Dead-beat stabilizability of discrete-time switched linear systems: algorithms and applications

Mirko Fiacchini, Gilles Millérioux.

***Abstract*—This paper deals with dead-beat stabilizability of autonomous discrete-time switched linear systems. Based on a constructive necessary and sufficient condition for dead-beat stabilizability, we propose two algorithms. The first one is concerned with the problem of testing dead-beat stabilizability and computing the shorter stabilizing mode sequence, whenever it exists. The other one implements a method to construct a switched system whose shorter dead-beat stabilizing sequence has a prescribed length. Then, we present numerical assessments and possible applications.**

***Index Terms*—Switched systems, dead-beat stabilizability**

I. Introduction

Switching systems are dynamical systems for which the state dynamics vary between different operating modes according to a switching sequence [1]. Such systems are found in many practical and theoretical domains. For example they are relevant models in networked control systems [2], [3], in congestion control for computer networks [4], in viral mitigation [5], as abstractions of more complex hybrid systems [6], and other fields (see e.g. [7]–[9] and references therein).

This papers addresses the problem of finite-time stabilizability of discrete-time switched linear systems. The objective is two-fold. First, we investigate the problem of checking whether there exists a finite sequence of switches so that the product of the dynamical matrices gives the null matrix. Secondly, we are concerned with building a dead-beat stabilizable system with a sequence of switches of prescribed length. The difficulty in obtaining some relevant results stems from the fact that the question of deciding the stabilizability of a switching system is known to be hard in general, see [7].

First, let us recall some existing results of closely related problems. Dead-beat stabilizability has been addressed for systems with control. In the survey [10], conditions for controllability and observability of switched linear systems based on the concept of A-invariant are provided. Special results concerning the discrete-time case given by [11], [12] are recalled. These papers provide bounds on the minimal length of a controlling sequence for completely controllable systems. The results are given for switched systems with nonsingular transition matrices, defined as reversible in [10].

For switched linear autonomous systems, exponential stabilizability has been addressed in the survey [8]. Therein, it is recalled that the discrete-time counterpart of stabilizability conditions for continuous-time autonomous switched linear systems is not straightforward. Only a sufficient condition, proposed by the authors themselves, is presented. Another criterion, probably less conservative, is given in [13] but is also sufficient. General necessary and sufficient conditions for exponential stabilizability are provided in [14] and their computation-oriented relaxations are provided and analyzed in [15] but the conditions are not appropriate for dead-beat stabilizability.

Furthermore, we should note that dead-beat stabilizability is closely related to the problem of mortality of a set of matrices. Indeed, a set of matrices is mortal if the null matrix can be expressed as the product of finite length of matrices. This problem is discussed in the papers [16]–[18]. It is proved that the decidability on the mortality of a set of matrices is unsolvable except for some particular cases whose mortality problem results NP-complete, e.g. pairs of integer matrices. In the paper [16], it is proved that the problem is unsolvable for finite sets of 3×3 matrices over the integers.

Finally, another closely related issue is the dead-beat stability for autonomous switched linear discrete-time systems. Unlike dead-beat stabilizability, the problems boils down to checking whether every sufficiently long sequence of switches leads to the null matrix. In [19], dead-beat stability has been used to characterize flatness of discrete-time controlled switched linear systems. The issue has been tackled with the notion of nilpotent semigroups. Dead-beat stability has been examined in [20] as well, in which constraints on the switching sequence are considered.

This literature overview shows that the problem of characterizing dead-beat stabilizability for discrete-time switched linear systems is an open problem. A constructive necessary and sufficient condition has been published in the preliminary paper [21]. In the present paper, we go further. It is proved that a system is dead-beat stabilizable if and only if there exists one switching sequence that stabilizes the whole state space. Next, such a condition is used for defining a complete algorithm to test if a given switched system is dead-beat stabilizable. This algorithm considerably outperforms the brute force exhaustive search approach. Furthermore, the condition is used to define a second algorithm for constructing dead-beat stabilizable switched systems. Finally, numerical evaluations and possible applications, namely flatness and cryptography, are presented.

The paper is organized as follows. In Section II, the problems of dead-beat stability and stabilizability are presented. Condition for dead-beat stability and dead-beat stabilizability along with existing algorithms to test them are recalled. The equivalence between dead-beat stabilizability and the existence of one switching sequence that stabilizes the whole state space is proved. In Section III, we provide two algorithms for respectively testing and building dead-beat stabilizable

M. Fiacchini is with Univ. Grenoble Alpes, CNRS, Gipsa-lab, F-38000 Grenoble, France. mirko.fiacchini@gipsa-lab.fr

G. Millérioux is with Université de Lorraine, CRAN, UMR 7039, France, CNRS, CRAN, UMR 7039, France. gilles.millerioux@univ-lorraine.fr

systems. Section IV presents a numerical example and two possible practical applications of dead-beat stabilizable systems like flatness and cryptography. Some concluding remarks are finally given in Section V.

Notation: Given $n \in \mathbb{N}$, define $\mathbb{N}_n = \{j \in \mathbb{N} : 1 \leq j \leq n\}$. The set of q switching modes is $\mathcal{I} = \mathbb{N}_q$ and the related matrices forms a finite collection $\mathcal{A} \subseteq \mathbb{R}^{n\times n}$, whose i-th element is denoted with A_i, i.e. $\mathcal{A} = \{A_i\}_{i\in\mathcal{I}}$, with $A_i \in \mathbb{R}^{n\times n}$ for all $i \in \mathcal{I}$. All the possible sequences of modes of length N is $\mathcal{I}^N = \prod_{j=1}^N \mathcal{I}$, with $\mathcal{I}^N = \emptyset$ if $N = 0$. The length of a sequence σ is denoted by $|\sigma|$ and $|\sigma| = N$ if $\sigma \in \mathcal{I}^N$. For all $L, M \in \mathbb{N}$ such that $0 \leq L \leq M$, define $\mathcal{I}^{[L:M]} = \bigcup_{N=L}^M \mathcal{I}^N$. Given $\sigma \in \mathcal{I}^N$ and $i, j \in \mathbb{N}_N$ with $i \leq j$, σ_i is the i-th element of σ and $\sigma_{[i:j]}$ the subsequence of σ starting at σ_i and terminating at σ_j. The subsequence $\sigma_{[i:j]}$ is the empty sequence if $i > j$. Given σ, δ sequences of modes, (σ, δ) is their concatenation. Given $\sigma \in \mathcal{I}^N$, define $\mathbb{A}_\sigma = \prod_{j=1}^N A_{\sigma_j} = A_{\sigma_N} \cdots A_{\sigma_1}$ and $\prod_{j=m}^n A_{\sigma_j} = I$ if $m > n$ where I stands for the identity matrix. Given $a \in \mathbb{R}$, $\lceil a \rceil$ is the smallest integer greater than or equal to a.

II. Necessary and sufficient condition for dead-beat stabilizability

A. Problem statement

Consider the discrete-time switched linear system

$$x_{k+1} = A_{\sigma_k} x_k, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state at time $k \in \mathbb{N}$ and $\sigma : \mathbb{N} \to \mathcal{I}$ is the switching law. With slight abuse of notation, we employ the index of σ, as in σ_k for instance, to denote both the time realization of the switching law at instant k and the k-th element of the sequence of modes $\sigma \in \mathcal{I}^N$, its meaning being determined by the context.

Before addressing the issue of dead-beat stabilizability, let us first recall a closely related problem, that is the robust dead-beat (or finite-time) stability. This property stipulates that for any sequences of switches, every state reaches the origin in finite steps. More formally,

$$\exists N \in \mathbb{N} \quad \text{s.t.} \quad x_N = 0 \quad \forall \sigma \in \mathcal{I}^N \quad \forall x_0 \in \mathbb{R}^n$$

which is equivalent to

$$\exists N \in \mathbb{N} \quad \text{s.t.} \quad \mathbb{A}_\sigma = A_{\sigma_N} \cdots A_{\sigma_1} = 0 \quad \forall \sigma \in \mathcal{I}^N.$$

The former condition is equivalent to state that the set $\mathcal{A}$ generates a nilpotent semigroup whose usual definition is recalled below.

Definition 1: A semigroup $\mathcal{S}$ is a set together with an associative internal law. A semigroup $\mathcal{S}$ with an absorbing element 0 is said to be nilpotent if there exists a positive integer $t \in \mathbb{N}$ such that the internal law applied to any t elements of $\mathcal{S}$ is always equal to 0. The smallest integer t is called the class of nilpotency of $\mathcal{S}$.

If $\mathcal{S}$ is a set of matrices the associative internal law is the matrix multiplication and the absorbing element is the null matrix.

Theorem 1 (Levitsky's Theorem): Any semigroup of nilpotent matrices can be triangularized.

This theorem is computationally useful since it gives a way of testing nilpotency, and thus robust dead-beat stability, by means of triangularization routines that are polynomially complex with respect to the dimension n of the matrices. Clearly, a necessary condition for the set $\{A_i\}_{i\in\mathbb{N}_q}$ to generate a nilpotent semigroup is that every A_i is nilpotent for $i \in \mathbb{N}_q$.

Similarly to dead-beat stability for which a necessary and sufficient condition and an algorithm to test the condition exist, we aim at providing a condition and an algorithm for characterizing dead-beat stabilizability. We first give its formal definition.

Definition 2: The system (1) is dead-beat stabilizable if for every $x \in \mathbb{R}^n$, there exist $N(x) \in \mathbb{N}$ and a finite sequence $\sigma(x) \in \mathcal{I}^{N(x)}$ such that $\mathbb{A}_{\sigma(x)} x = 0$.

Thus, dead-beat stabilizability implies the existence of a function, associating to every state, a stabilizing switching sequence, that is a closed-loop control, if the switching is considered as the control input. We prove below that a system is dead-beat stabilizable if and only if there exists one switching sequence that stabilizes the whole state space. This would mean that the system is stabilizable through an open-loop switching control. Recall that the existence of a switching sequence exponentially stabilizing the whole space, a condition often referred to as consistent stabilizability, is only sufficient for exponential stabilizability, see [22], [23]. On the other hand, as a peculiarity of dead-beat stabilizability, the following theorem applies.

Theorem 2: The system (1) is dead-beat stabilizable if and only if there exist $N \in \mathbb{N}$ and $\gamma \in \mathcal{I}^N$ such that $\mathbb{A}_\gamma = 0$.

Proof: Sufficiency is trivial. We must prove necessity, i.e. that if the system is dead-beat stabilizable, then there exists a finite mode sequence γ stabilizing the whole state space. We proceed by contradiction. Suppose that there exists a dead-beat stabilizable system for which $\mathbb{A}_\sigma \neq 0$ for all $\sigma \in \mathcal{I}^N$ and every $N \in \mathbb{N}$ and recall that $\mathbb{A}_\sigma \neq 0$ implies that $\ker(\mathbb{A}_\sigma) < n$. Stabilizability implies that $x \in \ker(\mathbb{A}_{\sigma(x)})$ for every $x \in \mathbb{R}^n$ and then

$$\bigcup_{N\in\mathbb{N}} \bigcup_{\sigma\in\mathcal{I}^N} \ker(\mathbb{A}_\sigma) = \mathbb{R}^n,$$

that is absurd since the space $\mathbb{R}^n$ cannot be expressed as the union of countably many subspaces of dimensions smaller than n. Indeed, the union of a (finite or infinite) sequence of n-dimensional closed sets is an n-dimensional closed set, see [24]. ■

Hence, from Theorem 2, to characterize dead-beat stabilizability, it suffices to derive a necessary and sufficient condition for the existence of $N \in \mathbb{N}$ and a finite sequence $\gamma \in \mathcal{I}^N$ such that for any initial condition $x_0 \in \mathbb{R}^n$, the state at time N reaches the origin. That is

$$\exists N \in \mathbb{N}, \quad \exists \gamma \in \mathcal{I}^N \quad \text{s.t.} \quad \mathbb{A}_\gamma = 0. \tag{2}$$

Remark 1: According to the considerations made in the introduction on the notion of mortality, the existence of such a N cannot, in general, be performed in finite time, resulting in an unsolvable problem.

Clearly, robust stability, associated to the notion of nilpotent semigroup, implies dead-beat stabilizability, related to mortality of a set of matrices.

Remark 2: Beyond analysis purposes, let us notice that dead-beat stabilizability can be interesting for control perspectives, in particular flatness-based control for switched linear systems in the form

$$\begin{cases} x_{k+1} = A_{\sigma_k} x_k + B_{\sigma_k} u_k, \\ y_k = C_{\sigma_k} x_k + D_{\sigma_k} u_k, \end{cases} \tag{3}$$

where $u_k \in \mathbb{R}^p$ is the input, $y_k \in \mathbb{R}^m$ is the output and A_{σ_k}, B_{σ_k}, C_{σ_k}, D_{σ_k} are state space matrices of appropriate dimension. Recall that a non-autonomous dynamical system (assumed to be square) is *flat* if there exist an output y_k, referred to as flat output, and an integer k_0, such that all system variables can be expressed, for $k \geq k_0$, as a function of the flat output and a finite number of its backward and/or forward shifts. It turns out that flatness of (3) can be characterized, see [25], by considering the product of matrices $\prod_{i=1}^{K} P_{\sigma'_{k+i-1}}$, where every $P_{\sigma'_{k+i-1}}$ belongs to the finite set of matrices defining the left inverse dynamics of (3) and is expressed in terms of the state space matrices of (3). The switching rule σ' is a function of a finite number r of modes $\sigma_k, \ldots, \sigma_{k-r}$ where r is the inherent delay of the system (coinciding with the relative degree for a SISO system). It has been proved that the system (3) is flat if and only if there exists a finite integer K such that

$$\prod_{i=1}^{K} P_{\sigma'_{k+i-1}} = 0 \tag{4}$$

holds for any arbitrary sequences σ'. However, we can consider a more flexible situation when both the modes σ_k and the control input u_k can be chosen at time k. In such a case, first, we can be interested in an algorithm that finds a sequence of switches which guarantees (4). Then, appropriately applying such a mode sequence ensures that the the system (3) is flat with flat output y_k and a flatness-based control u_k can be designed separately.

In next subsection, we recall from [21] the necessary and sufficient conditions to determine N and γ, whenever they exist.

B. Necessary and sufficient condition

Denote by

$$\mathcal{I}_s = \{i \in \mathcal{I} :\ \det A_i = 0\},\ \ \mathcal{I}_{ns} = \{i \in \mathcal{I} :\ \det A_i \neq 0\}, \tag{5}$$

the sets of singular and nonsingular matrices in $\mathcal{I}$ and by q_s and q_{ns} the number of elements of $\mathcal{I}_s$ and $\mathcal{I}_{ns}$, respectively. Clearly, one has $q_s + q_{ns} = q$ where q is the number of modes of the system (1). The following proposition will be instrumental for the sequel.

Proposition 1 ([21]): The system (1) is dead-beat stabilizable if and only if there exist $m \in \mathbb{N}_n$ finite mode sequences $\sigma^j \in \mathcal{I}^{n^j}$ with $n^j \in \mathbb{N}$, where $j \in \mathbb{N}_m$, such that

$$\sum_{j=1}^{m} \dim \left(\operatorname{im} \left(\prod_{k=1}^{j-1} \mathbb{A}_{\sigma^k} \right) \cap \ker \left(\mathbb{A}_{\sigma^j} \right) \right) = n, \tag{6}$$

and with $\sigma^j_{n^j} \in \mathcal{I}_s$ for every $j \in \mathbb{N}_m$.

Proposition 1 asserts that the system (1) is dead-beat stabilizable if and only if there exists a set of m switching sequences σ^j of finite length n^j, with $j \in \mathbb{N}_m$ and $1 \leq m \leq n$, whose last element is related to a singular matrix, i.e. $\sigma^j_{n^j} \in \mathcal{I}_s$, and such that the intersection of the kernel of $\mathbb{A}_{\sigma^j}$ and the image of the product $\prod_{k=1}^{j-1} \mathbb{A}_{\sigma^k}$ of matrices has a dimension strictly greater than zero. Moreover, the dimension of those intersections is equal to n but those intersections do not necessarily span $\mathbb{R}^n$. Besides, as shown by Corollary 1 in [21], if the dead-beat stabilizability condition is satisfied by the set of sequences σ^j with $j \in \mathbb{N}_m$, it also holds if the prefix of σ^1 involving nonsingular matrices is removed. All in all, every stabilizing sequence candidate starts and terminates with a singular matrix. The reader can find in Section IV-A an illustrative example of matrices, images and kernels dimensions of a possible sequence γ for the mortal set of 6 matrices of dimension $n = 3$ composed by 2 singular matrix and 4 nonsingular ones given in (13), see Table I.

On one hand, given a finite set of matrices corresponding to the modes of (1), we aim at an approach which is based on the necessary and sufficient condition to find the shorter stabilizing sequence γ that leads to (2). Clearly, such an approach should perform better than the exhaustive search, that is the brute-force test of all the possible switching sequences that leads to (2). On the other hand, we are concerned with proposing a method to build a dead-beat stabilizable system with a prescribed length switching sequence. Both issues are addressed in the following section.

III. Algorithms

A. Algorithm to test dead-beat stabilizability

From the explanations given after Proposition 1, the expected algorithm can restrict the search to sequences $\gamma = (\sigma^1, \ldots, \sigma^m) \in \mathcal{I}^N$ such that $\mathbb{A}_\gamma = 0$ is composed of $m \in \mathbb{N}_n$ subsequences σ^j terminating with an element of $\mathcal{I}_s$ and the first subsequence σ^1 is composed of only one singular matrix. Clearly, there might be an infinite number of such sequences. Thus, we propose an algorithm that finds all the stabilizing sequences whose subsequences lengths are bounded by $\bar{h} + 1$ with $\bar{h} \in \mathbb{N}$. As a consequence, the algorithm will find the minimal length stabilizing sequence γ, provided that it does not involve subsequences longer than $\bar{h}+1$. Algorithm 1 given below meets those requirements.

Algorithm 1 Define the dead-beat stabilizing sequences γ of the system (1) with subsequences length $\leq \bar{h}+1$.

```
Input: matrices $\mathcal{A} = \{A_i\}_{i\in\mathcal{I}}$, sets $\mathcal{I}$ and $\mathcal{I}_s$, parameter $\bar{h}$.
 1: $C \leftarrow \emptyset$,                                   ▷ List of candidates
 2: $M \leftarrow \emptyset$;                          ▷ List of stabilizing sequences
 3: for $s \in \mathcal{I}_s$ do insert $s$ in $C$                   ▷ Candidates $\sigma^1$
 4: end for
 5: while $C$ is not empty do
 6:     extract $\sigma$ from $C$        ▷ Candidate $\sigma = (\sigma^1, \cdots, \sigma^j)$
 7:     for $s \in \mathcal{I}_s,\ \delta \in \mathcal{I}^{[0:\bar{h}]}$ do
 8:         if $\dim\ker(A_s\mathbb{A}_\delta\mathbb{A}_\sigma) > \dim\ker(\mathbb{A}_\sigma)$ and
 9:             $\dim\ker(\mathbb{A}_\delta\mathbb{A}_\sigma) = \dim\ker(\mathbb{A}_\sigma)$ then
10:             $\theta \leftarrow (\sigma, \delta, s)$
11:             if $\dim\ker(\mathbb{A}_\theta) = n$ then
12:                 insert $\theta$ in $M$              ▷ $\theta$ is stabilizing
13:             else
14:                 insert $\theta$ in $C$              ▷ $\theta$ is a candidate
15:             end if
16:         end if
17:     end for
18: end while
Output: M                                  ▷ List of stabilizing sequences
```

Some comments on the main steps of Algorithm 1 follow. The structures C and M contain the sequences candidates to dead-beat stabilize (1) and the stabilizing ones, respectively. At Line 3, all the singular matrices are introduced in C, as the first subsequence candidates. At every successive iterative step (starting at Line 5), a candidate σ is considered and extracted from C. The loop starting at Line 7 aims at searching all the subsequence (δ, s) of length $\bar{h}+1$ at most. At iteration p with $p \in \mathbb{N}_m$, the sequences $\sigma^p = (\delta, s)$ must be such that $s \in \mathcal{I}_s$ and

$$\dim\left(\operatorname{im}\left(\prod_{k=1}^{p-1}\mathbb{A}_{\sigma^k}\right)\cap\ker(\mathbb{A}_{\sigma^p})\right) > 0 \qquad (7)$$

which ensures that every term involved in the sum (6) is strictly greater that zero and thus, that the dimension reaches n for a sufficient number of subsequences.

In practice and for the sake of numerical efficiency, the search is performed according to the equivalent test given at Line 8 which is based on the following lemma.

Lemma 1: ([26]) For every pair of matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$, we have that

$$\dim\ker(AB) = \dim\ker(B) + \dim\big(\operatorname{im}(B)\cap\ker(A)\big).$$

Indeed, from Lemma 1, it is clear that (7) is equivalent to

$$\dim\ker\left(\prod_{k=1}^{p}\mathbb{A}_{\sigma^k}\right) > \dim\ker\left(\prod_{k=1}^{p-1}\mathbb{A}_{\sigma^k}\right). \qquad (8)$$

Hence, at each iteration p, the subsequence $\sigma^p = (\delta, s)$ makes drop down the dimension of the image of the product of matrices associated to the new candidate sequence $(\sigma^1, \ldots, \sigma^p)$ or equivalently, it makes increase the kernel. The obtained sequence is stored in M if the dimension of the kernel reaches n or in C otherwise. Note that the test at Line 9 aims at disregarding subsequences δ (which may involve singular matrices) leading to $\dim\ker(\mathbb{A}_\delta\mathbb{A}_\sigma) < \dim\ker(\mathbb{A}_\sigma)$ and would cause redundancy.

Remark 3: Algorithm 1 computes all the mortal sequences with subsequences length $r = \bar{h}+1$ at most and thus can be used to determine the shorter mortal sequence γ by increasing the parameter $\bar{h}$. Let us compare the computational complexity of Algorithm 1 with respect to the brute force enumeration. Recall that the number of sequences of q modes, with positive length $l \in \mathbb{N}$ at most, is given by $\sum_{i=1}^{l} q^i = (q^{l+1}-1)/(q-1)-1$. Consider first the case of balanced subsequences σ^j, that is such that σ^j have length r for $i > 1$. Thus, γ has length $l = (m-1)r+1$ and Algorithm 1 has to inspect $(m-1)q_s(q^r-1)/(q-1)+q_s$ sequences, while the exhaustive search needs to test all the $(q^{(m-1)r+2}-1)/(q-1)-1$ sequences of lengths l at most. For the general non-balanced case, denote with $l \in \mathbb{N}$ the length of the shortest mortal sequence and $r \in \mathbb{N}$ the length of its longer subsequence. Then $\lceil (l-1)/(m-1)\rceil \leq r \leq l-m+1$ which means that Algorithm 1 finds γ testing between $(m-1)q_s(q^{\lceil (l-1)/(m-1)\rceil+1}-1)/(q-1)+q_s$ and $(m-1)q_s(q^{l-m+2}-1)/(q-1)+q_s$ sequences. Summarizing, the key idea is that Algorithm 1 performs an exhaustive search on $m-1$ subsequences σ^j instead of on the whole $\gamma = (\sigma^1, \ldots, \sigma^m)$, where $m \leq n$.

B. Algorithm to build a dead-beat stabilizable system

This section is devoted to the construction of dead-beat stabilizable switched linear systems whose shorter sequence γ such that $\mathbb{A}_\gamma = 0$ has a prescribed length. According to Proposition 1, it is recalled that such a sequence is composed of m subsequences σ^j with $j \in \mathbb{N}_m$. The first subsequence reduces to one element ($|\sigma^1| = 1$) and the corresponding matrix is singular. We deliberately impose the length of the other subsequences such that $|\sigma^j| = r^j$ with $r^j \in \mathbb{N}$, for $j \geq 1$. Hence, the length of the expected sequence is $1 + \sum_{j=2}^{p} r^p$. The prescribed number of singular and nonsingular matrices are q_s and q_{ns}, respectively. We aim at an iterative algorithm which consists in computing successively the subsequences σ^j with $j > 1$.

At Step $p \in \mathbb{N}$ with $2 \leq p \leq n$ and given σ^j with $j \in \mathbb{N}_{p-1}$ obtained at Step $p-1$, the algorithm must compute a subsequence $\sigma_p = (\delta_p, s_p)$ of given length r^p such that:

- the last element A_{s_p} is one of the singular matrices $\mathcal{I}_s$;
- the condition
$$\dim\ker\left(A_{s_p}\mathbb{A}_{\delta_p}\prod_{k=1}^{p-1}\mathbb{A}_{\sigma^k}\right) = \dim\ker\left(\prod_{k=1}^{p-1}\mathbb{A}_{\sigma^k}\right)+1 \qquad (9)$$
holds.

Actually, the last condition is a particular case of (8). Indeed, it is recalled that, according to Proposition 1, the dimension of the kernel between two consecutive subsequences σ^{p-1} and σ^p increases. Here, we impose that the kernel dimension increases by one between iteration $p-1$ and p. As a result, at Step p, $\dim\ker\left(\prod_{k=1}^{p-1}\mathbb{A}_{\sigma^k}\right) = p-1$ and the total number m of subsequences is equal to n.

After those considerations, we are able to detail the constructive procedure which is summarized in Algorithm 2. The initialization (Line 4-13) consists in randomly computing the q_s singular matrices and $q_{ns} - n + 1$ nonsingular ones. In particular the singular matrices are given by $A_s = T_s^{-1}\Lambda_s T_s$ with:

(i) Λ_s a matrix with one eigenvalue 0 and the other ones randomly generated but non-null. Without loss of generality, we impose the following structure of Λ_s

$$\Lambda_s = \left[\begin{array}{c|ccc} 0 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \bar{\Lambda}_s & \\ 0 & & & \end{array}\right]$$

with $\bar{\Lambda}_s \in \mathbb{R}^{n-1\times n-1}$ a nonsingular matrix.

(ii) T_s a randomly generated invertible matrix.

The columns of T_s^{-1} are the eigenvectors of A_s, with the first one related to the null eigenvalue.

The first iteration of Algorithm 2 (Line 15) consists in choosing the only matrix composing σ^1 among the singular ones. Then, the algorithm must build a sequence of subsequences σ^p (Line 16-24) causing the kernel dimension to increase of one, see (9). To this end, consider the following lemma.

Lemma 2: ([26]) Given $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$ then

$$\dim\ker(AX) = \dim(\mathrm{im}(B) \cap \ker(A))$$

holds, where X is a basis matrix of $\mathrm{im}(B)$.

Denoting with $X_{p-1} \in \mathbb{R}^{n\times n-p+1}$ a basis matrix of $\prod_{j=1}^{p-1}\mathbb{A}_{\sigma^j}$, condition (9) is equivalent to

$$\dim\ker(\mathbb{A}_{\sigma^p}X_{p-1}) = 1 \qquad (10)$$

from Lemmas 1 and 2.

This being the case, the computation at iteration p is detailed hereafter.

(a) compute the random sequences α_p and β_p of modes related to nonsingular matrices and such that $|\alpha_p|+|\beta_p| = r^p - 2$ (Line 17). Note that $\mathbb{A}_{\alpha_p}$ and $\mathbb{A}_{\beta_p}$ are nonsingular;

(b) select $s_p \in \mathcal{I}_s$, then A_{s_p} singular (Line 18);

(c) define (Line 19)

$$\begin{aligned} B_p &= \mathbb{A}_{\alpha_p}^{-1}T_{s_p}^{-1}R_pC_p^{-1} \\ C_p &= \left[\mathbb{A}_{\beta_p}X_{p-1} \;\; (\mathbb{A}_{\beta_p}X_{p-1})^{\perp}\right] \end{aligned} \qquad (11)$$

with

$$R_p = \left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \bar{R}_p & \\ 0 & & & \end{array}\right]$$

where $\bar{R}_p \in \mathbb{R}_p^{n-1\times n-1}$ is an arbitrary nonsingular matrix and where $V^{\perp}$ is a basis of the subspace orthogonal to the one spanned by the columns of V, implying the nonsingularity of C_p.

The matrix B_p is such that

$$\begin{aligned} &A_{s_p}\mathbb{A}_{\alpha_p}B_p\mathbb{A}_{\beta_p}X_{p-1} = T_{s_p}^{-1}\Lambda_{s_p}R_pC_p^{-1}\mathbb{A}_{\beta_p}X_{p-1} \\ &= T_{s_p}^{-1}\Lambda_{s_p}R_pI_{(:,n-p+1)} \end{aligned} \qquad (12)$$

where $I_{(:,n-p+1)}$ is the matrix given by the first $n-p+1$ columns of the identity matrix (since $M^{-1}M_{(:,m)} = I_{(:,m)}$ for all nonsingular M and positive $m \in \mathbb{N}$). Therefore, since

$$\Lambda_{s_p}R_p = \left[\begin{array}{c|ccc} 0 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & \bar{\Lambda}_{s_p}\bar{R}_p & \\ 0 & & & \end{array}\right]$$

then the kernel of $A_{s_p}\mathbb{A}_{\alpha_p}B_p\mathbb{A}_{\beta_p}X_{p-1}$ is $\{y \in \mathbb{R}^{n-p+1} : y_i = 0\ \forall i \geq 2\}$. Hence, the dimension of the kernel of $A_{s_p}\mathbb{A}_{\alpha_p}B_p\mathbb{A}_{\beta_p}X_{p-1}$ increases by 1;

(d) define the new mode i_p such that $A_{i_p} = B_p$. The matrix A_{i_p} is introduced in the matrix set of stabilizable system. Define $\sigma^p = (\beta_p, i_p, \alpha_p, s_p)$ (Line 20-23)

Remark 4: The sequences α_p and β_p implicitly determine the position of the matrix B_p within the the p-th subsequence. As particular cases one could either consider an empty α_p and $|\beta_p| = r^p - 2$, leading to put B_p just before A_{s_p}, or an empty β_p and $|\alpha_p| = r^p - 2$ yielding B_p to be the first matrix related to the p-th subsequence. Note moreover that $\mathbb{A}_{\alpha_p}$ and $\mathbb{A}_{\beta_p}$ are composed by only nonsingular matrices. Also singular ones could be considered, but this could lead to stabilizing sequences whose length is smaller than the desired one, i.e. $1 + \sum_{j=2}^{p} r^p$.

Algorithm 2 Build a dead-beat stabilizable system.

```
Input: $q_s$ and $q_{ns}$ cardinalities of $\mathcal{I}$ and $\mathcal{I}_s$, subsequences lengths $r^p$.
 1: $\mathcal{I}_s \leftarrow \emptyset$                              ▷ Initialization
 2: $\mathcal{I}_{ns} \leftarrow \emptyset$
 3: $\mathcal{A} \leftarrow \emptyset$
 4: for $s \in \mathbb{N}_{q_s}$ do                                  ▷ Generate $\mathcal{I}_s$
 5:     generate $A_s$ singular                                       ▷ Item (i)
 6:     insert $s$ in $\mathcal{I}_s$
 7:     insert $A_s$ in $\mathcal{A}$
 8: end for
 9: for $j \in \mathbb{N}_{q_{ns}-n+1}$ do                  ▷ Generate a part of $\mathcal{I}_{ns}$
10:     generate $A_{q_s+j}$ nonsingular                              ▷ Item (ii)
11:     insert $q_s + j$ in $\mathcal{I}_{ns}$
12:     insert $A_{q_s+j}$ in $\mathcal{A}$
13: end for
14: random selection of $s_1 \in \mathcal{I}_s$                       ▷ First step
15: $\sigma^1 \leftarrow s_1$
16: for $p \in \{i \in \mathbb{N} : 2 \leq i \leq n\}$ do             ▷ $p$-th step
17:     random selection of $\alpha_p$ and $\beta_p$                  ▷ Item (a)
18:     random selection of $s_p \in \mathcal{I}_s$                   ▷ Item (b)
19:     compute $B_p$                                                 ▷ Item (c)
20:     $A_{i_p} \leftarrow B_p$                                      ▷ Item (d)
21:     insert $i_p$ in $\mathcal{I}_{ns}$
22:     insert $A_{i_p}$ in $\mathcal{A}$
23:     $\sigma^p \leftarrow (\beta_p, i_p, \alpha_p, s_p)$
24: end for
Output: $\mathcal{A}$                        ▷ Matrix set of the stabilizable system
```

IV. Applications

A numerical example and a potential application to cryptography of the results presented in this paper are illustrated in this section.

A. Numerical example

This numerical example illustrates the construction of a dead-beat stabilizable system by using Algorithm 2 and then, the search of the shorter mortal sequence γ by using Algorithm 1.

First, by using Algorithm 2, we construct a mortal set of matrices in $\mathbb{R}^{3\times 3}$ for which the shorter sequence γ satisfying $\mathbb{A}_\gamma = 0$ has length 11. In particular, the first subsequence has length 1 and the second and third ones have length 5, i.e. $r^1 = 1$ and $r^2 = r^3 = 5$. Moreover, we set the number of singular matrices to be $q_s = 2$ and that of nonsingular ones to be $q_{ns} = 4$. The resulting matrices are

$$\left[\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \\ \hline A_5 & A_6 \end{array}\right] = \left[\begin{array}{ccc|ccc} -11.0239 & 7.7564 & -16.4615 & 0.5558 & -1.0979 & -0.4113 \\ 6.9246 & -4.7557 & 10.1980 & -1.1203 & -1.4158 & -0.3680 \\ 11.8426 & -8.3326 & 17.6842 & -1.5327 & 0.0596 & -1.3610 \\ \hline 0.5185 & -1.2927 & -0.6147 & 0.1960 & -2.2309 & 2.4620 \\ 0.2137 & 0.3748 & 0.0275 & 1.1627 & -1.0204 & 0.7634 \\ 0.4750 & 0.2060 & -0.1329 & -1.0936 & 1.6483 & -0.9587 \\ \hline -0.3712 & 0.5551 & -0.4093 & 1.6825 & 13.3531 & -17.0264 \\ -0.7578 & -0.5568 & -0.1609 & -2.7298 & -5.9840 & 13.4989 \\ -0.5640 & -0.8951 & 0.4093 & -1.5657 & -14.2341 & 18.4099 \end{array}\right] \tag{13}$$

with eigenvalues $(2.1003, -0.1957, 0)$, $(0.3802 + 0.7026i, 0.3802 - 0.7026i, 0)$, $(-0.5181 + 0.3747i, -0.5181 - 0.3747i, 0.5174)$, $(1.1848, -2.2680, -1.1378)$, $(-1.0175 + 2.0238i, -1.0175 - 2.0238i, 0.2518)$ and $(0.2256, 6.9414 + 7.8336i, 6.9414 - 7.8336i)$, respectively. The sequence of modes γ is $(2, 5, 4, 3, 3, 2, 5, 3, 4, 6, 1)$, for which the subsequences, image and kernel dimensions are given in Table I.

TABLE I
SEQUENCE γ, SUBSEQUENCES, IMAGE AND KERNEL DIMENSIONS

$\mathbb{A}_\gamma$	$\mathbf{A_1}\cdot$	$A_6\cdot$	$A_4\cdot$	$A_3\cdot$	$A_5\cdot$	$\mathbf{A_2}\cdot$	$A_3\cdot$	$A_3\cdot$	$A_4\cdot$	$A_5\cdot$	$\mathbf{A_2}$
$\dim \mathrm{im}(\mathbb{A}_\sigma)$	0	1	1	1	1	1	2	2	2	2	2
$\dim \ker(\mathbb{A}_\sigma)$	3	2	2	2	2	2	1	1	1	1	1
$\mathbb{A}_{\sigma^j}$	$\mathbb{A}_{\sigma^3}$					$\mathbb{A}_{\sigma^2}$					$\mathbb{A}_{\sigma^1}$

The time plot of x^i_k $(i = 1, 2, 3)$ for 10 different initial conditions (randomly generated in the set $\|x\|_\infty \leq 1$) are depicted in Figure 1.

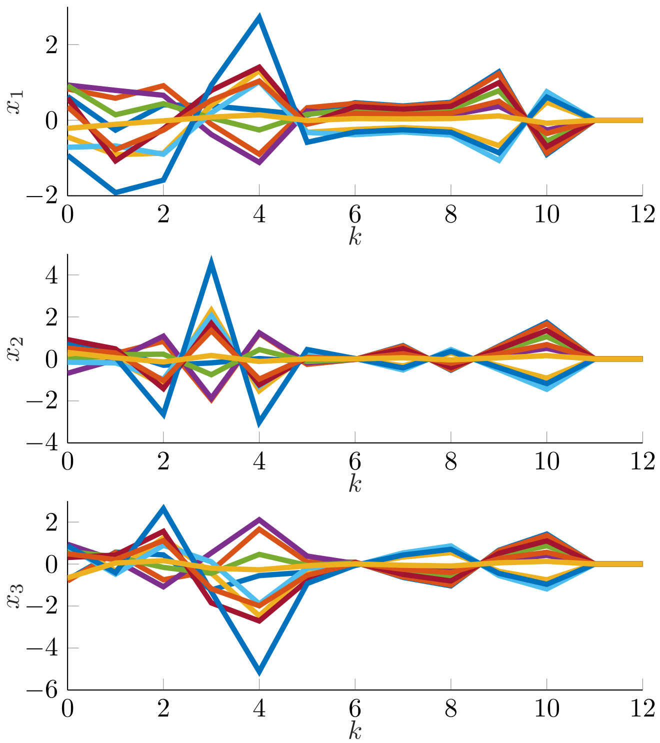

Fig. 1. Evolution in time of x^i_k $(i = 1, 2, 3)$

To illustrate the benefit of using Algorithm 1, the sequence γ has been searched by applying the brute force approach and Algorithm 1 for comparison, using an Intel® Core™ i7-6600U CPU @ 2.60GHz × 4 processor with 16GB of RAM. The brute force approach needed $817s$, that is more than 13 minutes, to find the solution. By comparison, Algorithm 1 found the solution in $0.105s$. Additional examples highlighting the efficiency of this algorithm can be found in the paper [21].

B. Cryptography

We illustrate here the potential relevance of dead-beat stabilizability and the use of Algorithm 2 in the context of cryptography. Actually, it is shown that dead-beat stabilizability may allow to break down a barrier in the design of so-called statistical Self-Synchronizing Stream Ciphers, statistical SSSC for short. They are an extension of the class of deterministic Self-Synchronizing Stream Ciphers, often merely named Self-Synchronizing Stream Ciphers. Roughly speaking, SSSC (deterministic or statistical) are based on automata that are dynamical systems operating on finite fields and that must be designed to deliver sequences of symbols, named keystreams, of high complexity from a statistical point of view. Then, those sequences are used to scramble information to be safely transmitted.

A well-admitted approach for the design of SSSC has been first suggested in [27]. An alphabet denoted by A is considered. It is a finite set of basic elements named symbols. At the ciphering side, the automaton is governed by

$$\begin{cases} z_{k+1} = g(z_k, c_k), \\ w_k = h(z_k), \end{cases} \tag{14}$$

where $z_k \in A$ is the internal state, g is the next-state transition function, $w_k \in A$ is the keystream. The ciphertext (cryptogram) is calculated by $c_k = e(m_k, w_k)$ where e is the ciphering function and m_k the plaintext to be encrypted.

At the deciphering side, the automaton is governed by

$$\begin{cases} \hat{z}_{k+1} = g(\hat{z}_k, c_k), \\ \hat{w}_k = h(\hat{z}_k), \end{cases} \tag{15}$$

where $\hat{z}_k \in A$ is the internal state, g and h are identical to the cipher functions and $\hat{w}_k \in A$ is the keystream. The recovered plaintext symbol is calculated by $\hat{m}_k = d(c_k, \hat{w}_k)$ where d is the deciphering function fulfilling $\hat{m}_k = m_k$ if $\hat{w}_k = w_k$.

If the functions g and h are such that there exist a function ℓ and an integer M verifying

$$z_k = \ell(c_{k-1}, \ldots, c_{k-M}), \tag{16}$$

the automata (14) and (15) are said to have a finite input memory and M is called the delay of memorization. If such a function exists, it means that after a transient time equal to M, the state z_k of the cipher does no longer depend on the initial state and so does the state $\hat{z}_k$ of the decipher. Actually, the states z_k and $\hat{z}_k$ coincide each other for any $k \geq M$. Hence, synchronization between the cipher and the decipher is automatically achieved and the decryption succeeds. It is the reason why such a cipher is called Self-Synchronizing Stream Cipher. Two classes of automata are defined according to the delay of synchronization:

- *Deterministic*: The delay of memorization is bounded by the constant M and a priori fixed.
- *Statistical*: The bound of the delay of memorization is not constant but is a random variable with respect to the sequence of ciphertexts or the initial state vector. Automata with a statistical delay of memorization have never been deeply explored so far. We can mention [28]–[30] for exceptions.

Till now, the usual method to obtain statistical Self-Synchronizing Stream Ciphers is to switch between two automata with and without finite input memory property. This hybrid architecture is equipped with a supervisor which is scanning on-line and at both ends (cipher and decipher) a sequence of cryptograms c of prescribed length. This sequence is compared with a cryptogram of reference (called sync pattern) of the same length. Then, when they coincide, the supervisor of both the cipher and the decipher switches from the automaton without input memory to the one with finite input memory, guaranteeing self-synchronization in finite time after the supervisor switch and so proper decryption.

Here, we show that dead-beat stabilizability can be interesting to propose design methodologies leading to statistical self-synchronizing architectures without supervisors. Obtaining lighter architectures can be a central issue in embedded systems. As a clue to tackle the problem, we propose nonlinear automata in the form

$$z_{k+1} = Q_{\sigma_k} z_k + R_{\sigma_k} c_k \tag{17}$$

where Q_{σ_k} belongs to a finite set of matrices and σ_k is a switching rule which depends in a nonlinear way on a finite sequence of past ciphertexts $c_k, \ldots, c_{k-s}$ with $s \in \mathbb{N}$. In other words, the switching function is of the form $\sigma_k = \varphi(c_k, c_{k-1}, \cdots, c_{k-s})$ where φ is a surjective nonlinear function which selects the mode σ_k at time k from the knowledge of s past ciphertext symbols. In cryptography, such nonlinear functions are commonly used and are called S-boxes. The equation of the decipher is identical to (17) but involves $\hat{z}_k$ instead of z_k.

The property of finite input memory with a statistical delay of memorization is obtained whenever there exists at least a switching sequence of length K so that $\prod_{i=1}^{K} Q_{\sigma_{k+i-1}} = 0$. In other words, the auxiliary system defined as $x_{k+1} = Q_{\sigma_k} x_k$ is dead-beat stabilizable. The quantity M is a random variable since it directly depends on the time before a stabilizing switching sequence is expected to appear. The probability law can be simply obtained whenever the ciphertext symbols are uniformly distributed, which is a common feature of the cipher.

We illustrate such a design through a simple example with $n = 4$ and $q = 10$ matrices Q_i, with $i \in \mathcal{I}$. To operate on the field of two elements $\{0, 1\}$, we adapted the Algorithm 2 by replacing the matrix product with the product modulo 2. We obtained the following mortal set of matrices

$$\left[\frac{Q_1}{Q_6} \middle| \frac{Q_2}{Q_7} \middle| \frac{Q_3}{Q_8} \middle| \frac{Q_4}{Q_9} \middle| \frac{Q_5}{Q_{10}} \right] =$$

$$\left[\begin{array}{cccc|cccc|cccc|cccc|cccc}
0&1&0&0 & 0&1&0&0 & 0&0&1&0 & 0&1&0&0 & 1&0&0&0 \\
0&0&0&1 & 1&0&0&0 & 0&1&0&0 & 0&0&1&0 & 1&1&1&1 \\
0&0&0&0 & 0&0&1&0 & 0&0&0&1 & 1&0&1&0 & 0&0&0&1 \\
1&0&0&0 & 0&0&1&1 & 1&0&0&1 & 0&0&1&1 & 0&1&0&0 \\
\hline
0&1&0&0 & 0&1&0&0 & 0&1&1&0 & 1&1&1&1 & 0&1&1&1 \\
1&0&0&0 & 0&0&1&1 & 1&0&1&0 & 0&1&0&0 & 1&1&0&0 \\
0&0&0&1 & 0&0&1&0 & 0&1&0&0 & 0&0&1&0 & 1&0&0&1 \\
0&0&1&1 & 1&0&0&0 & 0&0&0&1 & 0&1&0&1 & 1&0&0&0
\end{array}\right]$$

for which the shorter sequences γ generating the null matrix have length $K = 10$. Note that Q_1 is the only singular matrix.

Then, $N_r = 1000$ random switching sequences σ, with σ_k uniformly distributed, have been generated. Recall that $\sigma_k = \varphi(c_k, c_{k-1}, \cdots, c_{k-s})$.

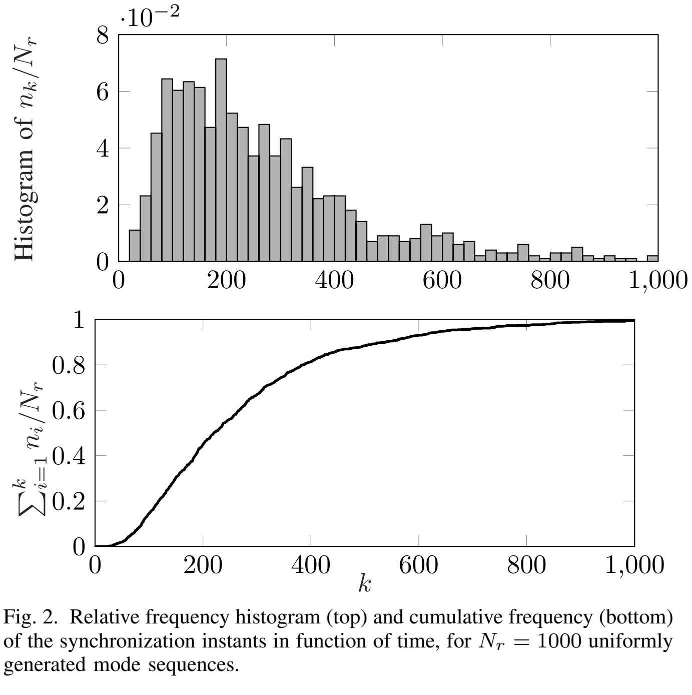

Fig. 2. Relative frequency histogram (top) and cumulative frequency (bottom) of the synchronization instants in function of time, for $N_r = 1000$ uniformly generated mode sequences.

Denoting with n_i the instant at which the i-th sequence generates the null matrix, Figure 2 (top) shows an histogram depicting the relative frequencies of switching sequences that ensure synchronization within a given range of time. Figure 2 (bottom) shows the obtained approximation of the cumulative probability distribution of synchronization with respect to the time k. As expected, the probability tends towards 1 as k increases. It is clear that the shape of the probability depends on the length of the shorter sequences generating the null matrix and the number of modes. Those parameters should be included in the overall design parameters of the cipher and the decipher but this matter is out of the scope of the paper.

V. Conclusion

In this paper, we have considered the problem of characterizing the dead-beat stabilizability for discrete-time switched linear systems. Based on a constructive necessary and sufficient condition, we have proposed an algorithm to check the dead-beat stabilizability of a system and to compute the shorter stabilizing sequence, whenever it exists. It has been shown through a complexity analysis and numerical examples, that the algorithm significantly outperforms an exhaustive search approach. Besides, from the condition, we have derived a second algorithm to build a dead-beat stabilizable systems. Possible applications have been discussed to highlight the interest of the results.

References

[1] D. Liberzon, *Switching in Systems and Control*, ser. Systems and Control: Foundations and Applications. Boston, MA: Birkhuser, 2003.

[2] R. Alur, A. D'Innocenzo, K. H. Johansson, G. J. Pappas, and G. Weiss, "Compositional modeling and analysis of multi-hop control networks," *IEEE Transactions on Automatic control*, vol. 56, no. 10, pp. 2345–2357, 2011.

[3] R. M. Jungers, A. D'Innocenzo, and M. D. Di Benedetto, "Feedback stabilization of dynamical systems with switched delays," in *Proc. of the 51st IEEE Conference on Decision and Control*, 2012, pp. 1325–1330.

[4] R. Shorten, F. Wirth, and D. Leith, "A positive systems model of TCP-like congestion control: asymptotic results," *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 616–629, 2006.

[5] E. A. Hernandez-Vargas, R. H. Middleton, and P. Colaneri, "Optimal and MPC switching strategies for mitigating viral mutation and escape," in *Proc. of the 18th IFAC World Congress Milano (Italy) August*, 2011.

[6] 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, 1999.

[7] R. Jungers, "The joint spectral radius," *Lecture Notes in Control and Information Sciences*, vol. 385, 2009.

[8] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: a survey of recent results," *IEEE Transaction on Automatic Control*, vol. 54, no. 2, pp. 308–322, 2009.

[9] R. Shorten, F. Wirth, O. Mason, K. Wulff, and C. King, "Stability criteria for switched and hybrid systems," *SIAM review*, vol. 49, no. 4, pp. 545–592, 2007.

[10] Z. Sun and S. S. Ge, "Analysis and synthesis of switched linear control systems," *Automatica*, vol. 41, pp. 181–195, 2005.

[11] L. T. J. Conner and D. P. Stanford, "State deadbeat response and observability in multi-modal systems," *SIAM Journal of Control and Optimization*, vol. 22, no. 4, pp. 630–644, 1984.

[12] ——, "The structure of the controllable set for multimodal systems," *Linear Algebra and its Applications*, vol. 95, pp. 171–180, 1987.

[13] J. C. Geromel and P. Colaneri, "Stability and stabilization of discrete-time switched systems," *International Journal of Control*, vol. 79, no. 7, pp. 719–728, July 2006.

[14] M. Fiacchini and M. Jungers, "Necessary and sufficient condition for stabilizability of discrete-time linear switched systems: A set-theory approach," *Automatica*, vol. 50, no. 1, pp. 75 – 83, 2014.

[15] M. Fiacchini, A. Girard, and M. Jungers, "On the stabilizability of discrete-time switched linear systems: Novel conditions and comparisons," *IEEE Transactions on Automatic Control*, vol. 61, no. 5, pp. 1181–1193, 2016.

[16] M. S. Paterson, "Unsolvability in 3×3 matrices," *Studies in Applied Mathematics*, vol. 49, no. 1, pp. 105–107, 1970.

[17] V. D. Blondel and J. N. Tsitsiklis, "When is a pair of matrices mortal?" *Information Processing Letters*, vol. 63, pp. 283–286, 1997.

[18] O. Bournez and M. Branicky, "The mortality problem for matrices of low dimensions," *Theory of Computing Systems*, vol. 35, no. 4, pp. 433–448, 2002.

[19] J. Parriaux and G. Millérioux, "Nilpotent semigroups for the characterization of flat outputs of switched linear and LPV discrete-time systems," *Systems and Control Letters*, vol. 62, no. 8, pp. 679–685, 2013.

[20] M. Philippe, G. Millerioux, and R. Jungers, "Deciding the boundedness and dead-beat stability of constrained switching systems," *Nonlinear Analysis: Hybrid Systems*, 2016.

[21] M. Fiacchini and G. Millérioux, "Dead-beat stabilizability of autonomous switched linear discrete-time systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4576–4581, 2017.

[22] Z. Sun, "Stabilizability and insensitivity of switched linear systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1133–1137, 2004.

[23] Z. Sun and S. S. Ge, *Stability Theory of Switched Dynamical Systems*. Springer, 2011.

[24] K. Kuratowski, *Introduction to Set Theory and Topology*, ser. International Series of Monographs on Pure and Applied Mathematics. Pergamon, 1972.

[25] G. Millérioux and J. Daafouz, "Flatness of switched linear discrete-time systems," *IEEE Trans. on Automatic Control*, vol. 54, no. 3, pp. 615–619, March 2009.

[26] C. D. Meyer, *Matrix analysis and applied linear algebra*. SIAM, 2000.

[27] U. M. Maurer, "New approaches to the design of self-synchronizing stream cipher," *Advance in Cryptography, In Proc. Eurocrypt '91, Lecture Notes in Computer Science*, pp. 548–471, 1991.

[28] A. Geraldy, B. Pfitzmann, and A.-R. Sadeghi, "Optimized self-synchronizing mode of operation," in *Proceedings of Fast Software Encryption International Workshop (FSE'2001)*, 2001.

[29] O. Jung and C. Ruland, "Encryption with statistical self-synchronization in synchronous broaband networks," in *Proc. of Cryptographic Hardware and Embedded Systems - CHES99*, 1999, pp. 340–352.

[30] H. M. Heys, "An analysis of the statistical self-synchronization of stream ciphers," in *Proc. of Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, 2001, pp. 897–904.