Characterization of flat outputs of switched linear discrete-time systems: algebraic condition and algorithm

Gilles Millérioux[a], Raphaël Jungers[b]

[a] *Université de Lorraine, CNRS, CRAN,*
2 Rue Jean Lamour, 54519, Vandoeuvre-les-Nancy, France
gilles.millerioux@univ-lorraine.fr
[b] *UCL/ICTEAM, Louvain-la-Neuve, Belgium*
raphael.jungers@uclouvain.be

Abstract

The problem of flat output characterization for switched linear discrete-time systems is addressed. First, an algebraic condition for an output to be flat is provided. It applies for I-flat outputs with the integer I potentially strictly greater than 1. Next, it is proved that such a characterization is decidable. Finally, an efficient algorithm which allows to decide in polynomial time whether a given output is flat is given. The algorithm is built from the observation that the flat output characterization can be expressed in terms of dead-beat stability of an associated constrained switched system. Thus, the notions of joint spectral radius and graph theory play relevant roles in this context.

Keywords: flatness; constrained switched linear systems; graph theory

1. Introduction

This paper investigates the flatness for switched linear discrete-time systems. Flatness for discrete-time systems, first reported in [1, 2], is called difference flatness. It is the counterpart of differential flatness, introduced in [3], dedicated to continuous-time systems. A discrete-time system is flat if it admits flat outputs. A flat output is a function of the state of the system and/or a finite number of consecutive inputs that admits a specific property. This property stipulates that the state as well as the inputs of the system can be written as a function of a finite number of forward or backward shifts

in the output. Such a property is especially interesting both for state reconstruction and control perspectives. Indeed, from the definition, it is clear that flatness provides a generic way of reconstructing the state vector in finite time despite possibly unknown inputs. On the other hand, for control purposes, flatness is also relevant insofar as, given a flat output, the definition of flatness provides in a straightforward manner a constructive way of designing a feedforward control to track a prescribed trajectory of the plant output. For linear discrete-time systems, many applications are described in the book [1]. Beyond process control, it turns out that flatness has been a central purpose in secure communication as well. Indeed, it allows to design specific cryptosystems called self-synchronizing stream ciphers encountered in symmetric cryptography. In this context, the dynamical systems operate on finite fields and take the form of automata. Flatness guarantees that the state of an automaton can be expressed as a function of its past outputs, the output playing the role of the cryptogram [4][5][6]. Besides, flatness for discrete-time systems has also been considered in the context of security in Cyber Physical Systems. In particular, the problem of reconstructing the state of a flat system from measurements that may be corrupted by an adversarial attack has been investigated in [7].

When dealing with flatness, we can be interested by at least two different tasks: the construction of flat outputs and the test of flat outputs. As for the first issue, the reader may refer to [1, 8, 9, 10] for LTI discrete-time systems or [11, 12, 13] for nonlinear discrete-time systems. The other issue amounts to check whether a given output is flat or not. A first approach consists in trying to directly agree with the definition, that is attempting to express the input and the state vector as a function exclusively involving shifts of the output. Alternatively, dedicated approaches have been proposed in the literature to deal with specific classes of discrete systems like linear systems [1, 2, 8], submersive nonlinear systems [14]. For static feedback linearisable discrete-time systems, an efficient test, based on the computation of certain distributions, which can be generalized to the class of forward-flat systems can be found in [15].
In [16], a first characterization of flat outputs for switched linear discrete-time systems was given. It captured the hybrid feature of the problem inherent to the class of switched system. Indeed, the flat output characterization was recast as the characterization of dead-beat stability of a so-called switched auxiliary system. However, until now and except for particular situations, no

methodology to verify the dead-beat stability was proposed in the literature. The contributions of this paper aim at giving a formal framework along with a practical approach to tackle this issue. The notion of constrained switched systems is central to this end. Three main results are provided. First, it is shown that the characterization is decidable. Second, it is shown that the decision can be performed in polynomial time. Finally, a step-by-step algorithm, easily implementable using any standard numerical software, is provided. Moreover, it is shown that the framework is well suited to enlarge the class of admissible flat outputs called I-flat outputs.

The paper is organized as follows. In Section 2, the definitions of flatness and I-flat output are recalled. They are illustrated by basic examples. Section 3 is devoted to an algebraic characterization of flat outputs. The problem, first involving 0-flat and 1-flat outputs, is extended to I-flat outputs with the integer I potentially strictly greater than 1. The computational complexity for checking the conditions is discussed. Then, a graph-based framework involving de Bruijn's graphs is detailed in Section 4. It gives the necessary material to establish a complete tractable characterization along with an efficient algorithm provided in Section 5 to decide whether a given output is flat. Finally, Section 6 aims at illustrating the results.

Standard notation: For any two integers n and m, $\mathbf{1}_n$ refers to the $n-$dimensional identity matrix and $\mathbf{0}_{n\times m}$ stands for the $n \times m$ zero matrix. If irrelevant, the dimension of the zero matrix is omitted and is merely written as $\mathbf{0}$. For a matrix Z, Z' stands for the transpose of Z. The value of a vector z at time k is denoted z_k. For a positive integer i, z_{k+i} denotes the ith forward shift of z_k and z_{k-i} denotes the ith backward shift of z_k.

2. Background on flatness and problem statement

2.1. The general case

Let us first consider a general (linear or nonlinear) discrete-time controlled dynamical system described by

$$x_{k+1} \;=\; f(x_k, u_k), \tag{1}$$

where $k \in \mathbb{N}$ stands for the discrete time, $x_k \in \mathbb{R}^n$ is the state vector (the i-th component is denoted by $x_k^{(i)}$), $u_k \in \mathbb{R}^m$ is the input, f is the state

transition function. Slightly different definitions of a flat output have been given in the literature dealing with discrete-time systems. We give here the definition from which the results of the present paper apply. The remarks that will follow briefly highlight the peculiarities of the definitions.

Definition 1 (flat output)**.** Let h be a function defined over either $\mathbb{R}^n$ or $\mathbb{R}^n \times (\mathbb{R}^m)^I$ with I a strictly positive integer and that takes values in $\mathbb{R}^m$ (a flat output must be defined over the same set as the one of the input u_k). In those respective cases, for any integer $k \in \mathbb{N}$, let $y_k = h(x_k)$ or $y_k = h(x_k, u_{k+\ell_0}, \ldots, u_{k+\ell_0+I-1})$ with ℓ_0 an integer ranging in $\{-I+1, \cdots, 0\}$ (past and future inputs are possibly considered altogether). The function h is said to be a flat output for the dynamical system (1) if every variable of the system can be expressed as a function of y_k and a finite number of its backward and/or forward iterates. In particular, there exist two functions $\mathcal{F}$, $\mathcal{G}$ and integers $r_1,\ r_2,\ s_1,\ s_2$ in $\mathbb{Z}$, satisfying $r_1 \leq r_2$ and $s_1 \leq s_2$, and such that,

$$\left\{ \begin{array}{rcl} x_k & = & \mathcal{F}(y_{k+r_1}, \cdots, y_{k+r_2}) \\ u_k & = & \mathcal{G}(y_{k+s_1}, \cdots, y_{k+s_2}) \end{array} \right. \tag{2}$$

Then, the standard definition of a flat system follows.

Definition 2 (flat system)**.** The dynamical system (1) is flat if it admits a flat output.

Remark 1. A flat output is a function but, hereafter and with a somehow abusive terminology, the value y_k of the function h will also be referred as flat output.

Remark 2. Definition 1 involves past, present and future values of the output to express x_k and u_k in $\mathcal{F}$ and $\mathcal{G}$. Such a characterization has been first considered in [17] for LTI discrete-time systems, then in [16] for switched linear discrete-time systems and finally, considered in recent papers like [18][19]. It generalizes the definitions given in [1] or [8] where only backward and forward shifts were allowed leading to backward and forward difference flatness. In particular, it allows to cope with arbitrary relative degrees for Single Input Single Output systems (or arbitrary inherent delays, an extension of the relative degree to Multiple Input Multiple Outputs systems, see [20]).

Remark 3. Definition 1 involves possible successive forward-shifted inputs as arguments of the function h. Such a feature characterizes the so-called I-flat outputs, a discrete-time counterpart of the definition given in [21]. Hence,

the flat outputs can be classified into several categories. If the flat output is exclusively a function of the state x_k, then the output is called 0-flat. If the flat output is a function of only one (possibly shifted) input $u_{k+\ell_0}$, then the output is called 1-flat. More generally, if the flat output is a function of $I > 0$ consecutive inputs, then the output is called I-flat. For discrete-time systems, I-flat outputs have been discussed in [8] for LTI systems or in [18] for nonlinear systems. Let us notice that similar functions h have been considered in [19][22] although they are not explicitly called I-flat outputs. When only forward shifts of the input is involved in the flat output, the flatness is called forward flatness. In this respect, let us recall that the property of forward flatness is equivalent to linearisability by an endogenous dynamic feedback [23].

Remark 4. Hereafter, it will be assumed that the systems are controllable to comply with Definition 1. However, the results still hold when this property is not fulfilled. Let us just point out that in such a case, the definition of flatness should be slightly reconsidered as in [18]. In such a case, Equations (2) are only defined for trajectories lying in the controllable set.

Remark 5. If y_k is a flat output for the system (1), the R-shift output $y'_k = y_{k+R}$ of (1), with R a non-negative integer, is still a flat output. Indeed, Equations (2) still hold by substituting y_{k+i} with y'_{k+i-R}. The integers r_i and s_i must be shifted accordingly. The output y'_k is well defined as a function h involving the state x_k and a finite number of consecutive inputs.

Remark 6. Let us recall (see [24] for example) that a dynamical system is said to be left invertible if the input u_k can be recovered from a finite number $r \in \mathbb{N}$ of observations y_{k+i} ($i \in \mathbb{Z}$). It can be noticed that for the system (1) to be flat, the property of left invertibility with respect to the output y_k must be fulfilled. Indeed, it is precisely what the existence of the function $\mathcal{G}$ in Equations (2) means.

2.2. The special case of switched linear discrete-time systems

Hereafter, we will examine switched linear discrete-time systems denoted by $\mathcal{S}$ and obeying

$$x_{k+1} = A_{\sigma(k)}x_k + B_{\sigma(k)}u_k. \tag{3}$$

The state vector is $x_k \in \mathbb{R}^n$ and the input is $u_k \in \mathbb{R}^m$. The matrices $A_{\sigma(k)} \in \mathbb{R}^{n\times n}$ and $B_{\sigma(k)} \in \mathbb{R}^{n\times m}$ belong to the respective finite sets $\mathcal{A} = \{A_1, \ldots, A_J\}$, $\mathcal{B} = \{B_1, \ldots, B_J\}$ of cardinality J. At a given time k, the mode $\sigma(k)$ is

delivered by a switching function $\sigma : k \in \mathbb{N} \longmapsto \sigma(k) \in \{1, \ldots, J\} = \mathcal{J}$. A mode sequence over an interval of time $[k_1, k_2]$, that is $\{\sigma(k_1), \ldots, \sigma(k_2)\}$, is shortly denoted by $\sigma(k_1 : k_2)$. If $k_2 < k_1$ then $\sigma(k_1 : k_2)$ is the null sequence. Let us notice that such a notation is somehow abusive since σ is defined over $\mathbb{N}$. For a given switching rule σ, the set of corresponding mode sequences over an interval of time $[k, k+T]$ belongs to $\mathcal{J}^{T+1}$.

The output of the system $\mathcal{S}$ is a function h defined over either $\mathbb{R}^n$ or $\mathbb{R}^n \times (\mathbb{R}^m)^I$ with I a strictly positive integer and that takes values in $\mathbb{R}^m$. In those respective cases, for any integer $k \in \mathbb{N}$ and any integer ℓ_0 ranging in $\{-I+1, \cdots, 0\}$, since the context of switched linear systems is considered, we restrict our attention to the functions h of the form:

$$\begin{array}{rccl} h: & \mathbb{R}^n & \to & \mathbb{R}^m \\ & x_k & \mapsto & y_k = C_{\sigma(k)} x_k, \end{array}$$

or

$$\begin{array}{rccl} h: & \mathbb{R}^n \times (\mathbb{R}^m)^I & \to & \mathbb{R}^m \\ & (x_k, u_{k+\ell_0}, \ldots, u_{k+\ell_0+I-1}) & \mapsto & y_k = C_{\sigma(k)} x_k + \sum_{\ell=\ell_0}^{\ell_0+I-1} D^{\ell}_{\sigma(k+\ell)} u_{k+\ell}. \end{array} \tag{4}$$

The matrices $C_{\sigma(k)} \in \mathbb{R}^{m \times n}$ and $D^{\ell}_{\sigma(k)} \in \mathbb{R}^{m \times m}$ belong to the respective finite sets $\mathcal{C} = \{C_1, \ldots, C_J\}$ and $\mathcal{D} = \{D_1, \ldots, D_J\}$ of cardinality J. Since both u_k and y_k lie in $\mathbb{R}^m$, the dynamical system $\mathcal{S}$ is called a *square* system. The space of input sequences is denoted by $\mathcal{U}$. When the system (3) is driven by the input sequence $\{u\}_{k_1:k_2} = \{u_{k_1}, \ldots, u_{k_2}\} \in \mathcal{U}$, for a mode sequence $\sigma(k_1 : k_2)$, the notation $\{x(x_{k_1}, \sigma, u)\}_{k_1:k_2}$ refers to the solution of (3) in the interval of time $[k_1, k_2]$ starting from x_{k_1} and $\{y(x_{k_1}, \sigma, u)\}_{k_1:k_2} \in \mathcal{Y}$ refers to the corresponding output sequence in the same interval of time $[k_1, k_2]$.

The following basic example aims at illustrating the main aforementioned notions and remarks with a special focus on switched linear discrete-time systems like (3).

Example: Let us consider a switched linear discrete-time system like (3) defined by:

$$\left\{ \begin{array}{l} x^{(1)}_{k+1} = a_{\sigma(k)} x^{(1)}_k + u_k \\ x^{(2)}_{k+1} = u_k, \end{array} \right. \tag{5}$$

where $a_{\sigma(k)}$ belongs to a finite set of real numbers. The possible role of the switching rule σ regarding flatness is briefly discussed after inspection of the

following situations.
Case 1.1: Consider the output defined as $y_k = x_k^{(1)}$. Such an output is flat because Equations (2) are fulfilled. Indeed, one easily obtains $x_k^{(1)} = y_k$ and $x_k^{(2)} = y_k - a_{\sigma(k-1)} y_{k-1}$ that define the function $\mathcal{F}$ and $u_k = y_{k+1} - a_{\sigma(k)} y_k$ that defines $\mathcal{G}$. Let us notice that backward and forward shifts in the output are involved. That corroborates Remark 2. Indeed, as it turns out, the relative degree of (5) with respect to y_k is equal to 1. According to Remark 3, y_k is a 0-flat output since it exclusively involves the state x_k.
Case 1.2: Consider the output defined as $y_k = a_{\sigma(k)} x_k^{(1)} + u_k$. Again, such an output is flat because Equations (2) are fulfilled. Indeed, we obtain $x_k^{(1)} = y_{k-1}$ and $x_k^{(2)} = y_{k-1} - a_{\sigma(k-1)} y_{k-2}$ that define the function $\mathcal{F}$ and $u_k = y_k - a_{\sigma(k)} y_{k-1}$ that defines $\mathcal{G}$. For such an output, only backward shifts are involved, the relative degree of (5) with respect to y_k is equal to zero. It is a 1-flat output since it involves u_k only. Actually, the output $y_k = a_{\sigma(k)} x_k^{(1)} + u_k$ is the 1-shift output of $y_k = x_k^{(1)}$. The fact that it is also a flat output illustrates Remark 5.
Case 1.3: Consider the output defined as $y_k = x_k^{(2)}$. The system is left invertible with respect to y_k since $u_k = y_{k+1}$ and thus, clearly, u_k can be uniquely recovered from a finite sequence of shifted outputs. On the other hand, the whole state cannot be recovered from y_k because attempting to express $x_k^{(1)}$ as a function of the output fails down. That corroborates the fact that left invertibility is not sufficient for an output to be flat as pointed out in Remark 6.
It should be pointed out that the switching rule may or may not affect the flatness property. In the cases 1.1 and 1.2, flatness is verified regardless of the switching rule and values of $a_{\sigma(k)}$. Similarly, in the case 1.3, flatness is not verified regardless of the switching rule and values of $a_{\sigma(k)}$. But even in some basic situation, the switching rule may play a role. For example, let us consider the one-dimensional system $x_{k+1} = x_k + a_{\sigma(k)} u_k$. If, for a given mode $\sigma(k)$ at time k, $a_{\sigma(k)}$ vanishes, then, left invertibility and so flatness is not fulfilled whereas if not so, flatness is always verified.

The aim of this paper is to provide a tractable algebraic condition along with an efficient algorithm to decide whether a given output h is flat or not.

3. Characterization of flat outputs

As stressed in Section 2, a candidate flat output must necessarily be an output such that the system is left invertible. For this reason, we should now recall results on left invertibility for switched linear discrete-time system.

3.1. Left invertibility for switched linear discrete-time systems

First, let us introduce the subsequent vectors and matrices notation:

$$u_{k:k+i} = \begin{pmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+i} \end{pmatrix}, \; y_{k:k+i} = \begin{pmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+i} \end{pmatrix} \tag{6}$$

$$I_{m\times r} = \begin{pmatrix} \mathbf{1}_m & \mathbf{0}_{m\times(m\cdot r)} \end{pmatrix}$$

$$\mathcal{O}_{\sigma(k:k+i)} = \begin{pmatrix} C_{\sigma(k)} \\ C_{\sigma(k+1)}A_{\sigma(k)} \\ \vdots \\ C_{\sigma(k+i)}A_{\sigma(k)}^{\sigma(k+i-1)} \end{pmatrix}. \tag{7}$$

In the equation above, we use the following notation:

$$\begin{array}{rcl} A_{\sigma(k_0)}^{\sigma(k_1)} & = & A_{\sigma(k_1)}A_{\sigma(k_1-1)}\cdots A_{\sigma(k_0)} \text{ if } \; k_1 \geq k_0 \\ & = & \mathbf{1}_n \;\; \text{if } k_1 < k_0. \end{array}$$

Finally, we recursively define the matrix

$$M_{\sigma(k:k+i)} = \begin{pmatrix} D_{\sigma(k)} & \mathbf{0} \\ \mathcal{O}_{\sigma(k:k+i)}B_{\sigma(k)} & M_{\sigma(k+1:k+i)} \end{pmatrix} \tag{8}$$

with

$$M_{\sigma(k:k)} = D_{\sigma(k)}$$

The definition of left invertibility and its characterization recalled below are borrowed from [25]. The existing characterization is restricted to outputs of the form $y_k = C_{\sigma(k)}x_k$ and $y_k = C_{\sigma(k)}x_k + D_{\sigma(k)}u_k$. In the present paper, the extension to the more general outputs described by Equation (4) will be detailed in Section 3.3.

Definition 3 (left invertibility [25])**.** The system (3) is *left invertible* with respect to the output y_k if there exists a non-negative integer $r < \infty$ such that, for all mode sequences in $\mathcal{J}^{r+1}$, for any two inputs sequences $\{u\}_{k:k+r} \in \mathcal{U}$ and $\{u'\}_{k:k+r} \in \mathcal{U}$, the following implication applies for any $x_k \in \mathbb{R}^n$:

$$\{y(x_k, \sigma, u)\}_{k:k+r} = \{y(x_k, \sigma, u')\}_{k:k+r} \quad \Rightarrow \quad u_k = u'_k. \tag{9}$$

In others words, by *left invertibility*, we mean the ability to recover the input u_k from a finite number of $r+1$ observations y_i $(i = 0, \ldots, r)$, the state vector x_k at time k and the mode sequences $\sigma(k : k+r)$ being known. The least integer r for which (3) is left invertible coincides with its *left inherent delay*.

The following theorem, proved in [25], gives a necessary and sufficient condition for the system (3) to be left invertible with respect to y_k.

Theorem 1 ([25])**.** *The system (3) is* left invertible *with respect to the output y_k if and only if there exists a non negative integer $r < \infty$ such that, for all mode sequences in $\mathcal{J}^{r+1}$,*

$$\operatorname{rank} \begin{pmatrix} M_{\sigma(k:k+r)} \\ I_{m\times r} \end{pmatrix} = \operatorname{rank} M_{\sigma(k:k+r)}. \tag{10}$$

Remark 7. In practice, left invertibility is checked by an incremental approach. Start with $r = 0$ and if (10) is not fulfilled, repeat the test after incrementing r by one. For linear discrete-time systems, it is well-known that left invertibility is decidable. Indeed, r is bounded by the dimension n. Hence, the iterative procedure stops after a finite number of tests. On the other hand, for switched linear systems, the upper bound cannot be a priori determined.

Finally, assuming that the system defined by Equation (3) is left invertible with left inherent delay r, we recall from [25] the expression of the system that allows to recover the input sequence of (3) from its output in a sequential way,

$$\left\{ \begin{array}{rcl} \widehat{x}_{k+r+1} & = & P_{\sigma(k:k+r)}\widehat{x}_{k+r} \\ & & +B_{\sigma(k)}I_{m\times r}(M_{\sigma(k:k+r)})^{\dagger}y_{k:k+r} \\ \widehat{u}_{k+r} & = & -I_{m\times r}(M_{\sigma(k:k+r)})^{\dagger}\mathcal{O}_{\sigma(k:k+r)}\widehat{x}_{k+r} \\ & & +I_{m\times r}(M_{\sigma(k:k+r)})^{\dagger}y_{k:k+r} \end{array} \right. , \tag{11}$$

with

$$P_{\sigma(k:k+r)} = A_{\sigma(k)} - B_{\sigma(k)} I_{m\times r} (M_{\sigma(k:k+r)})^{\dagger} \mathcal{O}_{\sigma(k:k+r)}. \tag{12}$$

The matrix $(M_{\sigma(k:k+r)})^{\dagger}$ is the classical Moore-Penrose generalized inverse of $M_{\sigma(k:k+r)}$. Let us recall that for a given matrix Z, the Moore-Penrose generalized inverse $Z^{\dagger}$ is a matrix of the same dimension as Z' so that $ZZ^{\dagger}Z = Z$, $Z^{\dagger}ZZ^{\dagger} = Z^{\dagger}$, $ZZ^{\dagger}$ and $Z^{\dagger}Z$ are Hermitian. The matrices $P_{\sigma(k:k+r)}$ are the dynamical matrices of the left-inverse dynamical system (11).

The main line of the proof follows. After letting $\epsilon_k = x_k - \widehat{x}_{k+r}$, it is shown in [25] that, from (3), (11) and (12), ϵ_k fulfills the dynamics: $\epsilon_{k+1} = P_{\sigma(k:k+r)}\epsilon_k$. Hence, under identical initial conditions x_0 and identical mode sequences, when driven by a sequence of vectors $y_{k:k+r}$, the equalities $\widehat{x}_{k+r} = x_k$ and $\widehat{u}_{k+r} = u_k$ are ensured for all $k \geq 0$. Thus, the input can be indeed recovered. It is interesting to notice that the left inverse system plays a central role to obtain the expression of $\mathcal{F}$ and $\mathcal{G}$ in (2). The reader may refer to [16] to get the expressions as it is out of the scope of the present paper.

Next section is devoted to the characterization of 0-flat and 1-flat outputs reading respectively $y_k = C_{\sigma(k)}x_k$ and $y_k = C_{\sigma(k)}x_k + D_{\sigma(k)}u_k$, assuming that (3) is left invertible and that r has been determined (see Remark 7). Then, the extension to I-flat outputs defined as (4) with $I > 1$ will be detailed.

3.2. 0-flat and 1-flat output characterization

The first characterization of a flat switched systems that we are aware of is from [16]. It relies on the left-inverse dynamical system (11), and is restricted to 0-flat and 1-flat outputs.

Theorem 2. *Let $y_k = C_{\sigma(k)}x_k$ (resp. $y_k = C_{\sigma(k)}x_k + D_{\sigma(k)}u_k$) be the output of system $\mathcal{S}$ and assume that $\mathcal{S}$ is left invertible with left inherent delay r. Then, the output y_k is a* 0-flat output *(resp.* 1-flat output*), if there exists a non negative integer K such that, for all mode sequences in $\mathcal{J}^{r+K}$, and for all $k \geq 0$, the following product holds:*

$$P_{\sigma(k+K-1:k+K-1+r)} P_{\sigma(k+K-2:k+K-2+r)} \cdots P_{\sigma(k:k+r)} = \mathbf{0}, \tag{13}$$

where the matrices involved in the product are defined in (12)*.*

The detailed proof is given in [16]. However, for a better understanding, the main lines are recalled here. It substantially lies on the fact that,

if (13) is fulfilled, then, after a finite number K of forward iterations of (11), the state of the left inverse system $\hat{x}_{k+K+r}$ will no longer depend on the initial condition $\hat{x}_{k+r}$ and will exclusively depend on shifted outputs y_{k+i} ($i = 0, \cdots, r+K$) of $\mathcal{S}$. Besides, the proof also shows that the dynamics of the vector $\epsilon_k := x_k - \hat{x}_{k+r}$ fulfills $\epsilon_{k+1} = P_{\sigma(k:k+r)}\epsilon_k$. Hence, in view of (13), it is inferred that after a finite number of iterations, $x_k = \hat{x}_{k+r}$, and so x_k, will also exclusively depend of shifted outputs y_{k+i} ($i = 0, \cdots, r+K$) of $\mathcal{S}$. Finally, the proof shows that the same property applies for u_k.

Theorem 2 recalls a characterization of a 0-flat output or 1-flat output first given in [16]. The outcome of such a characterization is that it captures the hybrid feature of the problem related to the specific class of switched systems. However, until now, besides the exhaustive search or some very specific situations, no methodology to verify the condition (13) was proposed in the literature. The contributions of this paper aim at giving a formal framework along with a practical approach to improve on the computation-intensive exhaustive search. Indeed, the theorem above does not provide any bound on the minimal number K of output values that is necessary to recover the state and the input. Moreover, the computational cost of the test (13) grows exponentially with respect to K. More precisely, the number of matrices involved in the product is K while the number of sequences of length $r+1$ are J^{r+1}. Hence, an exhaustive search needs $J^{K(r+1)}$ tests, which can be prohibitive for large triplets (J, K, r). We show in Section 5 that not only it is possible to derive an upper bound on K, but also to propose an algorithm where the number of tests may be significantly reduced. In fact, for a given switched discrete-time system with left inherent delay r, the algorithm will allow to decide whether an output is flat in polynomial time. Both the notion of spectral radius and graph-based theory will be important to this end.

Before going into the graph-based framework, we address below, as new results, the extension of left invertibility and flat output characterization, to I-flat outputs ($I > 1$).

3.3. Extension to I-flat outputs with $I > 1$

The following theorem allows to characterize I-flat outputs in a similar manner than for 0-flat and 1-flat outputs by considering an extended system.

Theorem 3. *A I-flat output ($I > 1$) for the system $\mathcal{S}$ is a 1-flat output for the extended system defined by the state $\tilde{x}_k = [x_k, u_{k+\ell_0}, \cdots, u_{k+\ell_0+I-2}] = [x_k, \tilde{x}_k^{n+1}, \cdots, \tilde{x}_k^{n+I-1}] \in \mathcal{X} \times \mathcal{U}^{I-1}$ and input $\tilde{u}_k = u_{k+\ell_0+I-1}$.*

Proof. Let us consider the extended state $\tilde{x}_k = [x_k, u_{k+\ell_0}, \cdots, u_{k+\ell_0+I-2}] = [x_k, \tilde{x}_k^{n+1}, \cdots, \tilde{x}_k^{n+I-1}] \in \mathcal{X} \times \mathcal{U}^{I-1}$ and input $\tilde{u}_k = u_{k+\ell_0+I-1}$. It verifies the following $n + m(I-1)$-dimensional state space description which reads slightly differently according to the case $I = 2$ and $I > 2$.

Case $I = 2$

$$\begin{array}{rcl} \tilde{x}_{k+1} = \begin{pmatrix} x_{k+1} \\ \tilde{x}_{k+1}^{(n+1)} \end{pmatrix} & = & \begin{pmatrix} A_{\sigma(k)} & B_{\sigma(k)} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tilde{x}_k + \begin{pmatrix} \mathbf{0} \\ \mathbf{1}_m \end{pmatrix} u_{k+\ell_0+1} \\ y_k & = & \begin{pmatrix} C_{\sigma(k)} & D^{\ell_0}_{\sigma(k+\ell_0)} \end{pmatrix} \tilde{x}_k + D^{\ell_0+1}_{\sigma(k+\ell_0+1)} u_{k+\ell_0+1} \end{array} \tag{14}$$

Case $I > 2$

$$\begin{array}{rcl} \tilde{x}_{k+1} & = & \begin{pmatrix} x_{k+1} \\ \tilde{x}_{k+1}^{(n+1)} \\ \vdots \\ \tilde{x}_{k+1}^{(n+I-2)} \\ \tilde{x}_{k+1}^{(n+I-1)} \end{pmatrix} \\ & = & \left(\begin{array}{c|c|c|c|c} A_{\sigma(k)} & (\mathbf{0}/B_{\sigma(k)})_0 & (\mathbf{0}/B_{\sigma(k)})_1 & \cdots & (\mathbf{0}/B_{\sigma(k)})_{I-2} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \cdots \\ \vdots & \vdots & & \ddots & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & & \mathbf{0} \end{array}\right) \tilde{x}_k + \begin{pmatrix} (\mathbf{0}/B_{\sigma(k)})_{I-1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{1} \end{pmatrix} u_{k+\ell_0+I-1} \\ y_k & = & \begin{pmatrix} C_{\sigma(k)} & D^{\ell_0}_{\sigma(k+\ell_0)} & \cdots & D^{\ell_0+I-2}_{\sigma(k+\ell_0+I-2)} \end{pmatrix} \tilde{x}_k + D^{\ell_0+I-1}_{\sigma(k+\ell_0+I-1)} u_{k+\ell_0+I-1} \end{array} \tag{15}$$

where $(\mathbf{0}/B_{\sigma(k)})_i$ means that the block number i ($i \in \{0, \ldots, I-1\}$) is either $\mathbf{0}$ or $B_{\sigma(k)}$. Actually, only one block $(\mathbf{0}/B_{\sigma(k)})_i$ is equal to $B_{\sigma(k)}$. It is the block that multiplies the component u_k and thus, it depends on the value of l_0. Recalling that $\ell_0 \in \{-I+1, \ldots, 0\}$ and that $\tilde{x}_k = [x_k, u_{k+\ell_0}, \cdots, u_{k+\ell_0+I-2}]$, the only $(\mathbf{0}/B_{\sigma(k)})_i$ block that is $B_{\sigma(k)}$ is the block number $-l_0$, the other blocks are null matrices.

It is clear that the state space descriptions (14) and (15) depend on a sequence of modes $\{\sigma(k + l_0), \cdots, \sigma(k + l_0 + I - 1)\}$. Hence, it can be identified to a switched system of the form

$$\left\{ \begin{array}{rcl} \tilde{x}_{k+1} & = & \tilde{A}_{\tilde{\sigma}(k)} \tilde{x}_k + \tilde{B}_{\tilde{\sigma}(k)} \tilde{u}_k \\ y_k & = & \tilde{C}_{\tilde{\sigma}(k)} \tilde{x}_k + \tilde{D}_{\tilde{\sigma}(k)} \tilde{u}_k \end{array} \right. \tag{16}$$

where $\tilde{\sigma}$ is a switching rule that depends on the mode sequence $\{\sigma(k+l_0), \cdots, \sigma(k+l_0+I-1)\}$.

□

Example: Let us consider the MIMO square system of dimension n with m inputs and m outputs. The switching rule σ is not constrained and admits two modes ($J=2$) with the corresponding state space matrices

$$\left(\begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_1 \end{array}\right), \quad \left(\begin{array}{c|c} A_2 & B_2 \\ \hline C_2 & D_2 \end{array}\right).$$

Assume that we want to check whether the output $y_k = C_{\sigma(k+1)}A_{\sigma(k)}x_k + C_{\sigma(k+1)}B_{\sigma(k)}u_k + D_{\sigma(k+1)}u_{k+1}$ is a flat output (if so, it would be a 2-flat output). From this perspective, we must define the extended state $\tilde{x}_k = [x_k, u_k]$ (here, $l_0 = 0$ and $I = 2$) and the input $\tilde{u}_k = u_{k+1}$. It verifies the $(n+m)$-dimensional state space description

$$\begin{array}{rcl} \tilde{x}_{k+1} = \left(\begin{array}{c} x_{k+1} \\ \hline \tilde{x}_{k+1}^{(n+1)} \end{array}\right) & = & \left(\begin{array}{c|c} A_{\sigma(k)} & B_{\sigma(k)} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right) \tilde{x}_k + \left(\begin{array}{c} \mathbf{0} \\ \hline \mathbf{1} \end{array}\right) u_{k+1} \\ y_k & = & \left(C_{\sigma(k+1)}A_{\sigma(k)} \mid C_{\sigma(k+1)}B_{\sigma(k)}\right) \tilde{x}_k + D_{\sigma(k+1)}u_{k+1}. \end{array} \tag{17}$$

The state space description depends on the sequence of modes $\{\sigma(k), \sigma(k+1)\}$. Thus, it can be identified to a switched system of the form (16) with

$$\tilde{A}_{\tilde{\sigma}(k)} = \left(\begin{array}{c|c} A_{\sigma(k)} & B_{\sigma(k)} \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right), \; \tilde{B}_{\tilde{\sigma}(k)} = \left(\begin{array}{c} \mathbf{0} \\ \hline \mathbf{1} \end{array}\right),$$

$$\tilde{C}_{\tilde{\sigma}(k)} = \left(C_{\sigma(k+1)}A_{\sigma(k)} \mid C_{\sigma(k+1)}B_{\sigma(k)}\right), \quad \tilde{D}_{\tilde{\sigma}(k)} = D_{\sigma(k+1)}$$

The switching rule $\tilde{\sigma}$ admits 4 modes corresponding respectively to

$$\left(\begin{array}{cc|c} A_1 & B_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline C_1A_1 & C_1B_1 & D_1 \end{array}\right), \; \left(\begin{array}{cc|c} A_1 & B_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline C_2A_1 & C_2B_1 & D_2 \end{array}\right),$$

$$\left(\begin{array}{cc|c} A_2 & B_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline C_1A_2 & C_1B_2 & D_1 \end{array}\right), \; \left(\begin{array}{cc|c} A_2 & B_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline C_2A_2 & C_2B_2 & D_2 \end{array}\right).$$

Finally, from Theorem 3, all the developments and theorems (Theorem 1 for left invertibility and Theorem 2 for flat output characterization) of Subsections 3.1 and 3.2 respectively, still apply to check whether y_k is a flat output, after substituting the matrices A, B, C, D by $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ and the switching rule σ by $\tilde{\sigma}$.

4. A graph-based framework

This section is instrumental in the perspective of proving Theorem 5 that gives a solution to check whether a given output is flat or not. The notion of feasibility transitions and its natural graph interpretation are central elements to conclude on decidability by deriving an upper bound and to assess the polynomial-time complexity of Algorithm 1 proposed in Section 5. It highlights the role played by the relative degree (or the inherent delay) of the system when examining the combinatorics of the graph construction, the graph coinciding with de Bruijns graph.
Every dynamical matrix of the left inverse system (11) involved in the test product (13) depends on a sequence of $r+1$ modes. Hence, except for singular situations or the case when $r = 0$, although σ is not constrained, the mode sequences of two successive matrices involved in the product (13) are related to each other. Indeed, consider the bijective mapping $\phi : \mathcal{J}^{r+1} \to \mathcal{I} = \{1, \ldots, J^{r+1}\}$ that assigns to each possible sequence $\{\sigma(k), \ldots, \sigma(k+r)\}$ an integer i from the set $\mathcal{I}$ which uniquely identifies the sequence. Then, the switching rule σ' is defined as the function from $\mathbb{N}$ to $\mathcal{I}$ which associates to each integer $k \in \mathbb{N}$ the quantity $\sigma'(k) = \phi(\sigma(k), \ldots, \sigma(k+r)) \in \mathcal{I}$. It holds that $\sigma'(k) = \phi(\sigma(k), \ldots, \sigma(k+r))$ and $\sigma'(k+1) = \phi(\sigma(k+1), \ldots, \sigma(k+r+1))$ depend on the common subsequence $\{\sigma(k+1), \ldots, \sigma(k+r)\}$ that stands as the constraint on the switching rule σ'. To formalize the constraints, we introduce the notion of feasible transitions and de Bruijn's graphs.

4.1. Feasible transitions and de Bruijn's graph

Definition 4. The set $\Gamma(\sigma'(k))$ of feasible transitions from a mode $\sigma'(k)$ is the set defined by

$$\begin{aligned} \Gamma(\sigma'(k)) \;=\; & \{i \in \mathcal{I} : i = \phi(\sigma(k+1), \ldots, \\ & \sigma(k+r+1)), \sigma(k+r+1) \in \mathcal{J}\} \end{aligned} \tag{18}$$

In other words, $\Gamma(\sigma'(k))$ is the set of modes $i \in \mathcal{I}$ which can be reached at time $k+1$ when $\sigma(k+r+1)$ varies over the whole range $\mathcal{J}$, $\sigma'(k)$ and thus the sequence $\{\sigma(k+1), \ldots, \sigma(k+r)\}$ being imposed. One has $\Gamma(\sigma'(k)) \subseteq \mathcal{I}$. If the switching rule σ' is not constrained, then $\Gamma(\sigma'(k)) = \mathcal{I}$. It is clear that $\Gamma(\sigma'(k))$ can never be the empty set.

Example:
Consider system $\mathcal{S}$ with $J = 2$ and $r = 1$, then the sequences involved in the

test product (13) are of length $r+1=2$ and read $\{\sigma(k),\sigma(k+1)\}$. In such a case, the bijective mapping $\phi : \mathcal{J}^{r+1} \to \mathcal{I} = \{1,\ldots,J^{r+1}\}$ is defined as:

$$\begin{aligned}\phi(\{11\}) &= 1\\ \phi(\{12\}) &= 2\\ \phi(\{21\}) &= 3\\ \phi(\{22\}) &= 4\end{aligned}$$

The set of feasible transitions is defined as:

$$\begin{aligned}\Gamma(1) &= \{1,2\}\\ \Gamma(2) &= \{3,4\}\\ \Gamma(3) &= \{1,2\}\\ \Gamma(4) &= \{3,4\}\end{aligned}$$

Actually, the family of systems presented here, whose switching signal is constrained, is not new. They are called *constrained switched systems,* and an efficient way to represent these constraints is through a directed graph. See, e.g. [26, 27] for recent works and other applications involving constrained switched systems. The directed graph, denoted hereafter $\mathcal{G}$, describing the constraints on the switching sequences orchestrated by σ' is the combination of a vertex set $\mathcal{V}$ and an edge set $\mathcal{E}$. The vertex v_i represents the mode i of σ' (assigned through ϕ in a bijective way to a mode sequence $(\sigma(k),\cdots,\sigma(k+r))$) while the edge from the node v_{i_1} to node v_{i_2} describes a feasible transition from the mode i_1 to the mode i_2. The directed graph $\mathcal{G}$ having J^{r+1} vertices, since each vertex has exactly J incoming and J outgoing edges, thus the total number of edges equals $J \cdot J^{r+1} = J^{r+2}$.

Definition 5. A sequence $\{i_1, i_2, \ldots, i_N\}$ is said to be admissible or equivalently is accepted by $\mathcal{G}$ if for any $l \in \{1,\cdots,N-1\}$,

$$i_{l+1} \in \Gamma(i_l). \tag{19}$$

In other words, a sequence $\{i_1, i_2, \ldots, i_N\}$ is admissible if there exists a directed path in $\mathcal{G}$ that links v_{i_1} to v_{i_N} and passes through vertices v_l $(l = 1,\ldots,N)$ according to the order of appearance in this sequence. As it turns out, for a not constrained switching rule σ, the graph $\mathcal{G}$ is nothing but the so-called de Bruijn's graph [28] encountered in graph theory. A de Bruijn's graph is a directed graph representing overlaps between sequences of a symbols in a finite set.

Example:
Consider system $\mathcal{S}$ with $J = 2$ and $r = 1$ (sequences of length equal to 2), then the graph $\mathcal{G}$ is depicted in Figure 1.

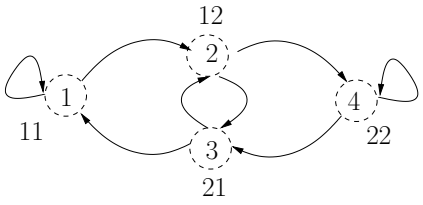

Figure 1: The de Bruijn's graph for sequences of length 2 and alphabet of size 2

Example:
Consider system $\mathcal{S}$ with $J = 2$ and $r = 2$ (sequences of length equal to 3), then the graph is depicted in Figure 2.

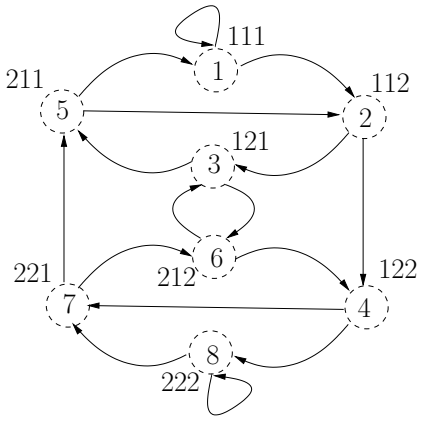

Figure 2: de Bruijn's graph for sequences of length 3 and alphabet of size 2

Hence, while the number of test in the product (13), performed by an exhaustive search is $J^{K(r+1)}$, the number of tests based on the consideration of the de Bruijn's graph reduces to $J^{r+1} \cdot J^{K-1} = J^{r+K}$. Indeed, while J^{r+1} possible sequences must be considered for the first matrix $P_{\sigma(k:k+r)}$ of the product (13), only J sequences must be considered for the $K-1$ remaining matrices of the product (13). Actually, it is an upper bound of the required number of tests which can be further reduced as highlighted by the following remark.

Remark 8. Different sequences $\{\sigma(k), \ldots, \sigma(k+r)\}$ of (3) and so, different modes $\sigma'(k) = \phi(\sigma(k), \ldots, \sigma(k+r))$ might lead to identical matrices $P_{\sigma(k:k+r)}$. This situation occurs for example when the output matrix is constant and does not depend on the mode. As a result, the set $\mathcal{P}$ of matrices $P_{\sigma(k:k+r)}$ is a multiset[1] and one may only consider distinct matrices of $\mathcal{P}$ in order to reduce the computational cost of the test (13). We will denote this set by $\mathcal{Q}$, the number of distinct matrices of $\mathcal{P}$ by L ($L \leq J^{r+1}$) and Q_l an element of $\mathcal{Q}$ ($l \in \{1, \cdots, L\}$). In such a case, the digraph $\mathcal{G}$ may be simplified by merging the vertices of which corresponding mode sequences $\{\sigma(k), \ldots, \sigma(k+r)\}$ give the same matrix $P_{\sigma(k:k+r)}$. The new digraph requires a relabeling of the modes through a new mapping ϕ, which is not bijective in this case. The set $\mathcal{Q}$ coincides with $\mathcal{P}$ when $\mathcal{P}$ involves only distinct elements.

Example: Let us consider again Example 1.1 of Section 2.2. It is a SISO system of dimension $n = 1$. Let us assume that σ has no constraint. The rank test (10) to check the left invertibility succeeds with left inherent delay $r = 1$. Thus, the matrices $P_{\sigma(k:k+1)}$ depend on mode sequences of length 2 ($\{\sigma(k), \sigma(k+1)\}$) and should involve 4 elements. They are computed from Equation (12) and generically read:

$$P_{\sigma(k:k+1)} = \begin{pmatrix} 0 & 0 \\ -a_{\sigma(k)} & 0 \end{pmatrix}.$$

Clearly, two distinct mode sequences $\{\sigma(k), \sigma(k+1)\}$ of length 2 give the same matrix $P_{\sigma(k:k+1)}$ since P only depends on $\sigma(k)$. As a result, there are only $L = 2$ distinct matrices that will define the set of matrices $\mathcal{Q}$.

In such a case, we can introduce an *auxiliary system* for (3), denoted by $\mathcal{S}(\mathcal{G}, \mathcal{Q})$, defined as the constrained switched linear system given by

$$q_{k+1} = Q_{\sigma'(k)} q_k \tag{20}$$

with $q_k \in \mathbb{R}^n$, σ' the constrained switching rule defined by $\sigma'(k) = \phi(\sigma(k), \ldots, \sigma(k+r))$, the matrices of the mode $\sigma'(k)$ verifying $Q_{\sigma'(k)} = P_{\sigma(k:k+r)}$ and the constraints on the switching rule σ' defined by the digraph $\mathcal{G}$.

[1]The notion of a multiset is a generalization of the notion of a set in which elements are allowed to appear more than once.

We are now able to reformulate Theorem 2.

Theorem 4. *Let $y_k = C_{\sigma(k)}x_k$ (resp. $y_k = C_{\sigma(k)}x_k + D_{\sigma(k)}u_k$) be the output of system $\mathcal{S}$ and assume that $\mathcal{S}$ is left invertible with left inherent delay r. Then, the output y_k is a* 0-flat output *(resp.* 1-flat output*) if there exists a non negative integer K such that, for all sequences σ' accepted by the graph $\mathcal{G}$ of the auxiliary system $\mathcal{S}(\mathcal{G}, \mathcal{Q})$, the following product of matrices applies for all $k \geq 0$:*

$$Q_{\sigma'(k+K-1)}Q_{\sigma'(k+K-2)} \cdots Q_{\sigma'(k)} = \mathbf{0} \tag{21}$$

It is worth stressing that the auxiliary system $\mathcal{S}(\mathcal{G}, \mathcal{Q})$ is a constrained switched system. Hence, in other words, Theorem 4 stipulates that the system $\mathcal{S}$ is 0-flat (*resp. 1-flat*) with flat output defined by $y_k = C_{\sigma(k)}x_k$ (*resp.* $y_k = C_{\sigma(k)}x_k + D_{\sigma(k)}u_k$) if the constrained switched system $\mathcal{S}(\mathcal{G}, \mathcal{Q})$ is dead-beat stable.

Remark 9. When the switching rule σ of the system $\mathcal{S}$ is constrained, the digraph $\mathcal{G}$ reduces to a subgraph of de Bruijn's graph.

Remark 10. For the case when $I > 1$, Theorem 4 still applies by considering the system defined by Equations (16) obtained after substituting the matrices A, B, C, D by $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ and the switching rule σ by $\tilde{\sigma}$.

Although the number of test products may be significantly reduced when considering the constraints on the switching rule σ' and the set $\mathcal{Q}$, the cost of the test (21) is still exponential with respect to K, and one can still not provide a bound on K. The next section aims at providing an efficient alternative to check (21). The characterization of flat outputs of $\mathcal{S}$ in terms of dead-beat stability of $\mathcal{S}(\mathcal{G}, \mathcal{Q})$ in central for that purpose.

5. A polynomial time complexity algorithm to characterize a flat output

As it turns out, if the switching rule σ' of the auxiliary system is not constrained, Equation (21) of Theorem 4 is equivalent to the property that $\mathcal{Q}$ generates a nilpotent semigroup. This situation corresponds to the case when the left inherent delay of the system is $r = 0$ (and some other singular

cases when $r > 0$). Hence, in such a case, one can solve the problem efficiently (see [29]) thanks to Levitsky's theorem (Theorem 2.1.7 stated in [30]) which stipulates that any semigroup of nilpotent matrices can be triangularized (for an efficient triangularization method, see e.g. [31] for example).

The aim of this section is to propose an efficient algorithm that checks whether a given output y_k of the system (3) is flat or not when the switching rule σ' is constrained. In such a case, the inherent delay r of the system (3) is necessarily strictly greater than 0. Let us stress that by Theorem 3, we can always assume that the output y_k is 0-flat or 1-flat. If the output y_k of the system (3) is I-flat with $I > 1$, we must consider (16) instead of (3) in Theorem 4 as stressed in Remark 10.

In order to verify whether (21) holds, the following algorithm is proposed. A formal proof of its validity is then given. In the algorithm, the notation $j\sigma(1:r)$ represents the concatenation of the mode j with the sequence of modes $\sigma(1:r)$.

Algorithm 1. input*:*
A left-invertible switched system (3) *(or* (16)*) with left-inherent delay* r,
output*:*
YES if the output y_k *is flat and* K*, NO otherwise.*
Main part:

- *Build all the matrices* $P_{\sigma(k:k+r)}$ *as in* (12)*.*
- *initialize variables* $S_{\sigma(1:r)} = \mathbf{1}_n$ *for every* r*-tuple* $\sigma(1:r)$
- *For* $i := 1$ *to* nJ^r*,*
  - *For all* r*-tuple* $\sigma(1:r)$*,* $S_{\sigma(1:r)} := \sum_{j\in\mathcal{J}} P_{j\sigma(1:r)} S_{j\sigma(1:(r-1))} P'_{j\sigma(1:r)}$
  - *if all the variables* $S_{\sigma(1:r)}$ *are zero, return YES,* $K = i$ *and STOP*
- *Return NO.*

end

Theorem 5. *For any fixed* $r \in \mathbb{N}$*, given a left invertible switched system* (3) *(or* (16)*) with left-inherent delay* r *and output* y_k*, Algorithm 1 decides whether the output is flat or not and runs in polynomial time.*

Proof. Dead-beat stability: apply the construction of Section 4 in order to obtain the auxiliary system in the form of a *constrained switched system* $\mathcal{S}(\mathcal{G}, \mathcal{Q})$. Then, by Theorem 4, the output y_k is flat if and only if $\mathcal{S}(\mathcal{G}, \mathcal{Q})$ is dead-beat stable.
Decidability: if the output y_k is flat, there exists an integer K such that all the products of length larger than K in (21) are equal to zero, recalling the equality $Q_{\sigma'(k)} = P_{\sigma(k:k+r)}$. Indeed, this is a consequence of [32, Theorem 3.1], which guarantees that if a constrained switched system is dead-beat stable, the maximal length of any nonzero product is bounded by $J^r n$.
Computational complexity: computing the total number of possible products of the form $Q_{\sigma'(k+K-1)}Q_{\sigma'(k+K-2)} \cdots Q_{\sigma'(k)}$ and of length K would give J^{nJ^r+r} matrix products, which can be prohibitive. However, from [32, Proposition 3.3], an alternative exists for the computation. Indeed, one can verify that the variable $S_{\sigma(1:r)}$ at step i in Algorithm 1 is equal to

$$\sum_{Z \in \{P_{\sigma(k+i-1:k+i-1+r)} P_{\sigma(k+i-2:k+i-2+r)} \cdots P_{\sigma(k:k+r)}\}} ZZ'.$$

Thus, at step $i = K$, the sum above is over all matrices of the shape $Z \in \{P_{\sigma(k+i-1:k+i-1+r)} P_{\sigma(k+i-2:k+i-2+r)} \cdots P_{\sigma(k:k+r)}\}$, and this sum is zero if and only if all matrices Z are zero. Indeed, the sum of symmetric positive semi-definite matrices is zero if and only if all the matrices are zero. Finally, in Algorithm 1, the main loop contains at most K iterations, and in each loop, one has to compute $2J^r$ products of matrices of size n. Hence, Algorithm 1 computes at most $2nJ^{2r}$ matrix products, which ends the proof. □

6. Illustrative examples

We consider in this section two examples, one devoted to a SISO system with a canonical flat output and another devoted to a MIMO system with a flat output involving linear combination of state components.

Example 1:
Let us consider a ball and beam bench system. A sensor detects the ball position along a beam and an actuator drives the beam to a desired angle through a torque. From the Newton's law, the linearized model can be described as a simple double integrator. A controller aims at regulating the ball at the desired angle. Having in mind a digital implementation, the linearized model

is discretized with a sampling period that may not be constant and properly adapted to the velocity of the ball (for computational reasons for example). For simplicity, let the sampling period having two possible values. The resulting state space model, obtained after an Euler-based discretization method admits a two-dimensional model with x_k the state vector, its first component $x_k^{(1)}$ as the position of the ball, its second component $x_k^{(2)}$ as the velocity of the ball, u_k as the one-dimensional input that corresponds to the controlled angle and y_k as the one-dimensional output that is nothing but the position $x_k^{(1)}$ of the ball. In this respect, the output is called canonical. Let T_k be the sampling period assumed to take values in the set $\{T_1 = 0.1s, T_2 = 0.2s\}$. At each time instant k, a supervisor decides on the value of T_k and thus, is governed by a switching rule σ that admits two modes ($J = 2$) with no constraints. Otherwise stated, the sequence of sampling periods can be arbitrary.

After normalization, the classical state space matrices, resulting for the discretization of a double integrator, read for each mode:

$$\left(\begin{array}{c:cc} A_1 & B_1 \\ \hdashline C_1 & D_1 \end{array}\right) = \left(\begin{array}{cc:c} 1 & T_1 & 0 \\ 0 & 1 & T_1 \\ \hdashline 1 & 0 & 0 \end{array}\right)$$

$$\left(\begin{array}{c:cc} A_2 & B_2 \\ \hdashline C_2 & D_2 \end{array}\right) = \left(\begin{array}{cc:c} 1 & T_2 & 0 \\ 0 & 1 & T_2 \\ \hdashline 1 & 0 & 0 \end{array}\right)$$

First, we perform the rank test (10) to check the left invertibility. It succeeds with left inherent delay $r = 2$. Thus, the set $\mathcal{P}$ of matrices $P_{\sigma(k:k+2)}$ involves 8 elements which are computed from Equation (12). It turns out that distinct mode sequences $\{\sigma(k), \sigma(k+1), \sigma(k+2)\}$ of length $r+1$ give the same matrix $P_{\sigma(k:k+2)}$ (actually, there are only $L = 4$ distinct matrices). Therefore, it is useful to compute the set of matrices $\mathcal{Q}$ for the $J^{r+1} = 8$ possible sequences $\{\sigma(k), \sigma(k+1), \sigma(k+2)\}$.
For the sequences 111 and 112, we have that

$$Q_1 = P_{111} = P_{112} = \begin{pmatrix} 1 & 0.1 \\ -10 & -1 \end{pmatrix}.$$

For the sequences 121 and 122, we have that

$$Q_2 = P_{121} = P_{122} = \begin{pmatrix} 1 & 0.1 \\ -5 & -0.5 \end{pmatrix}.$$

For the sequences 211 and 212, we have that

$$Q_3 = P_{211} = P_{212} = \begin{pmatrix} 1 & 0.2 \\ -10 & -2 \end{pmatrix}.$$

Finally, for the sequences 221 and 222, we have that

$$Q_4 = P_{221} = P_{222} = \begin{pmatrix} 1 & 0.2 \\ -5 & -1 \end{pmatrix}.$$

Therefore, the switching rule σ' of the auxiliary system $\mathcal{S}(\mathcal{G}, \mathcal{Q})$ admits 4 modes. The admissible mode sequences fulfills the constraints explicited in the graph $\mathcal{G}$ depicted on Figure 3. The graph $\mathcal{G}$ has been obtained by merging all vertices v_i of de Bruijn's graph corresponding to $J = 2$ and $r = 2$ assigned to the sequences $\{\sigma(k), \sigma(k+1), \sigma(k+2)\}$ leading to identical matrices $P_{\sigma(k:k+2)}$.

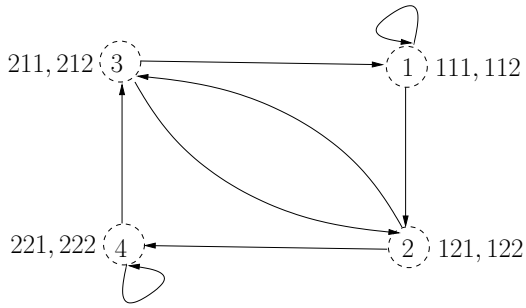

Figure 3: Simplified graph $\mathcal{G}$ from de Bruijn's graph for $J = 2$ and $r = 2$

Let us apply Algorithm 1. The computation at step $i = 1$ reads:

$$S_{11} := P_{111}\mathbf{1}_n P'_{111} + P_{211}\mathbf{1}_n P'_{211} \quad S_{12} := P_{112}\mathbf{1}_n P'_{112} + P_{212}\mathbf{1}_n P'_{212}$$

$$S_{21} := P_{121}\mathbf{1}_n P'_{121} + P_{221}\mathbf{1}_n P'_{221} \quad S_{22} := P_{122}\mathbf{1}_n P'_{122} + P_{222}\mathbf{1}_n P'_{222}.$$

which boils down, according to the graph of Figure 3, to

$$S_{11} = S_{12} := Q_1\mathbf{1}_n Q'_1 + Q_3\mathbf{1}_n Q'_3 \quad S_{21} = S_{22} := Q_2\mathbf{1}_n Q'_2 + Q_4\mathbf{1}_n Q'_4.$$

These matrices are nonzero. we have

$$S_{11} = S_{12} = \begin{pmatrix} 2.05 & -20.5 \\ -20.5 & 205 \end{pmatrix}, \quad S_{21} = S_{22} = \begin{pmatrix} 2.05 & -10.25 \\ -10.25 & 51.25 \end{pmatrix}$$

Hence, we now move to the second iteration of the loop, and we compute

$$S_{11} := P_{111}S_{11}P'_{111} + P_{211}S_{21}P'_{211} \quad S_{12} := P_{112}S_{11}P'_{112} + P_{212}S_{21}P'_{212}$$

$$S_{21} := P_{121}S_{12}P'_{121} + P_{221}S_{22}P'_{221} \quad S_{22} := P_{122}S_{12}P'_{122} + P_{222}S_{22}P'_{222}.$$

which boils down, according to the graph of Figure 3, to

$$S_{11} = S_{12} := Q_1S_{11}Q'_1 + Q_3S_{21}Q'_3 \quad S_{21} = S_{22} := Q_2S_{12}Q'_2 + Q_4S_{22}Q'_4.$$

All these matrices are zero, and we conclude that the output y_k is flat.

Example 2:
Let us consider a numerical example involving a MIMO square system of dimension $n = 4$ with 2 inputs and 2 outputs ($m = 2$). The switching rule σ admits two modes ($J = 2$). Let us assume that σ has no constraints. Hence, any mode sequence is admissible. The state space matrices for each mode read:

$$\left(\begin{array}{c:c} A_1 & B_1 \\ \hdashline C_1 & D_1 \end{array}\right) = \left(\begin{array}{cccc:cc} 0 & -0.3 & -0.4 & 0.2 & 0 & -1.5 \\ -1 & 0.3 & 2 & 0 & 1 & -0.5 \\ -0.05 & 0.005 & 0 & -0.2 & 0 & -0.5 \\ 0 & 0.005 & 0.4 & 0 & 0.5 & 0 \\ \hdashline 1 & -1 & -2 & 2 & 0 & 0 \\ 3 & -1 & -8 & 2 & 0 & 0 \end{array}\right)$$

$$\left(\begin{array}{c:c} A_2 & B_2 \\ \hdashline C_2 & D_2 \end{array}\right) = \left(\begin{array}{cccc:cc} 0 & -0.3 & -0.4 & 0.2 & 0 & -1.5 \\ -2 & 0.3 & 2 & 0 & 1 & -0.5 \\ -0.05 & 0.005 & 0 & -0.2 & 0 & -0.5 \\ 0 & 0.3 & 0.4 & 0 & 0.5 & 0 \\ \hdashline 1 & -1 & -2 & 2 & 0 & 0 \\ 3 & -1 & -8 & 2 & 0 & 0 \end{array}\right)$$

We illustrate here that the procedure for checking whether an output is flat or not applies in a similar fashion and with the same complexity for MIMO systems and even if the output is not canonical. Similarly to example 1, the rank test (10) succeeds with left inherent delay $r = 2$, the set $\mathcal{P}$ of matrices $P_{\sigma(k:k+2)}$ involves 8 elements which are computed from Equation (12) and actually, there are only $L = 4$ distinct matrices. The matrices of the set $\mathcal{Q}$ for the $J^{r+1} = 8$ possible sequences $\{\sigma(k), \sigma(k+1), \sigma(k+2)\}$ are the

following.
For the sequences 111 and 112,

$$Q_1 = P_{111} = P_{112} = \begin{pmatrix} -7.9451 & 3.6890 & 11.5604 & -4.3495 \\ -15.8462 & 6.9692 & 22.7692 & -7.5385 \\ -2.6984 & 1.3347 & 3.9868 & -1.7165 \\ -6.0989 & 2.7198 & 8.7912 & -3.0110 \end{pmatrix}.$$

For the sequences 121 and 122,

$$Q_2 = P_{121} = P_{122} = \begin{pmatrix} 0.5647 & -0.1580 & -0.7445 & 0.0044 \\ -4.1661 & 1.6890 & 5.8801 & -1.5626 \\ 0.1382 & 0.0523 & -0.1148 & -0.2652 \\ -1.6772 & 0.7208 & 2.3975 & -0.7487 \end{pmatrix}.$$

For the sequences 211 and 212,

$$Q_3 = P_{211} = P_{212} = \begin{pmatrix} -15.1648 & 0.0791 & 11.5604 & -4.3495 \\ -30.4615 & -0.3385 & 22.7692 & -7.5385 \\ -5.1049 & 0.1314 & 3.9868 & -1.7165 \\ -11.7033 & -0.0824 & 8.7912 & -3.0110 \end{pmatrix}.$$

Finally, for the sequences 221 and 222,

$$Q_4 = P_{221} = P_{222} = \begin{pmatrix} 1.1356 & 0.1274 & -0.7445 & 0.0044 \\ -8.0883 & -0.2721 & 5.8801 & -1.5626 \\ 0.3285 & 0.1475 & -0.1148 & -0.2652 \\ -3.2334 & -0.0573 & 2.3975 & -0.7487 \end{pmatrix}.$$

The same considerations as in example 1 lead to a switching rule σ' of the auxiliary system $\mathcal{S}(\mathcal{G}, \mathcal{Q})$ admitting 4 modes with admissible mode sequences fulfilling the constraints explicited in the same graph $\mathcal{G}$ depicted on Figure 3.

Let us apply Algorithm 1. The computation at step $i = 1$ gives matrices which are nonzero. As an example, we have

$$S_{11} = 1000 \cdot \begin{pmatrix} 0.6118 & 1.2055 & 0.2109 & 0.4654 \\ 1.2055 & 2.3782 & 0.4150 & 0.9179 \\ 0.2109 & 0.4150 & 0.0728 & 0.1603 \\ 0.4654 & 0.9179 & 0.1603 & 0.3543 \end{pmatrix}.$$

Hence, we now move to the second iteration of the loop. The computation gives $S_{11} = S_{12} = S_{21} = S_{22} = \mathbf{0}$. We conclude that the output y_k is flat.

7. Conclusion

We have addressed the problem of flat output characterization for switched linear discrete-time systems. An algebraic condition has been provided and extends existing results to I-flat outputs with the integer I potentially strictly greater than 1. It has been proved that such a characterization is decidable and an efficient algorithm which allows to decide in polynomial time whether a given output is flat has been given. The algorithm has been derived from the context of constrained switched systems and a graph-based framework involving de Bruijn's graph.

Funding: For Gilles Millérioux, this work is partly supported by the french PIA project "Lorraine Universit d'Excellence", reference ANR-15-IDEX-04-LUE. Raphal Jungers is a FNRS honorary Research Associate. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 864017 - L2C. He is also supported by the Walloon Region and the Innoviris Foundation.

References

[1] H. Sira-Ramirez and S. K. Agrawal. *Differentially Flat Systems*. Marcel Dekker, New York, 2004.

[2] M. Fliess and R. Marquez. Toward a module theoretic approach to discrete-time linear predictive control. *International Journal of Control*, 73:606–623, 2000.

[3] M. Fliess, J. Levine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *Int. Jour. of Control*, 61(6):1327–1361, 1995.

[4] 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*, pages 548–471, 1991.

[5] J. Daemen and K. Paris. The self-synchronzing stream cipher moustique. Technical report, e-Stream Project, 2006. Available at: http://www.ecrypt.eu.org/stream/p3ciphers/mosquito/mosquito_p3.pdf.

[6] B. Dravie, P. Guillot, and G. Millérioux. Flatness and structural analysis as a constructive framework for private communication. *Nonlinear Analysis: Hybrid Systems*, 30:92–105, 2018.

[7] Y. Shoukry, P. Nuzzo, N. Bezzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada. Secure state reconstruction in differentially flat systems under sensor attacks using satisfiability modulo theory solving. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015.

[8] S. Z. Yong, B. Paden, and E. Frazzoli. Computational methods for mimo flat linear systems: Flat output characterization, test and tracking control. In *Proc. of the American Control Conference (ACC'2015)*, Chicago, Il, USA, July 1-3 2015.

[9] G. Millérioux. A constructive approach for the design of finite time self-synchronizing coupled systems with unknown inputs. In *Proc. of the International Symposium on Nonlinear Theory and its Applications (NOLTA 2010)*, Krakow, Poland, September 2010.

[10] T. Boukhobza and G. Millérioux. Graphical characterization of the set of all flat outputs for structured linear discrete-time systems. In *6th Symposium on System Structure and Control, (SSSC 2016)*, Istanbul, Turkey, June 2016.

[11] A. Kaldmae and Ü Kotta. On flatness of discrete-time nonlinear systems. In *Proc. of the 9th IFAC Symposium on Nonlinear Control Systems*, Toulouse, France, 2013.

[12] B. Kolar, A. Kaldmäe, M. Schöberl, Ü. Kotta, and K. Schlacher. Construction of flat outputs of nonlinear discrete-time systems in a geometric and an algebraic framework. In *NOLCOS 2016*, 2016.

[13] B. Kolar, M. Schöberl, and K. Schlacher. A decompsition procedure for the construction of flat outputs of discrete-time nonlinear control systems. In *22nd International Symposium on Mathematical Theory of Networks and Systems*, 2016.

[14] K. Sato. On an algorithm for checking whether or not a nonlinear discrete-time system is difference flat. In *Proc. of 20th International Symposium on Mathematical Theory of Networks and Systems*, Melbourne, Australia, July 2012.

[15] B. Kolar, J. Diwold, and M Schöberl. Necessary and sufficient conditions for difference flatness. 2019. arXiv e-prints (arXiv:1909.02868v2 [math.OC]).

[16] G. Millérioux and J. Daafouz. Flatness of switched linear discrete-time systems. *IEEE Trans. on Automatic Control*, 54(3):615–619, March 2009.

[17] J. Daafouz, M. Fliess, and G. Millérioux. Une approche intrinsèque des observateurs linéaires à entrées inconnues. In *Proc. of the Conférence Internationale Francophone d'Automatique*, Bordeaux, May 2006.

[18] P. Guillot and G. Millérioux. Flatness and submersivity of discrete-time dynamical systems. *IEEE Control Systems Letters*, 4(2):337–342, 2020.

[19] Johannes Diwold, Bernd Kolar, and Markus Schberl. A trajectory-based approach to discrete-time flatness, 2020.

[20] M. K. Sain and J. L. Massey. Invertibility of linear time-invariant dynamical systems. *IEEE Trans. Automatic Control*, 14:141–149, 1969.

[21] S. P. Pereira da Silva. Flatness of nonlinear control systems : a cartan-kähler approach. *Proc. Mathematical Theory of Networks et Systems MTNS'2000*, 01 2000.

[22] Johannes Diwold, Bernd Kolar, and Markus Schberl. A normal form for two-input forward-flat nonlinear discrete-time systems. *International Journal of Systems Science*, 0(0):1–13, 2021.

[23] E. Aranda-Bricaire and C. H. Moog. Linearization of discrete-time systems by exogenous dynamic feedback. *Automatica*, 44(7):1707–1717, 2008.

[24] S. N. Singh. A modified algorithm for invertibility in nonlinear systems. *IEEE Trans. Automat. Control*, 26:595–598, 1981.

[25] S. Sundaram and C. Hadjicostis. Designing stable inverters and state observers for switched linear systems with unknown inputs. In *Proc. of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, December 2006.

[26] Raphael M Jungers, Atreyee Kundu, and WPMH Heemels. Observability and controllability analysis of linear systems subject to data losses. *IEEE Transactions on Automatic Control*, 63(10):3361–3376, 2017.

[27] Matthew Philippe, Ray Essick, Geir Dullerud, and Raphaël M Jungers. Stability of discrete-time switching systems with constrained switching sequences. *Automatica*, 72:242–250, 2015.

[28] I. J. Good. Normal recurring decimals. *Journal of the London Mathematical Society*, s1-21(3):167–169, 1946.

[29] J. Parriaux and G. Millerioux. Nilpotent semigroups for the characterization of flat outputs of discrete-time switched linear and lpv systems. In *Proc. of 51st IEEE Conference on Decision and Control*, Grand Wailea, Maui, Hawaii, December 2012.

[30] H. Radjavi and P. Rosenthal. *Simultaneous Triangularization*. Springer, 2000.

[31] C. Dubi. An algorithmic approach to simultaneous triangularization. *Linear Algebra and its Applications*, 430(11-12):2975 – 2981, 2009.

[32] M. Philippe, G. Millerioux, and R. Jungers. Deciding the boundedness and dead-beat stability of constrained switching systems. *Nonlinear Analysis: Hybrid Systems*, 23:287–299, 2017.