+|$  denotes the outdegree of  $v$  in  $V'$ . For a walk  $W$  on  $G(V, E)$ ,  $|W|$  denotes its length.

## 2 Preliminaries

We consider a family of discrete-time linear systems

$$x(t+1) = A_i x(t), \quad x(0) \text{ given}, \quad i \in \mathcal{P}, \quad t \in \mathbb{N}_0, \quad (1)$$

where  $x(t) \in \mathbb{R}^d$  is the vector of states at time  $t$ ,  $\mathcal{P} = \{1, 2, \dots, N\}$  is a finite index set, and  $A_i \in \mathbb{R}^{d \times d}$  is a known constant full-rank matrix for each  $i \in \mathcal{P}$ . Let  $\sigma : \mathbb{N}_0 \rightarrow \mathcal{P}$  be a *switching signal* that specifies at every time  $t$ , the index of the active subsystem from the family (1). The discrete-time *switched linear system* generated by the given family of systems (1) and the switching signal  $\sigma$  is given by

$$x(t+1) = A_{\sigma(t)} x(t), \quad x(0) \text{ given}, \quad t \in \mathbb{N}_0. \quad (2)$$

Given a family of systems (1), in [12] the authors identify a general class of switching signals  $\sigma$  that admits a crisp characterization under which the resulting switched system (2) is globally asymptotically stable. Recall that by definition, the switched system (2) is *globally asymptotically stable* (GAS) for a given switching signal  $\sigma$  if (2) is

- Lyapunov stable, and
- globally asymptotically convergent:  $\forall x(0), \lim_{t \rightarrow +\infty} x(t) = 0$ .

Preparatory to our results, we begin with:

### 2.1 Properties of the family (1)

Let  $\mathcal{P}_S$  and  $\mathcal{P}_U \subset \mathcal{P}$  be the sets of indices of the asymptotically stable and unstable systems in the family (1), respectively.

**Fact 1 ([12, Fact 1])** *For each  $i \in \mathcal{P}$  there exists a pair  $(P_i, \lambda_i)$ , where  $P_i \in \mathbb{R}^{d \times d}$  is a symmetric and positive definite matrix, and*

- if  $i \in \mathcal{P}_S$ , then  $0 < \lambda_i < 1$ ;
- if  $i \in \mathcal{P}_U$ , then  $\lambda_i > 1$ ;

such that, with  $\mathbb{R}^d \ni \xi \mapsto V_i(\xi) := \xi^\top P_i \xi \geq 0$ , we have  $V_i(\gamma_i(t+1)) \leq \lambda_i V_i(\gamma_i(t))$ ,  $t \in \mathbb{N}_0$ , where  $\gamma_i(\cdot)$  solves the  $i$ -th recursion in (1),  $i \in \mathcal{P}$ . We call  $V_i$ ,  $i \in \mathcal{P}$ , Lyapunov-like functions.

**Fact 2 ([12, Fact 2, Proposition 1])** *There exist numbers  $\mu_{ij} > 0$  such that  $V_j(\xi) \leq \mu_{ij} V_i(\xi)$  for all  $\xi \in \mathbb{R}^d$ , whenever it is admissible to switch from  $i$  to  $j$ ,  $i, j \in \mathcal{P}$ . In particular, the smallest such constants  $\mu_{ij}$  are given by  $\mu_{ij} = \lambda_{\max}(P_j P_i^{-1})$ ,  $i, j \in \mathcal{P}$ , where for a matrix  $M \in \mathbb{R}^{n \times n}$  having real spectrum,  $\lambda_{\max}(M)$  denotes its maximal eigenvalue.*

### 2.2 Properties of the switching signal

We associate a weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$  with a switched system in the following fashion:

- The index set  $\mathcal{P}$  denotes the set of vertices of  $G$ ,
- The set of edges  $E(\mathcal{P})$  of  $G$  consists of:
  - a directed edge from  $i$  to  $j$ ,  $i, j \in \mathcal{P}$ , whenever a switching from system  $i$  to system  $j$  is admissible, and
  - a self-loop at vertex  $j$ ,  $j \in \mathcal{P}$ , whenever it is admissible to dwell on system  $j$  for at least two consecutive time steps.
- $w(i, j) := \ln \mu_{ij}$ ,  $(i, j) \in E(\mathcal{P})$  (à la Fact 2) and  $w(j) := |\ln \lambda_j|$ ,  $j \in \mathcal{P}$ , (à la Fact 1) denote the edge weights and vertex weights of  $G(\mathcal{P}, E(\mathcal{P}))$ , respectively.

We may abbreviate  $G(\mathcal{P}, E(\mathcal{P}))$  by  $G$  if there is no risk of confusion. Recall that a *walk*  $W$  on a digraph  $G(V, E)$  [4, p. 4] is an alternating sequence of vertices and edges  $W = x_0, e_1, x_1, e_2, x_2, \dots, x_{\ell-1}, e_\ell, x_\ell$ , where  $x_i \in V$ ,  $e_i = (x_{i-1}, x_i) \in E$ ,  $0 < i \leq \ell$ . The *initial vertex* of  $W$  is  $x_0$  and the *final vertex* of  $W$  is  $x_\ell$ . If  $x_0 = x_\ell$ , we say that the walk is closed. In this article we follow the convention: A closed walk  $W = x_0, (x_0, x_1), x_1, \dots, x_{\ell-1}, (x_{\ell-1}, x_0), x_0$  is a *circuit* if all its edges are distinct.  $W$  is said to be a *cycle* if the vertices  $x_i$ ,  $0 < i < \ell$  are distinct from each other and  $x_0$ . The length of a walk [4, p. 5] is its number of edges, counting repetitions, e.g., in the above case  $W$  the length of the walk  $W$  is  $\ell$ . In the sequel by the term *infinite walk* we mean a walk of infinite length, i.e., it has infinitely many edges. An *initial subwalk*  $W'$  of a walk  $W$  is an initial segment of  $W$ , which we write as  $W' \leq W$ . The following essentially obvious fact associates a switching signal  $\sigma$  to an infinite walk on the weighted digraph  $G$ :

**Fact 3 ([12, Fact 3])** *The set of switching signals  $\sigma : \mathbb{N}_0 \rightarrow \mathcal{P}$  and the set of infinite walks on  $G(\mathcal{P}, E(\mathcal{P}))$  (defined as above) are in bijective correspondence.*

For a walk  $W$  on  $G$ :

- Let  $N_W$  be the number of distinct vertices that appear in  $W$ , and define

$$\nu(W) := \frac{N_W}{|W|}, \quad |W| > 0 \quad (3)$$

to be the *transition frequency* of  $W$ ;

- we define the function

$$\Xi(W) := \frac{\sum_{(k,\ell) \in E(\mathcal{P})} w(k,\ell) \#\{k \rightarrow \ell\}_W + \sum_{j \in \mathcal{P}_U} w(j) \#\{j\}_W}{\sum_{j \in \mathcal{P}_U} w(j) \#\{j\}_W}, \quad (4)$$

where  $\#\{k \rightarrow \ell\}_W$  and  $\#\{j\}_W$  denote the number of times the edge  $(k, \ell)$  and the vertex  $j$  appear in  $W$ , respectively.

### 2.3 The main result of [12]

In the light of Fact 3 we rephrase [12, Theorem 1] in the following manner:

**Theorem 4 ([12, Theorem 1])** Consider the underlying weighted digraph  $G$  of the switched system (2). The switched system (2) is GAS under all switching signals  $\sigma$ , whose corresponding infinite walks (à la Fact 3)  $W$  satisfy

$$\overline{\lim}_{\substack{|W'| \rightarrow +\infty \\ W' \leq W}} v(W') > 0, \quad (5)$$

and

$$\overline{\lim}_{\substack{|W'| \rightarrow +\infty \\ W' \leq W}} \Xi(W') < 1, \quad (6)$$

where  $v(W')$  and  $\Xi(W')$  are as defined in (3) and (4), respectively.

**Remark 5** Since we are in the discrete-time setting, the association (à la Fact 3) of the length of a walk with time is natural. Condition (5) in the above theorem corresponds to the condition that the switching frequency of  $\sigma$  is not asymptotically vanishingly small [12, Theorem 1]. In the presence of unstable systems in (1), this condition is necessary to ensure that the switched system (2) does not eventually adhere to an unstable system. The first term in the numerator of  $\Xi(W')$  captures the number of times each admissible transition  $(k, \ell) \in E(\mathcal{P})$  occurs in  $\sigma$  till time  $t$ , weighted by  $w(k, \ell) = \ln \mu_{k\ell}$ , where  $\mu_{k\ell}$  is as in Fact 2. The term  $\sum_{j \in \mathcal{P}_S} w(j) \#\{j\}_{W'}$  (and  $\sum_{j \in \mathcal{P}_U} w(j) \#\{j\}_{W'}$ ) captures the number of times system  $j \in \mathcal{P}_S$  (resp.  $\mathcal{P}_U$ ) is activated till time  $|W| = t$  by  $\sigma$ , weighted by  $|\ln \lambda_j|$ , where  $\lambda_j$  obeys Fact 1.

**Remark 6** Of course there is an element of “choice” in the selection of the Lyapunov-like functions in Fact 1 and consequently,  $\lambda_i$ ’s and  $\mu_{ij}$ ’s are not unique. Ideally one would like to algorithmically determine the possibility of co-designing the matrices  $P_i$ ’s and the scalars  $\lambda_i$ ’s such that switching signals satisfying (6) exist, and if so, to construct such a switching signal. This particular co-design problem, to our knowledge, is numerically difficult and in the absence of a numerical solution to it, we consider the matrices  $P_i$ ’s and the scalars  $\lambda_i$ ’s as given, and focus on algorithmic synthesis of switching signals satisfying (6).

Given a family of systems (1), both

- (a) the admissible transitions (connectivity of  $G$ ), and
- (b) the edge and vertex weights of  $G$

play a role in determining whether there exists an infinite walk  $W$  that satisfies (6). Indeed:

**Effect of (a)** Consider  $\mathcal{P} = \{1, 2, 3\}$  with  $A_1 = \begin{pmatrix} 0.2 & 0.4 \\ 0.6 & 0.1 \end{pmatrix}$ ,

$A_2 = \begin{pmatrix} 0.1 & 0.9 \\ 0.8 & 1.0 \end{pmatrix}$  and  $A_3 = \begin{pmatrix} 1.0 & 0.3 \\ 0.7 & 1.2 \end{pmatrix}$ . Consequently,  $\mathcal{P}_S = \{1\}$  and  $\mathcal{P}_U = \{2, 3\}$ . Let  $E(\mathcal{P}) = \{(2, 3), (3, 2)\}$ . In this case, even without the knowledge of the edge and vertex weights, we can conclude that there exists no infinite walk on  $G$  that satisfies condition (6) because the term in the denominator of  $\Xi(W')$ , i.e.,  $\#\{1\}_{W'} = 0$  for all walks on  $G$ .

**Effect of (b)** Consider the given family of systems (1) as in the above case. Assume that  $E(\mathcal{P}) = \{(1, 2), (2, 1)\}$ . From Fact 1 and Fact 2, we have the following estimates:  $\mu_{12} = 0.8878$ ,  $\mu_{21} = 1.7586$ ,  $\lambda_1 = 0.4314$  and  $\lambda_2 = 4.0281$ . The term  $w(1, 2) \#\{1 \rightarrow 2\}_W + w(2, 1) \#\{2 \rightarrow 1\}_W + w(2) \#\{2\}_W = -0.1190 \#\{1 \rightarrow 2\}_W + 0.5645 \#\{2 \rightarrow 1\}_W + 1.3933 \#\{2\}_W$  is greater than the denominator  $w(1) \#\{1\}_W = 0.8407 \#\{1\}_W$  for all walks on  $G$ . Consequently, condition (6) is not satisfied.

In view of the above observations, given the underlying weighted digraph of the switched system (2), we arrive at the important and natural question:

**Problem A** Given a weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$ , does there exist an infinite walk  $W$  on  $G(\mathcal{P}, E(\mathcal{P}))$  that satisfies (6)? If yes, can we provide a mechanism to detect/synthesize it?

In the language of switched systems, of course, Problem A is: “Does there exist a switching signal  $\sigma$  satisfying (6)? If such a  $\sigma$  exists, then can we provide an algorithmic mechanism to detect/synthesize it?”

**Remark 7** It is important to clarify what we mean by algorithmic solutions to Problem A. We provide an algorithm that consists of (i) a finite walk  $W_0$  of length  $n_0 > 0$ , and (ii) an iterative process consisting of a mechanism requiring a bounded quantum of memory, to generate finite walks  $W_k$  of length  $n_k > 0$ ,  $k \in \mathbb{N}$ , satisfying the condition that the final vertex of  $W_{k-1}$  is identical to the initial vertex of  $W_k$  for each  $k$ . We build the infinite walk  $W$  as the limit of  $W_1 W_2 \cdots W_{k-1} W_k$ ,  $k \in \mathbb{N}$ .

Suppose that for a weighted digraph  $G$  there is a self-loop at a vertex  $j \in \mathcal{P}_S$ . Consider a walk  $W$  on  $G$  such that it begins at this vertex  $j$  and keeps on traversing that self-loop repeatedly. In this case  $w(j, j) = 0$  by Fact 2 and no vertex  $j \in \mathcal{P}_U$  is visited. Consequently, the infinite walk generated satisfies (6). Given a weighted digraph  $G$ , the above walk  $W$  can be obtained from an algorithm that detects the above vertex  $j \in \mathcal{P}_S$ .

However, given a weighted digraph  $G$ , detection of an infinite walk  $W$  satisfying (6) is not simple beyond the above trivial case. Indeed, finding an infinite walk on  $G$  that satisfies some prespecified condition involving the vertex and edge weights of  $G$  is a computationally difficult problem. We define

**Definition 8** A walk  $W$  on the weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$  is contractive if

$$\Xi(W) < 1. \quad (7)$$

Given a weighted digraph  $G$ , we provide:

**Solution A** We establish a sufficient condition for the existence of an infinite walk satisfying (6) in terms of a closed contractive walk (necessarily of finite length) on  $G$ . This settles Problem A. We propose algorithmic techniques for synthesis of the above closed walk. Towards this, we derive a set of necessary and sufficient conditions in terms of a contractive *circuit* and a contractive *cycle* for the existence of a closed contractive walk on  $G$ , and apply numerically tractable algorithms to detect/design this circuit and/or cycle on  $G$ .

Moving a step ahead from Problem A, and entailing a shift in paradigm, we pose:

**Problem B** What class of weighted digraphs admits an infinite walk that satisfies (6)?

At the level of abstractness that Problem B pertains to, there are two natural apparatuses to turn to: the first is the Baire Category theorem and its consequences, and the second is probability theory. Considering the ensemble of switched systems as the sample space, we ask how likely is a switched system sampled from this ensemble to admit closed contractive switching signals, and to Problem B we provide:

**Solution B** We propose a polynomial time algorithm that detects a *cycle* of a certain fixed maximal length on  $G$ . Under mild assumptions on the connectivity and the weights associated to the vertices and edges of  $G$  we provide probabilistic guarantee that the above cycle is contractive. Other perspectives and salient features of our algorithm have already been mentioned in the Introduction.

Solution A and Solution B are provided in the following two sections 3 and 4, respectively.

### 3 Solution A

In this section we expose Solution A. Since our solutions must be algorithmic (see Remark 7), we specialize to finitary objects directly in:

**Lemma 9 ([12, Theorem 2(a)])** Consider the underlying weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$  of the switched system (2). If there exists a closed contractive walk  $W$  on  $G(\mathcal{P}, E(\mathcal{P}))$ ,

then the infinite walk obtained by repeating the closed walk  $W$  satisfies (6).

The task of algorithmic detection of a closed contractive walk on  $G$  is computationally simpler under Lemma 9 since the length of the walk is finite.

**Convention 10** The total number of times a closed walk  $W$  visits a vertex  $j \in \mathcal{P}$  is the same as the total number of times  $W$  visits the outgoing edges of the vertex  $j \in \mathcal{P}$ . Consequently, for a vertex  $j \in \mathcal{P}$ ,  $\#\{j\}_W$  can be replaced by  $\sum_{(j,\ell) \in E(\mathcal{P})} \#\{j \rightarrow \ell\}_W$ . Since we are concerned with an infinite walk constructed by repeating the closed contractive walk  $W$  indefinitely many times, the above convention is no loss of generality.

Following Convention 10, the condition (7) becomes

$$\begin{aligned} \bar{\Xi}(W) := \sum_{(k,\ell) \in E(\mathcal{P})} & \left( w(k, \ell) + w(k) \mathbf{1}_{\mathcal{P}_v}(k) \right. \\ & \left. - w(k) \mathbf{1}_{\mathcal{P}_s}(k) \right) \#\{k \rightarrow \ell\}_W < 0. \end{aligned} \quad (8)$$

The mechanism explained in Remark 7 shows that for a walk  $W$  generated by concatenating the walks  $W_1$  and  $W_2$  satisfying the usual contractivity condition, we have  $\bar{\Xi}(W) = \bar{\Xi}(W_1) + \bar{\Xi}(W_2)$ .

However, algorithmic detection of a closed contractive walk on  $G$  is also difficult due to the absence of a bound on the length of the closed walk  $W$ . Consequently, the length at which the algorithm that attempts to detect a closed contractive walk should terminate must be specified and its selection is a difficult task a priori. A natural alternative is to specialize the closed walk  $W$  to a walk of bounded length, for example, a circuit or a cycle. Our first main result provides necessary and sufficient conditions for the existence of a closed contractive walk on  $G$  in terms of a contractive circuit and a contractive cycle:

**Theorem 11** Consider the underlying weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$  of the switched system (2) as discussed in §2. The following are equivalent:

- i)  $G(\mathcal{P}, E(\mathcal{P}))$  admits a closed contractive walk,
- ii)  $G(\mathcal{P}, E(\mathcal{P}))$  admits a closed contractive circuit,
- iii)  $G(\mathcal{P}, E(\mathcal{P}))$  admits a closed contractive cycle.

Consequently, the infinite walk obtained by repeating one of the above satisfies (6).

See §7 for a detailed proof of Theorem 11.

**Remark 12** Theorem 11 gives a set of necessary and sufficient conditions for the existence of a closed contractive walk on a given weighted digraph  $G$ . We now seek algorithms that detect/design a contractive circuit or a contractive cycle on  $G$ . This task is numerically simpler compared to detection/design of a closed contractive walk of some length that is not known a priori for obvious reasons. In the remainder of this section we address these two algorithmic

detection/design issues. An algorithm that detects a contractive circuit on the given weighted digraph  $G$  is discussed in [12, Theorem 2(b) and (c)]. Here we improve upon the above algorithm to find a contractive circuit  $W$  on  $G$  that minimizes  $\bar{\Xi}(W)$ , if one such circuit  $W$  exists. As regard to a contractive cycle, we demonstrate the applicability of the existing algorithms in our context.

### 3.1 Algorithm to detect/design contractive circuits

Given a weighted digraph  $G$ , our algorithm for detection/design of a contractive circuit on  $G$  is motivated by the shortest path algorithm proposed in [22, §3.4].

**Proposition 13 ([12, Theorem 2(b),(c)])** *Let the underlying weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$  of the switched system (2) as defined in §2 be given.*

**Step 1** A contractive circuit  $W$  on  $G(\mathcal{P}, E(\mathcal{P}))$  that minimizes  $\bar{\Xi}(W)$  is obtained from the solution to the following feasibility problem in the variable  $\eta \in \mathbb{R}^{|E(\mathcal{P})|}$ :

$$\begin{aligned} & \text{minimize}_W \quad \bar{\Xi}(W) \\ & \text{subject to} \quad \begin{cases} A\eta = 0 \in \mathbb{R}^{|\mathcal{P}|}, \\ \bar{\Xi}(W) < 0, \\ 0 \leq \eta_j \leq 1 \text{ for all } j = 1, 2, \dots, |E(\mathcal{P})|, \\ \sum_{j=1}^{|E(\mathcal{P})|} \eta_j \geq 1, \end{cases} \end{aligned} \quad (9)$$

where  $A$  is the node (arc) incidence matrix of  $G(\mathcal{P}, E(\mathcal{P}))$ .<sup>1</sup>

**Step 2** From the solution to the feasibility problem (9), a contractive circuit  $W$  on  $G(\mathcal{P}, E(\mathcal{P}))$  that minimizes  $\bar{\Xi}(W)$  can be obtained by the application of Hierholzer's algorithm.<sup>2</sup>

**Remark 14** a) There are two steps to the algorithm in Proposition 13: In Step 1, we employ a feasibility problem (9) to determine a contractive circuit  $W$  on  $G(\mathcal{P}, E(\mathcal{P}))$  that minimizes  $\bar{\Xi}(W)$ , if  $G(\mathcal{P}, E(\mathcal{P}))$  admits such a circuit. The feasibility problem (9) involves solving a linear program for the vector  $\eta$ . Even though (9) is a bona fide linear program, [22, Corollary to Theorem 13.3] guarantees that it has integer optimal solutions. In other words, the condition  $0 \leq \eta_j \leq 1$  for all  $1 \leq j \leq |E(\mathcal{P})|$  implies that each entry of the feasible solution vector  $\eta$  is either 1 or 0, corresponding to whether an edge is included in the circuit or not, respectively. By definition of a circuit, the size of the vector  $\eta$  is

<sup>1</sup> The node (arc) incidence matrix  $A = [a_{ij}]$  [22, §3.4] of  $G$  is defined by

$$a_{ij} = \begin{cases} +1 & \text{if edge } (i, j) \text{ leaves node } i, \quad i = 1, \dots, |\mathcal{P}|, \\ -1 & \text{if edge } (i, j) \text{ enters node } i, \quad j = 1, \dots, |E(\mathcal{P})|, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

<sup>2</sup> See [8, p. 57] for Hierholzer's algorithm.

at most  $|E(\mathcal{P})|$ . If the feasibility problem (9) has a solution, we obtain a subgraph on  $G$  from the vector  $\eta$  that admits a circuit satisfying condition (8). If the feasibility problem (9) has a solution, we proceed to Step 2 and apply Hierholzer's algorithm to find such a circuit on  $G$ . Hierholzer's algorithm admits the above subgraph as input in Step 2. Recall [8, p. 57] that given an Eulerian graph  $G$ , Hierholzer's algorithm finds an Eulerian circuit of  $G$ . The applicability of this algorithm in our context is explained in detail in the proof of Proposition 13 [12, §6.2].

b) The condition  $A\eta = 0 \in \mathbb{R}^{|\mathcal{P}|}$  in the feasibility problem (9) corresponds a circuit, and this equation always has a trivial solution where vector  $\eta$  has all entries equal to 0. The condition  $\sum_{j=1}^{|E(\mathcal{P})|} \eta_j \geq 1$  prevents the above trivial solution.

**Remark 15** With respect to [12, Theorem 2(a) and (b)], we have modified the “objective function” in (9) above. This ensures that the solution to the feasibility problem (9) (if any) corresponds to a contractive circuit on  $G$  that minimizes  $\bar{\Xi}(W)$ .

### 3.2 Algorithms to detect/design contractive cycles

Given a weighted digraph  $G$ , the algorithmic detection/design of a contractive cycle is equivalent to finding what is commonly known as a negative cycle. Indeed, a negative cycle is one for which the sum of the edge weights is less than zero, which is precisely condition (8). A large class of algorithms is available to achieve the above; see e.g., [17,26] for detailed surveys. Perhaps the most well-known one in this class is the Bellman-Ford-Moore algorithm, which is a shortest path algorithm, and it detects and reports negative weight cycles that are reachable from a pre-specified source vertex [5, p. 646]. In our context, the Bellman-Ford-Moore algorithm suffices insofar as the detection of a contractive cycle on  $G$  is concerned; further work is however required for constructing a negative cycle if such a cycle exists. Beyond the Bellman-Ford-Moore algorithm, a newer algorithm proposed in [25] lists all elementary negative cycles on a given weighted digraph; this particular algorithm jointly serves purposes of detection and design of negative cycles.

**Remark 16** Since we are interested in detecting/designing an infinite walk that satisfies (6), algorithmically finding either a contractive circuit or a contractive cycle and designing an infinite walk by repeating the above suffices. However, the question of whether to find a contractive circuit or a contractive cycle algorithmically based on the following aspect appears to be interesting: Consider minimizing  $\bar{\Xi}(W)$  in the sense of maximally negative  $\bar{\Xi}(W)$  over all contractive circuits or cycles  $W$  on a given weighted digraph  $G$ .

- On the one hand, a contractive cycle is necessarily a contractive circuit.<sup>3</sup> On the other hand, although a contrac-

<sup>3</sup> If an edge appears more than once in a walk, the corresponding vertices are also repeated.

tive circuit is not necessarily a contractive cycle, it necessarily contains at least one contractive cycle.<sup>4</sup> Consequently, a contractive circuit  $W$  on  $G$  that minimizes  $\bar{\Xi}(W)$  may contain one or more contractive cycles.

- The algorithm in Proposition 13 detects a contractive circuit  $W$  on  $G$  that minimizes  $\bar{\Xi}(W)$ . If this circuit is only a contractive cycle, then the output of the above algorithm is a contractive cycle as well.
- To obtain a cycle  $W$  on  $G$  that minimizes  $\bar{\Xi}(W)$  by the application of negative cycle detection/design algorithms, a two step mechanism is required: first, to list all elementary negative cycles on  $G$  (by applying the algorithm in [25]) and second, to find the most negative one from this list.
- Instead of applying the above mechanism to  $G$ , we may apply it to the subgraph of  $G$  obtained from the feasibility problem (9) in Proposition 13. Since this subgraph admits a circuit  $W$  that minimizes  $\bar{\Xi}(W)$ , it necessarily admits the most negative cycle on  $G$ . It is immediate that considering a subgraph of  $G$  instead of  $G$  reduces both time and space complexity associated to the search of the most negative cycle.

#### 4 Solution B

We first propose the following algorithm for detection of cycles in  $\mathcal{P}_S$ ; it will be utilized in Theorem 20 below to furnish certain genericity assertions (see Remark 21).

##### Algorithm 1

**Step 1** Set  $k = 0$ .

Pick  $j_k \in \mathcal{P}_S$  uniformly at random.

**Step 2** If  $N_{\mathcal{P}_S}^+(j_k) \setminus \{j_0, \dots, j_k\} \neq \emptyset$ ,

Pick  $j_{k+1} \in N_{\mathcal{P}_S}^+(j_k) \setminus \{j_0, \dots, j_k\}$  uniformly at random.

Set  $k = k + 1$ .

Go to Step 2.

Else

Pick  $j_{k+1} = j_i$  such that  $j_i \in N_{\mathcal{P}_S}^+(j_k)$  and  $(k - i)$  is maximum.

Go to Step 3.

**Step 3** End.

For example, let  $\mathcal{P} = \{1, 2, 3, 4, 5\}$  with  $\mathcal{P}_S = \{1, 2, 3\}$  and  $\mathcal{P}_U = \{4, 5\}$ . Let

$$\begin{aligned} E(\mathcal{P}) = & \{(1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 3), \\ & (2, 4), (2, 5), (3, 2), (3, 4), (3, 5), (4, 1), \\ & (4, 2), (4, 3), (4, 5)\}. \end{aligned}$$

Let  $j_0 = 1 \in \mathcal{P}_S$ . Then  $j_1$  is selected from  $\{2, 3\} \setminus \{1\}$ . Let  $j_1 = 2$ . Then  $j_2$  is selected from  $\{1, 3\} \setminus \{1, 2\}$ . Consequently,  $j_2 = 3$ . Now,  $\{2\} \setminus \{1, 2, 3\} = \emptyset$ . As a result,  $j_3 = 2$ . So we obtain the walk  $1, (1, 2), 2, (2, 3), 3, (3, 2), 2$ , which contains the cycle  $2, (2, 3), 3, (3, 2), 2$ .

<sup>4</sup> This is evident from the proof of Theorem 11.

In the remainder of this section we show that under mild connectivity and generic weight assumptions on the given weighted digraph  $G$ , a cycle obtained from Algorithm 1 satisfies (8) with high probability. Let  $\Phi : \mathbb{N} \rightarrow \mathbb{R}$  be a monotone increasing function.

**Definition 17** A weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$  is said to be

- nicely connected if  $d_{\mathcal{P}_S}^+(j) \geq \lfloor \Phi(|\mathcal{P}_S|) \rfloor$  for all  $j \in \mathcal{P}$ ;
- nicely weighted if the vertex and edge weights on  $G$  satisfy the following conditions:
  - ▷ there exist  $\beta, B > 0$  satisfying  $0 < \beta < B$  such that the vertex weights  $w(j)$  are independent and  $0 < w(j) \leq B$  with  $E[w(j)] = \beta$  for all  $j \in \mathcal{P}$ , and
  - ▷ there exist constants  $A > 0$  and  $\alpha < \beta$  such that for every  $(i, j) \in E(\mathcal{P})$ , the edge weight  $w(i, j) \in [-A, A]$  and  $E[w(i, j)] \leq \alpha$ .

**Remark 18** The condition that the vertex and edge weights  $w(j)$  and  $w(i, j)$  are uniformly bounded if  $G$  is nicely weighted is no loss of generality on account of the graph  $G$  being finite. However, it is also possible to consider the case in which the bounds on the weights depend on the size of the graph  $G$ , as explained in Remark 22 below. We stick to the simpler case for ease of exposition.

The following lemma guarantees the existence of a cycle in  $\mathcal{P}_S$  of length at least  $\lfloor \Phi(|\mathcal{P}_S|) \rfloor$ .

**Lemma 19** If the given weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$  is nicely connected, then Algorithm 1 detects a cycle  $W$  on  $G(\mathcal{P}, E(\mathcal{P}))$  such that all vertices in  $W$  are from  $\mathcal{P}_S$  and the length of  $W$  is at least  $\lfloor \Phi(|\mathcal{P}_S|) \rfloor$ .

See §7 for a short proof of Lemma 19. Our final result is the following:

**Theorem 20** Consider the switched system (2) and the underlying weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$  as described in §2. Suppose that  $G(\mathcal{P}, E(\mathcal{P}))$  is nicely connected and nicely weighted. Then a cycle of length at least  $\lfloor \Phi(|\mathcal{P}_S|) \rfloor$  on  $G(\mathcal{P}, E(\mathcal{P}))$  obtained from Algorithm 1 is contractive with probability at least

$$1 - \exp\left(-\frac{1}{2}\left(\frac{\alpha - \beta}{A + B}\right)^2 \lfloor \Phi(|\mathcal{P}_S|) \rfloor\right).$$

Consequently, the infinite walk obtained by repeating the above cycle satisfies (6).

We present a proof of Theorem 20 in §7.

**Remark 21** Theorem 20 asserts that a cycle obtained via Algorithm 1 is contractive with high probability provided  $|\mathcal{P}_S|$  is large. Consequently, repeating such a cycle derived from Algorithm 1 generates an infinite walk  $W$  that, in view of Lemma 9, satisfies (6). This in turn identifies a class of switched systems (whose underlying weighted directed graph  $G$  is nicely connected and nicely weighted) that admits switching signals satisfying the conditions proposed in [12, Theorem 1] with overwhelming probability.

**Remark 22** The primary engine leading to the estimate in Theorem 20 is Azuma's inequality. Our assumption of

a uniform bound for the weights due to  $G$  being nicely weighted led to a uniform bound on the martingale increments  $(M_m - M_{m-1})_{m=1}^n$  in the proof of Theorem 20, and our estimate followed at once from Azuma's inequality. A more general version of Azuma's inequality may be employed in an identical fashion to cater to vertex- and edge-dependent weights, leading to a possibly sharper bound. The numerical value of the confidence with which a contractive cycle may be found, however, depends on the size of  $\mathcal{P}_S$  and the ability of the function  $\Phi$  in Definition 17 to dominate the accumulation of the weights along the martingale increments.

**Remark 23** Given the underlying weighted digraph  $G$  of the switched system (2), the deterministic algorithms for detection/design of a contractive cycle, as discussed in §3.2, require the complete knowledge of all the vertex and edge weights of  $G$  prior to their application. Consequently, these are ill suited for large graphs where all the weights cannot be stored in the memory at once. In contrast, Algorithm 1 explores  $G$  without prior knowledge of the vertex and edge weights and during this exploration designs a cycle, which is contractive with high probability. In particular, this lends an “online” flavour to our algorithm. Consequently, it is suited for a class of large weighted digraphs for which deterministic guarantees are difficult or impossible to give. In addition, consider the case when certain parameters of the subsystems in the given family (1) (and consequently the vertex and edge weights of  $G$ ) evolve over time in a manner that is not completely known. A cycle obtained from Algorithm 1 is contractive with high probability independent of this evolution as long as  $G$  is nicely weighted.

## 5 Numerical Examples

**Example 24** This example corresponds to our Solution A. Based on our discussion regarding “contractivity” in §3, we select the algorithm in Proposition 13 to find a “most” contractive circuit. We consider the family of systems (1) with  $\mathcal{P} = \{1, 2, 3, 4\}$ , where

$$A_1 = \begin{pmatrix} 0.2 & -0.7 \\ 0.8 & 0.3 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0.5 & 0.1 \\ 0.4 & 0.2 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} 1.2 & 0.9 \\ 1.4 & 0.2 \end{pmatrix}, \quad A_4 = \begin{pmatrix} 1.1 & 0.2 \\ 0.2 & 0.7 \end{pmatrix}.$$

For this family  $\mathcal{P}_S = \{1, 2\}$  and  $\mathcal{P}_U = \{3, 4\}$ . Let all transitions among the systems in the given family be admissible. Let it also be permissible for switching signals to dwell on systems 3 and 4 for two (or more) consecutive time steps. That is,

$$\begin{aligned} E(\mathcal{P}) = \{(1, 2), (1, 3), (1, 4), (2, 1), (2, 3), (2, 4), \\ (3, 1), (3, 2), (3, 3), (3, 4), (4, 1), (4, 2), \\ (4, 3), (4, 4)\}. \end{aligned}$$

We construct the node(arc) incidence matrix  $A$  as described in (10).<sup>5</sup> We avoid presenting the matrix here for reasons of space. The elements of the column vector  $\eta$  are associated with the entries of  $E(\mathcal{P})$ .

For the given family of systems (1), we obtain an estimate for the numbers  $\lambda_i$  and  $\mu_{ij}$  from Fact 1 and Fact 2, respectively.

$$\begin{aligned} \lambda_1 &= 0.6480, & \lambda_2 &= 0.4200, & \lambda_3 &= 4.9946, \\ \lambda_4 &= 3.3657, & \mu_{12} &= 0.6094, & \mu_{13} &= 0.4067, \\ \mu_{14} &= 0.4067, & \mu_{21} &= 2.4470, & \mu_{13} &= 0.9914, \\ \mu_{14} &= 0.9914, & \mu_{31} &= 2.8406, & \mu_{32} &= 1.7241, \\ \mu_{33} &= 1, & \mu_{34} &= 1, & \mu_{41} &= 2.8406, \\ \mu_{42} &= 1.7241, & \mu_{43} &= 1, & \mu_{44} &= 1, \end{aligned}$$

and associate  $|\ln \lambda_j|$  and  $\ln \mu_{ij}$  as vertex weights  $w(j)$  and edge weights  $w(i, j)$  of  $G$ , respectively.

Solving the feasibility problem (9) in the context of this setting with the aid of MATLAB by employing the program YALMIP [20] and the solver SDPT3 [24], we obtain the following solution:

$$\eta = (1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^T,$$

with the corresponding  $\bar{\Xi}(W) = -0.084851 < 0$ . A circuit  $W$  obtained from the vector  $\eta$  with the aid of Hierholzer's algorithm is:

$$1, (1, 2), 2, (2, 1), 1.$$

We consider the switching signal corresponding to the infinite walk obtained by repeating the above circuit  $W$  and study  $(\|x(t)\|)_{t \in \mathbb{N}_0}$  for 100 different initial conditions chosen uniformly at random from the interval  $[-1000, 1000]^2$ .

**Example 25** This example corresponds to Solution B. Consider a nicely connected and nicely weighted directed graph  $G$  with

- o  $|\mathcal{P}_S| = 1000$ ,
- o  $\Phi(r) = \frac{1}{10} \sqrt{r}$ ,
- o  $d^+(j) = \lfloor \Phi(|\mathcal{P}_S|) \rfloor$  for all  $j \in \mathcal{P}$ , and
- o  $A = 2.5$ ,  $B = 5$ ,  $\alpha = 0$  and  $\beta = 2.5$ .

We extract and fix a cycle  $W$  obtained from Algorithm 1 on  $\mathcal{P}_S \subset \mathcal{P}$ . The vertex and edge weights on  $W$  are sampled uniformly at random 1000 times from the intervals as stipulated in Definition 17. We calculate  $X_n$ , as defined in (13) empirically for  $n$  being the length of the cycle  $W$ .

The above experiment is repeated for cycles of different length  $n$  obtained from Algorithm 1 with uniformly randomly selected initial vertex. We plot the empirical probability of  $\{X_n < 0\}$  vs length  $n$  of the cycle in Figure 2. Observe that the detection of a contractive cycle from Algorithm 1

<sup>5</sup> Incidence matrices are in general defined for graphs without self-loops. We accommodate self-loops in an incidence matrix in the following manner: to a vertex  $j \in \mathcal{P}$  such that  $j$  has a self-loop, we associate an auxiliary vertex  $j'$ . The transitions  $j$  to  $j'$  represent the self-loops.

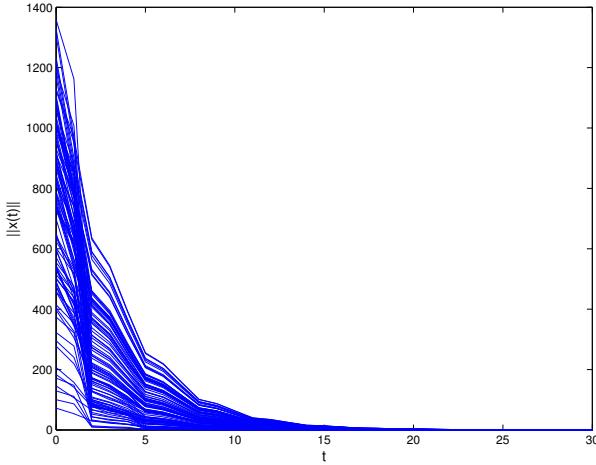


Fig. 1. Plot for  $(\|x(t)\|)_{t \in \mathbb{N}_0}$  for 100 different initial conditions chosen uniformly at random from the interval  $[-1000, 1000]$ .

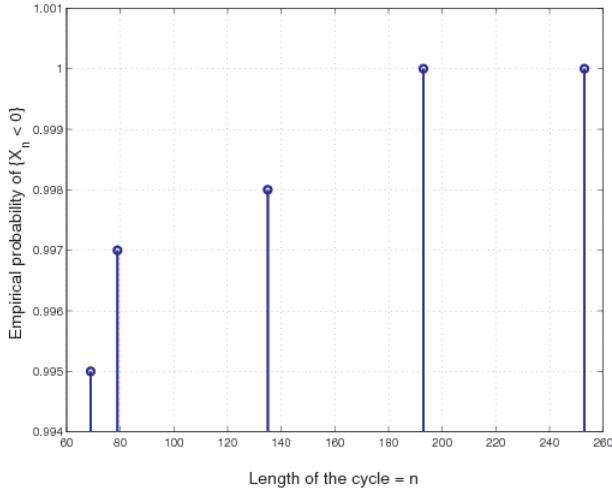


Fig. 2. Plot for empirical probability of the cycle being contractive vs the length of the cycle  $n$  with  $\Phi(r) = \frac{1}{10} \sqrt{r}$ .

does not require a priori knowledge of the vertex and edge weights of  $G$ . It is evident from this example as we first fix a cycle  $W$  and then select weights from a specified interval. This is not the case with deterministic negative cycle detection algorithms, which require complete knowledge of the vertex and edge weights of  $G$  prior to their application. In addition, the weights are sampled uniformly at random 1000 times and we find high empirical probability for  $\{X_n < 0\}$ . This highlights the feature that even if the systems in the given family are prone to evolve over time, our algorithm provides uniform probabilistic guarantees.

## 6 Conclusion

In this article we discussed several methods to algorithmically synthesize members of the class of stabilizing switching signals proposed in [12]. A weighted digraph was associated to a switched system and the switching signal is ex-

pressed as an infinite walk on this weighted digraph in a natural way. In this setting we proposed a sufficient condition for the existence of an infinite walk whose corresponding switching signal satisfies the conditions in [12, Theorem 1], and discussed algorithms for the synthesis of the above infinite walk. Thereafter we tackled the question of how likely is it for a “generic” switched system to admit a stabilizing switching signal, and we identified such a class of switched systems in terms of connectivity and vertex and edge weights of the underlying weighted digraph of the switched system. Necessary conditions for the existence of the above infinite walk is currently under investigation and will be reported elsewhere.

## 7 Proofs of Main Results

**Proof of Theorem 11:** i)  $\Rightarrow$  ii) Assume that the given weighted digraph  $G$  admits a closed contractive walk  $W = x_0, (x_0, x_1), x_1, \dots, x_{\ell-1}, (x_{\ell-1}, x_0), x_0$  of length  $\ell$ , but does not admit a contractive circuit. By assumption,  $W$  is not a circuit. We claim that  $W$  can be recursively decomposed into circuits.

Suppose that  $W$  has  $n$  edges,  $n \geq 1$ , which appear more than once in  $W$ .

*Basis Step:* Pick any one of the edges, say  $(x_{k-1}, x_k)$ , that appears more than once in  $W$ . Obtain  $W_1$  and  $W_2$  such that  $W_1$  is the subwalk generated by concatenating the following:

- o  $x_0, (x_0, x_1), x_1, \dots, x_{k-1}, (x_{k-1}, x_k)$ , i.e., the walk from the beginning of  $W$  till the *first* instance of  $(x_{k-1}, x_k)$  in  $W$ , and
- o  $x_k, (x_k, x_{k+1}), x_{k+1}, \dots, x_{\ell-1}, (x_{\ell-1}, x_0), x_0$ , i.e., the walk beginning after the *last* instance of  $(x_{k-1}, x_k)$  in  $W$  till the end of  $W$ ;

and  $W_2 = x_k, \dots, x_k$  is the subwalk generated by removing  $W_1$  from  $W$ . Clearly,  $W_1$  is a closed walk with at most  $(n-1)$  edges, which appear more than once, and  $W_2$  is a closed walk with at most  $n$  edges, that appear more than once. In case  $n = 1$ ,  $W_1$  is a circuit. If in addition  $(x_{k-1}, x_k)$  appears exactly twice,  $W_2$  is also a circuit. Since  $W$  satisfies  $\bar{\Xi}(W) < 0$ , one of the following three conditions holds: i.  $\bar{\Xi}(W_1) < 0$ , ii.  $\bar{\Xi}(W_2) < 0$ , iii. both  $\bar{\Xi}(W_1)$  and  $\bar{\Xi}(W_2) < 0$ .

*Recursive Step:* In case one of the first two conditions holds, we select the subwalk  $W_i$  that satisfies  $\bar{\Xi}(W_i) < 0$ . In case the last condition holds, we select the subwalk  $W_i$  with fewer number of edges, that appear more than once. If the selected  $W_i$  is a circuit, we stop; else we generate  $W_i^{(1)}$  and  $W_i^{(2)}$  by the same procedure explained in Basis Step and continue till we obtain  $W_i^{(j)}$ ,  $j \in \{1, 2\}$  as a circuit.

Armed with the above claim, consider the case when a closed contractive walk  $W$  that satisfies  $\bar{\Xi}(W) < 0$  is decomposed into  $p$  circuits —  $W_1, W_2, \dots, W_p$ . Now, since  $\bar{\Xi}(W) < 0$ , at least one of the  $W_i$ 's,  $i \in \{1, 2, \dots, p\}$ , satisfies  $\bar{\Xi}(W_i) < 0$ . A circuit  $W_i$  satisfying  $\bar{\Xi}(W_i) < 0$  is the one that we are looking for. It contradicts our assumption that there is no circuit on  $G$  that is contractive.

ii)  $\Rightarrow$  iii) Assume that there is a contractive circuit  $W = x_0, (x_0, x_1), x_1, \dots, x_{\ell-1}, (x_{\ell-1}, x_0), x_0$  of length  $\ell$  on the given

weighted digraph  $G$  but no contractive cycle. By assumption,  $W$  is not a cycle. We claim that  $W$  can be recursively decomposed into cycles.

**Basis Step:** Step 1. Suppose  $x_0$  appears  $n$  times other than at the first and last positions of  $W$ . If  $n = 0$ , apply Step 2 on  $W$ ; else decompose  $W$  into subwalks  $W_1, W_2, \dots, W_{n+1}$  in the following fashion:

- o  $W_1$  is the subwalk from the beginning of  $W$  till the first repeated instance of  $x_0$ ,
- o  $W_2$  is the subwalk beginning from the first repeated instance of  $x_0$  till the second repeated instance of  $x_0$ ,
- ⋮
- o  $W_{n+1}$  is the subwalk beginning from the  $n$ th repeated instance of  $x_0$  till the end of  $W$ .

Clearly, each of the above subwalks is a circuit. Since  $W$  satisfies  $\bar{\Xi}(W) < 0$ , there is at least one  $i, i \in \{1, 2, \dots, n+1\}$  such that  $\bar{\Xi}(W_i) < 0$ . Consider the  $W_i$  from above that satisfies  $\bar{\Xi}(W_i) < 0$ . In case there are more than one such  $W_i$ 's, we select the one with the least number of vertices that appear more than once. If the selected  $W_i$  is a cycle, we stop; else we proceed to Step 2.

**Step 2.** Pick a vertex  $x_i, i \neq 0$  that appears more than once in  $W_i$ . Consider the subwalks

- o  $W_i^{(1)}$  obtained by concatenating  $x_0, (x_0, x_1), x_1, \dots, x_{i-1}, (x_{i-1}, x_i), x_i$  (beginning from the initial vertex of  $W_i$  and ending at the first instance of  $x_i$ ), and  $x_i, (x_i, x_{i+1}), x_{i+1}, \dots, x_0$  (beginning from the last instance of  $x_i$  and ending at the final vertex of  $W_i$ );
- o  $W_i^{(2)}$  beginning from the first instance of  $x_i$  and ending at the last instance of  $x_i$ .

Since  $W_i$  satisfies  $\bar{\Xi}(W_i) < 0$ , one of the following is true: i.  $\bar{\Xi}(W_i^{(1)}) < 0$ , ii.  $\bar{\Xi}(W_i^{(2)}) < 0$ , iii. both  $\bar{\xi}(W_i^{(1)})$  and  $\bar{\xi}(W_i^{(2)}) < 0$ .

**Recursive Step:** In case i. or ii. holds, we select the subwalk  $W_i^{(j)}, j \in \{1, 2\}$  that satisfies  $\bar{\Xi}(W_i^{(j)}) < 0$ . In case of the last one, we select the subwalk  $W_i^{(j)}$  with less number of vertices, which appear more than once. If the selected  $W_i^{(j)}$  is a cycle, we stop; else we generate  $W_i^{(j)(1)}$  and  $W_i^{(j)(2)}$  by the same procedure explained in Step 2 (Basis Step) and continue till we obtain a cycle.

Armed with the above claim, consider the case when a closed contractive circuit  $W$  satisfies  $\bar{\Xi}(W) < 0$  is decomposed into  $p$  cycles —  $W_1, W_2, \dots, W_p$ . Now, since  $\bar{\Xi}(W) < 0$ , at least one of the  $W_i$ 's,  $i \in \{1, 2, \dots, p\}$  satisfies  $\bar{\Xi}(W_i) < 0$ . A cycle  $W_i$  satisfying  $\bar{\Xi}(W_i) < 0$  is the one that we are looking for. It contradicts our assumption that there is no cycle on  $G$  that is contractive.

Consequently, i)  $\Rightarrow$  iii).

iii)  $\Rightarrow$  ii) A contractive cycle  $W$  is a contractive circuit. If not, then there is at least one edge in  $W$  that is repeated. But then the corresponding vertices are also repeated, which contradicts the definition of a cycle.

ii)  $\Rightarrow$  i) By definition, a contractive circuit is a closed contractive walk.

The implication iii)  $\Rightarrow$  i) follows at once.

The last claim follows at once from Lemma 9.  $\square$

**Proof of Lemma 19:** Let  $W' = j_0, (j_0, j_1), j_1, \dots, j_{k-1}, (j_{k-1}, j_k), j_k, (j_k, j_i), j_i$  be a walk obtained from Algorithm 1. Consider the sub-walk  $W = j_i, (j_i, j_{i+1}), j_{i+1}, \dots, j_{k-1}, (j_{k-1}, j_k), j_k, (j_k, j_i), j_i$ , which is a cycle by construction. By of Algorithm 1 all the vertices of  $W$  are in  $\mathcal{P}_S$ . We claim that  $|W| \geq \lfloor \Phi(|\mathcal{P}_S|) \rfloor$ . Assume, if possible,  $|W| < \lfloor \Phi(|\mathcal{P}_S|) \rfloor$ . But

$$|W| = |j_i, (j_i, j_{i+1}), j_{i+1}, \dots, j_{k-1}, (j_{k-1}, j_k), j_k| + |j_k, (j_k, j_i), j_i| \\ = |j_i, (j_i, j_{i+1}), j_{i+1}, \dots, j_{k-1}, (j_{k-1}, j_k), j_k| + 1.$$

By hypothesis,  $d_{\mathcal{P}_S}^+ \geq \lfloor \Phi(|\mathcal{P}_S|) \rfloor$ , which implies that

$$|N_{\mathcal{P}_S}^+| \geq \lfloor \Phi(|\mathcal{P}_S|) \rfloor. \quad (11)$$

By choice of  $j_i$  in Algorithm 1,

$$\{j_0, j_1, \dots, j_{i-1}\} \notin N_{\mathcal{P}_S}^+(j_k). \quad (12)$$

From (11) and (12), it follows that  $|\{j_i, j_{i+1}, \dots, j_k\}| \geq |N_{\mathcal{P}_S}^+(j_k)|$ . But  $|N_{\mathcal{P}_S}^+(j_k)| \geq \lfloor \Phi(|\mathcal{P}_S|) \rfloor$ , which implies that  $|\{j_i, j_{i+1}, \dots, j_k\}| \geq \lfloor \Phi(|\mathcal{P}_S|) \rfloor$ , and it is a contradiction. Consequently,  $|W| \geq \lfloor \Phi(|\mathcal{P}_S|) \rfloor + 1$ .  $\square$

**Proof of Theorem 20:** Since the given weighted digraph  $G(\mathcal{P}, E(\mathcal{P}))$  is nicely connected, by Lemma 19 there exists cycle on  $G(\mathcal{P}, E(\mathcal{P}))$  with all vertices of the cycle being in  $\mathcal{P}_S$  and the length of the cycle is at least  $\lfloor \Phi(|\mathcal{P}_S|) \rfloor$ . Such a cycle can be detected by Algorithm 1.

Consider a cycle  $W = j_0, (j_0, j_1), j_1, \dots, j_{n-1}, (j_{n-1}, j_0)$ , of length exactly  $\lfloor \Phi(|\mathcal{P}_S|) \rfloor = n$  (say). Since  $\{j_0, j_1, \dots, j_{n-1}\} \in \mathcal{P}_S$ , (8) can be written as:  $\sum_{k=1}^n w(j_{k-1}, j_k) - \sum_{k=0}^n w(j_k) < 0$ . Let

$$X_n := \sum_{k=1}^n w(j_{k-1}, j_k) - \sum_{k=0}^n w(j_k). \quad (13)$$

Define the filtration  $(\mathfrak{F}_m)_{m=0}^n$  by

$$\mathfrak{F}_m = \sigma\{w(j_{k-1}, j_k), w(j_\ell) \mid k = 0, \dots, m-1, \ell = 0, \dots, m\}.$$

Since  $G$  is nicely weighted,

$$\begin{aligned} \mathbb{E}^{\mathfrak{F}_{k-1}}[X_k] &= X_{k-1} + \mathbb{E}^{\mathfrak{F}_{k-1}}[w(j_{k-1}, j_k) - w(j_k)] \\ &= X_{k-1} + \alpha - \beta \\ &< X_{k-1} \text{ since } \alpha < \beta \text{ by Definition 17.} \end{aligned} \quad (14)$$

Let  $(X_m)_{m=0}^n := (\xi_0 + M_m + A_m)_{m=0}^n$  denote the a.s. unique Doob decomposition [7, Theorem 5.2.10] of the process  $(X_m)_{m=0}^n$ .

In other words, with  $M_0 := 0$  and  $A_0 := 0$ , we have

$$\begin{cases} \xi_0 := X_0, \\ M_m := \sum_{k=1}^m (X_k - \mathbb{E}^{\tilde{\delta}_{k-1}}[X_k]), & m = 1, \dots, n, \\ A_m := \sum_{k=1}^m (\mathbb{E}^{\tilde{\delta}_{k-1}}[X_k] - X_{k-1}), \end{cases}$$

The inequality (14) shows that  $(X_k)_{k=0}^n$  is an  $(\tilde{\delta}_k)_{k=0}^n$  strict supermartingale; the compensator process  $(A_k)_{k=0}^n$  is, therefore, strictly decreasing.

The definition of  $\xi_0$  shows that  $\xi_0 \leq 0$ , and from (14) we get  $A_n \leq (\alpha - \beta)n$ . Since

$$\begin{aligned} \mathbb{P}(X_n > 0) &= \mathbb{P}(\xi_0 + M_n + A_n > 0) \leq \mathbb{P}(M_n + A_n > 0) \\ &\leq \mathbb{P}(M_n > -n(\alpha - \beta)) \\ &= \mathbb{P}\left(\frac{M_n}{A + B} > \frac{-n(\alpha - \beta)}{A + B}\right), \end{aligned}$$

we apply Azuma's inequality [2, Theorem 7.2.1] to the zero-mean martingale process  $(M_m)_{m=0}^n$  to get

$$\begin{aligned} \mathbb{P}(X_n > 0) &\leq \mathbb{P}\left(\frac{M_n}{A + B} > \left(\frac{-(\alpha - \beta)\sqrt{n}}{A + B}\right)\sqrt{n}\right) \\ &\leq \exp\left(-\frac{1}{2}\left(\frac{(\alpha - \beta)\sqrt{n}}{A + B}\right)^2\right), \end{aligned}$$

which gives the estimate in the theorem. The final assertion follows at once from Lemma 9.  $\square$

## References

- [1] A. A. Agrachev, Y. Baryshnikov, and D. Liberzon. On robust Lie-algebraic stability conditions for switched linear systems. *Systems Control Lett.*, 61(2):347–353, 2012.
- [2] N. Alon and J. H. Spencer. *The probabilistic method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., Hoboken, NJ, third edition, 2008. With an appendix on the life and work of Paul Erdős.
- [3] P. A. Bliman and G. Ferrari-Trecate. Stability analysis of discrete-time switched systems through lyapunov functions with nonminimal state. *IFAC Conference on the Analysis and Design of Hybrid Systems (ADHS03)*, St. Malo, France, 2003.
- [4] B. Bollobás. *Modern graph theory*, volume 184 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1998.
- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, third edition, 2009.
- [6] J. Daafouz, P. Riedinger, and C. Iung. Stability analysis and control synthesis for switched systems: a switched Lyapunov function approach. *IEEE Trans. Automat. Control*, 47(11):1883–1887, 2002.
- [7] R. Durrett. *Probability: theory and examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, fourth edition, 2010.
- [8] J. M. Harris, J. L. Hirst, and M. J. Mossinghoff. *Combinatorics and graph theory*. Undergraduate Texts in Mathematics. Springer, New York, second edition, 2008.
- [9] W. P. M. H. Heemels, B. De Schutter, J. Lunze, and M. Lazar. Stability analysis and controller synthesis for hybrid dynamical systems. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 368(1930):4937–4960, 2010.
- [10] J. P. Hespanha and A. S. Morse. Stability of switched systems with average dwell-time. In *Proc. of the 38th Conf. on Decision and Contr.*, pages 2655–2660, Dec 1999.
- [11] H. Ishii, T. Başar, and R. Tempo. Randomized algorithms for synthesis of switching rules for multimodal systems. *IEEE Trans. Automat. Control*, 50(6):754–767, 2005.
- [12] A. Kundu and D. Chatterjee. Stabilizing discrete-time switched linear systems. Proceedings of the 17th ACM International Conference on Hybrid Systems: Computation & Control, 2014, Berlin, Germany, pp. 11–20.
- [13] A. Kundu and D. Chatterjee. Stabilizing switching signals for switched systems. To appear in *IEEE Transactions on Automatic Control*, doi: 10.1109/TAC.2014.2335291.
- [14] J. W. Lee and G. E. Dullerud. Uniform stabilization of discrete-time switched and markovian jump linear systems. *Automatica*, 42:205–218, 2006.
- [15] J. W. Lee and G. E. Dullerud. Uniformly stabilizing sets of switching sequences for switched linear systems. *IEEE Transactions on Automatic Control*, 52:868–874, 2007.
- [16] J. W. Lee and P. P. Khargonekar. Detectability and stabilizability of discrete-time switched linear systems. *IEEE Transactions on Automatic Control*, 54:424–437, 2009.
- [17] S. Lewandowski. Shortest paths and negative cycle detection in graph with negative weights I. the Bellman-Ford-Moore algorithm revisited. *Technical Report 2010/05, Universität Stuttgart, FMI, Stuttgart, Germany*, 2010.
- [18] D. Liberzon. *Switching in systems and control*. Systems & Control: Foundations & Applications. Birkhäuser Boston Inc., Boston, MA, 2003.
- [19] H. Lin and P. J. Antsaklis. Stability and stabilizability of switched linear systems: a survey of recent results. *IEEE Trans. Automat. Control*, 54(2):308–322, 2009.
- [20] J. Löfberg. YALMIP : a toolbox for modeling and optimization in matlab. In *Proceedings of IEEE International Symposium on Computer Aided Control Systems Design*.
- [21] S. Mitra, N. Lynch, and D. Liberzon. Verifying average dwell time by solving optimization problems. In *Hybrid systems: computation and control*, volume 3927 of *Lecture Notes in Comput. Sci.*, pages 476–490. Springer, Berlin, 2006.
- [22] C. H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Publications Inc., Mineola, NY, 1998. Corrected reprint of the 1982 original.
- [23] R. Shorten, F. Wirth, O. Mason, K. Wulff, and C. King. Stability criteria for switched and hybrid systems. *SIAM Rev.*, 49(4):545–592, 2007.
- [24] K. C. Toh, M. J. Todd, and R. H. Tütüncü. On the implementation and usage of SDPT3—a Matlab software package for semidefinite-quadratic-linear programming, version 4.0. In *Handbook on semidefinite, conic and polynomial optimization*, volume 166 of *Internat. Ser. Oper. Res. Management Sci.*, pages 715–754. Springer, New York, 2012.
- [25] T. Yamada and H. Kinoshita. Finding all the negative cycles in a directed graph. *Discrete Appl. Math.*, 118(3):279–291, 2002.
- [26] C. Zaroliagis. Negative cycles in weighted digraphs. *Encyclopedia of Algorithms*, pages 576–578, 2008.
- [27] G. Zhai, H. Bo, K. Yasuda, and A.N. Michel. Qualitative analysis of discrete-time switched systems. *Proc. of the American Control Conference*, pages 1880–1885, 2002.