Nilpotent semigroups for the characterization of flat outputs of switched linear and LPV discrete-time systems

Jeremy Parriaux, Gilles Millérioux

Nilpotent semigroups for the characterization of flat outputs of switched linear and LPV discrete-time systems

Jérémy Parriaux[a], Gilles Millérioux[a]

[a] *Université de Lorraine, Research Center for Automatic Control of Nancy (CRAN UMR 7039), 2 rue Jean Lamour, 54519 Vandoeuvre-les-Nancy, France*
jeremy.parriaux@univ-lorraine.fr, gilles.millerioux@univ-lorraine.fr

Abstract

This paper addresses the problem of flat output characterization for switched linear systems. The characterization is also extended to LPV systems. The characterization is based on the notion of nilpotent semigroups. A complete corresponding recursive algorithm is provided. It stops after a finite number of steps bounded by the dimension of the system. Illustrative examples, for the respective class of switched linear and LPV systems, highlight the efficiency of the characterization.

Keywords: flatness; switched linear systems; linear parameter varying; nilpotent semigroups

1. Introduction

Flatness is a control-theoretical concept introduced in [1] and the assets of flatness-based approaches are well-established. A deep insight on flatness along with applications can be found in the book [2]. For a flat discrete-time system (linear or nonlinear), the state variable as well as the input of the system can be written as some function of the output (including forward and backward shifts in the output). Such a property is especially interesting both for state reconstruction and control perspectives. Indeed, it is clear from the definition that flatness provides a generic way of reconstructing the state vector despite possibly unknown inputs. Even more is true, flatness is a structural property of a dynamical system and so ensures the existence of an unknown input observer without any *a priori* structure of the observer.

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. This being the case, an important issue related to flatness is the problem of checking whether a given output of a dynamical system is flat or not. Indeed, it is precisely the flat output which will be exclusively used for the design of the controller or the state reconstructor according to the purpose. 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 derivatives of the output in the continuous case or shifts of the output in the discrete-time case. A more relevant approach has been proposed in [3] for continuous linear systems. For nonlinear systems, flat output characterization has been addressed to a much lesser extent. We may refer to the recent work [4] dealing with flatness of time invariant nonlinear discrete-time systems from a behavioral perspective. In this paper, we propose a characterization for switched linear systems. It is based on the notion of nilpotent semigroups and a complete tractable algorithm is given for checking the conditions. Furthermore, it is shown that the theoretical condition and the corresponding algorithm can be extended to LPV systems with little effort. The layout is as follows. In Section 2, we recall some basics on flatness with a special emphasis on switched linear systems. Section 3 is devoted to the flat output characterization and a description of the corresponding algorithm. An extension to LPV systems is proposed in Section 4. Finally, Section 5 is devoted to illustrative examples.

2. Preliminaries and Definitions

Throughout this paper, we shall examine switched linear systems in the form

$$\left\{\begin{array}{lcl} x_{k+1} & = & A_{\sigma(k)}x_k + B_{\sigma(k)}u_k \\ y_k & = & C_{\sigma(k)}x_k + D_{\sigma(k)}u_k \end{array}\right. \tag{1}$$

The state vector is $x_k \in \mathbb{R}^n$, the input is $u_k \in \mathbb{R}^m$ and the output is $y_k \in \mathbb{R}^p$. All the matrices, namely $A_{\sigma(k)} \in \mathbb{R}^{n\times n}$, $B_{\sigma(k)} \in \mathbb{R}^{n\times m}$, $C_{\sigma(k)} \in \mathbb{R}^{p\times n}$ and $D_{\sigma(k)} \in \mathbb{R}^{p\times m}$ belong to the respective finite sets of cardinality J: $\mathcal{A} = \{A_1, \ldots, A_J\}$, $\mathcal{B} = \{B_1, \ldots, B_J\}$, $\mathcal{C} = \{C_1, \ldots, C_J\}$ and $\mathcal{D} = \{D_1, \ldots, D_J\}$. At a given time k, the mode is delivered by a switching function $\sigma : k \in \mathbb{N} \longmapsto \sigma(k) \in \{1, \ldots, J\} = \mathcal{J}$. A sequence of modes (also called a path) over an interval of time $[k_1, k_2]$, that is $\{\sigma(k_1), \ldots, \sigma(k_2)\}$, is denoted by $\{\sigma\}_{k_1:k_2}$.

For a given switching rule σ, the set of corresponding mode sequences over an interval of time $[k, k+T]$ belongs to $\mathcal{J}^{T+1}$. Let $\mathcal{U}$ be the space of input sequences over the time interval $[0, \infty)$ and $\mathcal{Y}$ the corresponding output space. At time k, for each initial state $x_k \in \mathbb{R}^n$, when the system (1) is driven by the input sequence $\{u\}_{k:k+T} = \{u_k, \dots, u_{k+T}\} \in \mathcal{U}$, for a mode sequence $\{\sigma\}_{k:k+T}$, $\{x(x_k, \sigma, u)\}_{k:k+T}$ refers to the solution of (1) in the interval of time $[k, k+T]$ starting from x_k and $\{y(x_k, \sigma, u)\}_{k:k+T} \in \mathcal{Y}$ refers to the corresponding output sequence in the same interval of time $[k, k+T]$.

For any integer n, $\mathbf{1}_n$ 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 it shall be merely written as $\mathbf{0}$. We introduce the subsequent vectors and matrices:

$$u_{k:k+i} = \begin{pmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+i} \end{pmatrix}, \; y_{k:k+i} = \begin{pmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+i} \end{pmatrix} \tag{2}$$

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

$$\mathcal{O}_{\sigma(k:k+i)} = \begin{pmatrix} C_{\sigma(k)} \\ C_{\sigma(k+1)} A_{\sigma(k)} \\ \vdots \\ C_{\sigma(k+i)} A_{\sigma(k)}^{\sigma(k+i-1)} \end{pmatrix} \tag{3}$$

The matrix $\mathcal{O}_{\sigma(k:k+i)}$ involves the transition matrix defined by

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

Finally, we recursively define the matrix

$$M_{\sigma(k:k+i)} = \begin{pmatrix} D_{\sigma(k)} & \mathbf{0} \\ \mathcal{O}_{\sigma(k:k+i)} B_{\sigma(k)} & M_{\sigma(k+1:k+i)} \end{pmatrix} \tag{4}$$

with

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

Let us notice that the notation $\sigma(k : k+i)$, which points out that the related matrix depends on the sequence $\{\sigma(k), \ldots, \sigma(k+i)\}$ is somehow abusive since σ is defined over $\mathbb{N}$ and not over $\mathbb{N}^{i+1}$. However, since it does not induce confusion, such a notation will be used accordingly for the sake of shortness.

Flatness is closely related to the notions of left invertibility which actually stands for a necessary condition. Roughly speaking, invertibility of a dynamical system is the ability of uniquely determining the input sequence from the output sequence. The works dealing with left invertibility reported in [5] are considered throughout the literature as the pioneering ones. Left invertibility for switched linear systems has been addressed in [6] for continuous-time systems and in [7, 8] for discrete-time systems.
The concept of left inverse systems, related to left invertibility, will play a central role for our purpose. The following definition is in accordance with the papers [7, 8, 9].

Definition 1. A system, with state vector $\widehat{x}_k$, is a left r-delay inverse for (1) if, under identical initial conditions x_0 and identical sequences $\{\sigma\}_{0:\infty}$, there exists a non negative integer r such that, when driven by $y_{k:k+r}$, the equalities $\widehat{x}_{k+r} = x_k$ and $\widehat{u}_{k+r} = u_k$ for all $k \geq 0$ are ensured, $\widehat{u}_k$ being the output of (1) at time k. The non negative integer r is called the inherent delay.

Let us notice that the terminology of r-delay inverse and inherent delay is borrowed from the work [10] which deals with linear systems. Besides, the consideration of the initial condition x_0 stands as a counterpart of the continuous case and the definition of *invertibility at point* x_0 introduced in [11]. Actually, the initial condition x_0 has been disregarded in [10] by assuming that it is zero or that "its effect can be subtracted".

The papers [7, 8, 9] give an explicit form of the left r-delay inverse system for (1). It is recalled below.

$$\left\{ \begin{array}{lcl} \widehat{x}_{k+r+1} & = & P_{\sigma(k:k+r)}\widehat{x}_{k+r} \\ & & +B_{\sigma(k)}I_{m\times r}(M_{\sigma(k:k+r)})^{\dagger}y_{k:k+r} \\ \widehat{u}_{k+r} & = & -I_{m\times r}(M_{\sigma(k:k+r)})^{\dagger}\mathcal{O}_{\sigma(k:k+r)}\widehat{x}_{k+r} \\ & & +I_{m\times r}(M_{\sigma(k:k+r)})^{\dagger}y_{k:k+r} \end{array} \right. \tag{5}$$

with

$$P_{\sigma(k:k+r)} = A_{\sigma(k)} - B_{\sigma(k)}I_{m\times r}(M_{\sigma(k:k+r)})^{\dagger}\mathcal{O}_{\sigma(k:k+r)} \tag{6}$$

The matrix $(M_{\sigma(k:k+r)})^\dagger$ is the classical Moore-Penrose generalized inverse of $M_{\sigma(k:k+r)}$. The matrices $P_{\sigma(k:k+r)}$ are called the left-inverse dynamical matrices.

2.1. Flatness

Definition 2 ([9]). A square ($p = m$) dynamical system is said to be *flat* if there exists a set of independent variables y_k, referred to as flat outputs, such that all system variables can be expressed as a function of the flat output and a finite number of its backward and/or forward shifts. In particular, there exist two functions $\mathcal{F}$ and $\mathcal{G}$ which obey

$$\left\{ \begin{array}{lcl} x_k & = & \mathcal{F}(y_{k+k_\mathcal{F}}, \ldots, y_{k+k'_\mathcal{F}}) \\ u_k & = & \mathcal{G}(y_{k+k_\mathcal{G}}, \ldots, y_{k+k'_\mathcal{G}}) \end{array} \right. \tag{7}$$

where $k_\mathcal{F}$, $k'_\mathcal{F}$, $k_\mathcal{G}$ and $k'_\mathcal{G}$ are $\mathbb{Z}$-valued integers.

The issue of flat output characterization consists in checking whether or not a given output of a dynamical system is flat. Theorem 1 stated in [9] and recalled below gives a first characterization by considering the left-inverse dynamical system (5).

Theorem 1 ([9]). *An output y_k of the system (1) assumed to be square, with left inherent delay r, is a* flat output *if there exists a positive integer $K < \infty$ such that, for all sequences in $\mathcal{J}^{r+K}$, the following equality, involving the product of left-inverse dynamical matrices, applies for all $k \geq 0$:*

$$P_{\sigma(k+K-1:k+K-1+r)} P_{\sigma(k+K-2:k+K-2+r)} \cdots P_{\sigma(k:k+r)} = \mathbf{0} \tag{8}$$

Condition (8) only involves matrices (6) of the left r-delay inverse system (5). Besides, the matrices (6) depend on sequences of modes. Hence, even if σ is arbitrary, the sequences parametrizing two successive matrices involved in the product (8) are related to each other. To cope with this constraint without introducing too heavy notations and to make the subsequent technical developments more explicit, it is convenient to define an auxiliary system and to rewrite Theorem 1 accordingly.

2.2. Auxiliary system

Let us define the auxiliary system of (1) as the switched linear system given by

$$q_{k+1} = Q_{\sigma'(k)} q_k \tag{9}$$

with $q_k \in \mathbb{R}^n$ and σ' a switching rule defined as follows.
Consider the mapping $\phi : \mathcal{J}^{r+1} \to \mathcal{H} = \{1, \dots, J^{r+1}\}$ that assigns to each possible sequence $\{\sigma(k), \dots, \sigma(k+r)\}$ an integer h from the set $\mathcal{H}$ which uniquely identifies the sequence. Then, the switching rule σ' is defined as the function from $\mathbb{N}$ to $\mathcal{H}$ which associates to each integer $k \in \mathbb{N}$ the quantity $\sigma'(k) = \phi(\sigma(k), \dots, \sigma(k+r)) \in \mathcal{H}$. The value $\sigma'(k)$ is the mode of the auxiliary switched linear system (9) and $Q_{\sigma'(k)} = P_{\sigma(k:k+r)}$. We denote by $\mathcal{Q}$ the set of all the matrices Q_h ($h \in \mathcal{H}$).

By considering the auxiliary system (9), we are now in a position of reformulating Theorem 1 which turns into

Theorem 2. *An output y_k of the system (1) assumed to be square, with left inherent delay r, is a* flat output *if there exists a positive integer $K < \infty$ such that, for all sequences $\{\sigma'(k), \dots, \sigma'(k+K-1)\} \in \mathcal{H}^K$, the following equality, involving the product of the dynamical matrices of the auxiliary system* (9)*, applies for all $k \geq 0$:*

$$Q_{\sigma'(k+K-1)} Q_{\sigma'(k+K-2)} \cdots Q_{\sigma'(k)} = \mathbf{0} \tag{10}$$

Proof. The proof is a straightforward consequence of the definition of the auxiliary system. □

The point is that the computational cost of the test (10) grows exponentially with respect to the number K of matrices involved in (10). Besides, no upper bound for K is given. Hence, we cannot a priori know how many tests we should perform. Actually, such a condition should not be seen as a characterization but rather as a definition which stipulates that the inverse dynamics of (1) must be trivial. As a result, it should be convenient to propose a new and complete characterization without prohibitive computational cost. It is the purpose of the subsequent sections and

an alternative to Theorem 2 will be given. It is based on the notion of nilpotent semigroups. This notion will allow us to derive an algorithm, tractable and with fair complexity, for flat output characterization.

3. Flat output characterization and nilpotent semigroups

3.1. Main result

Let us first recall some basics related to nilpotent semigroups.

Definition 3 (Semigroup)**.** A semigroup $\mathcal{S}$ is a set together with an associative internal law.

The semigroup $\mathcal{S}$ is said to be finite if it has a finite number of elements. If $\mathcal{S}$ is a set of matrices, the associative internal law is the matrix multiplication. Let 0 denote the absorbing element of a semigroup when it exists.

Definition 4 (Nilpotent semigroup)**.** A semigroup $\mathcal{S}$ with an absorbing element 0 is said to be nilpotent if there exists an 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, applying the internal law to any t elements of $\mathcal{S}$ amounts to performing the product of t matrices of $\mathcal{S}$. The absorbing element is, in this case, the null matrix. Hereafter, the elements of $\mathcal{S}$ are the matrices of $\mathcal{Q}$.

Theorem 3. *If the matrices of $\mathcal{Q}$ of the auxiliary system* (9) *generate a nilpotent semigroup, then y_k is a flat output.*

Proof. If the matrices of $\mathcal{Q}$ of the auxiliary system (9) generate a nilpotent semigroup, by definition, for any $t-$tuple $(h_1, \dots, h_t) \in \mathcal{H}^t$, t being the class of nilpotency of $\mathcal{Q}$, one has

$$\prod_{i=1}^{t} Q_{h_i} = \mathbf{0} \tag{11}$$

Hence, (10) is fulfilled with $K = t$. As a result, Theorem 2 holds and means that y_k is a flat output. □

Corollary 1. *If the matrices of $\mathcal{Q}$ generate a nilpotent semigroup, the integer K is finite and is upper bounded by the dimension n of the system* (1)*.*

Proof. If the matrices of $\mathcal{Q}$ generate a nilpotent semigroup, the integer K is equal to the class of nilpotency t of the semigroup. The class of nilpotency being actually bounded by the dimension of the matrices involved in the semigroup, K is bounded by the dimension of the matrices of $\mathcal{Q}$, that is precisely n, the dimension of the system (1). □

Remark 1. A necessary condition for the matrices of $\mathcal{Q}$ to generate a nilpotent semigroup is that all the matrices of $\mathcal{Q}$ are nilpotent, that is all their eigenvalues are zero. Indeed, (11) must hold in particular for the $t-$tuples $(h_i, \ldots, h_i)$ for all $i \in \{1, \ldots, t\}$. It should be pointed out that it is not sufficient. Indeed, a set of nilpotent matrices may not generate a nilpotent semigroup and the examples are not singular. For instance, consider the following sets of two elements

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \text{ or } \begin{pmatrix} -2 & -2 & -6 \\ -1 & -4 & -9 \\ 2 & 2 & 6 \end{pmatrix}, \begin{pmatrix} 3 & 1 & 14 \\ 3 & -1 & 10 \\ 0 & -1 & -2 \end{pmatrix}$$

For each set, both matrices are nilpotent. However, they do not generate a nilpotent semigroup.

3.2. Comparative study

The switching rule σ' of the auxiliary system (9) is constrained. Indeed, since by definition $\sigma'(k) = \phi(\sigma(k), \ldots, \sigma(k+r))$ and $\sigma'(k+1) = \phi(\sigma(k+1), \ldots, \sigma(k+r+1))$, it holds that $\sigma'(k)$ and $\sigma'(k+1)$ depend on the common subsequence $\{\sigma(k+1), \ldots, \sigma(k+r)\}$. Hence, $\sigma'(k)$ and $\sigma'(k+1)$ are related one another. Hence, even in the case when the switching rule σ of (1) is arbitrary, given a matrix $Q_{\sigma'(k)} = P_{\sigma(k:k+r)}$, the matrix $Q_{\sigma'(k+1)} = P_{\sigma(k+1:k+1+r)}$ may not take an arbitrary value over $\mathcal{Q}$. To formalize this constraint, it is convenient to introduce a so-called set of feasible transitions.

Definition 5. The set $\Gamma(\sigma'(k))$ of feasible transitions from a mode $\sigma'(k)$ is the set defined by

$$\begin{aligned} \Gamma(\sigma'(k)) \quad = \quad & \{h \in \mathcal{H} : h = \phi(\sigma(k+1), \ldots, \\ & \sigma(k+r+1)), \forall \sigma(k+r+1) \in \mathcal{J}\} \end{aligned} \tag{12}$$

In other words, $\Gamma(\sigma'(k))$ is the set $h \in \mathcal{H}$ which can be reached 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 $\Gamma(\sigma'(k)) \subseteq \mathcal{H}$ which clearly formalizes that σ' is constrained. It is clear that $\Gamma(\sigma'(k))$ can never be the empty set.

Definition 6. A sequence $\{h_1, h_2, \ldots\}$ is said to be admissible if for any $i \geq 1$

$$h_{i+1} \in \Gamma(h_i) \tag{13}$$

Let us introduce the map $\mu : \mathcal{H} \to \mathcal{Q}$ which assigns to each integer $h \in \mathcal{H}$ the matrix $Q_h \in \mathcal{Q}$. The restriction of μ to a particular subset $\Gamma(h)$ of $\mathcal{H}$ is denoted by $\mu_{\Gamma(h)}$.

Definition 7. A sequence of matrices $\{Q_{h_1}, Q_{h_2}, \ldots\}$ ($h_i \in \mathcal{H}$) is said to be admissible if for any $h_i \in \mathcal{H}$

$$Q_{h_{i+1}} \in \mathcal{R}(\mu_{\Gamma(h_i)}) \tag{14}$$

where the notation $\mathcal{R}$ denotes the range of the function. The following proposition applies:

Proposition 1. *The conditions* (10) *and* (11) *are equivalent if and only if*

$$\forall h_i \in \mathcal{H}, \mathcal{R}(\mu_{\Gamma(h_i)}) = \mathcal{Q} \tag{15}$$

Proof. The statement (11) $\Rightarrow$ (10) is always true regardless of the condition (15). Hence, it must be shown that (10) implies (11) provided that (15) is fulfilled. The condition $\forall h_i,\ \mathcal{R}(\mu_{\Gamma(h_i)}) = \mathcal{Q}$ means that, for any arbitrary mode $h_i \in \mathcal{H}$, $Q_{h_{i+1}}$ can be any matrix in $\mathcal{Q}$. Hence, for any $t-$tuple $(h_1, \ldots, h_t)$, the sequence $\{Q_{h_1}, \ldots, Q_{h_t}\}$ is an admissible sequence for (9). Finally, the set of products $Q_{h_1} \cdots Q_{h_t}$ for all $t-$tuples $(h_1, \ldots, h_t)$ coincides with the set of products (10) for all $k \geq 0$. That completes the proof. $\square$

Remark 2. It can be easily seen that (15) is always satisfied for at least two particular cases: when the inherent delay r is equal to zero or if it is equal to one and that C does not depend on σ. However, (15) may also apply regardless of the inherent delay. In Section 5, examples will illustrate how and when (15) plays or does not play a role. Such a condition must not be considered as necessary for Theorem 3 to apply.

3.3. Computational issues

In this section, we propose an algorithm that allows us to check whether or not a set of matrices generates a nilpotent semigroup that is, whether Theorem 3 is fulfilled. It is motivated by Levitsky's theorem (Theorem 2.1.7 stated in [12]).

Theorem 4 (Levitsky's theorem)**.** *Any semigroup of nilpotent matrices can be triangularized.*

In other words, all the matrices of the same nilpotent semigroup can be rewritten as upper triangular matrices with zeros on the diagonal up to a common change of basis. The consequence of this theorem is central for our purpose. Indeed, determining whether or not the matrices of $\mathcal{Q}$ generate a nilpotent semigroup amounts to checking whether or not $\mathcal{Q}$ can be simultaneously triangularized. It is a necessary and sufficient condition. The approach we propose to check Theorem 4 and thus Theorem 3 is inspired from the general triangularization method provided in [13] and corresponds to Algorithm 1. Some peculiarities that apply to our special case are addressed in order to come up with a complete algorithm for flat output characterization.

The algorithm is expected to return a change of basis S that simultaneously triangularizes the set $\mathcal{Q}$.

It is worth pointing out that different sequences $\{\sigma(k), \ldots, \sigma(k+r)\}$ of (1) and so different modes $\sigma'(k) = \phi(\sigma(k), \ldots, \sigma(k+r))$ of (9) might lead to identical matrices $Q_{\sigma'(k)}$. As a result, $\mathcal{Q}$ is a multiset[1] and we should only consider distinct matrices of $\mathcal{Q}$ to reduce the computational cost. We denote by L (with $L \leq J^{r+1}$) the number of distinct matrices of $\mathcal{Q}$ and by Z_l $(l = 1, \ldots, L)$ the elements of the multiset.

Algorithm 1.

$\triangleright$ *Initialization*

for $l=1$ **to** L **do**

  $T_l \leftarrow Z_l$

end for

$S_2 \leftarrow \mathbf{1}_n$; $S_1 \leftarrow \mathbf{0}_{n\times 0}$

$\triangleright$ *Triangularization*

for $i \leftarrow 1$ **to** $n-1$ **do**

  $v_i \leftarrow$ *One eigenvector common to the matrices* $\{T_l\}$, $l = 1, \ldots, L$

  if v_i *does not exist* **then**

    $\triangleright$ *No triangularization basis exists*

    return

  end if

  $w_i \leftarrow S_2 \cdot v_i$

  $S_1 \leftarrow \begin{pmatrix} S_1 & w_i \end{pmatrix}$

[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.

15: $\quad S_2 \leftarrow$ *matrix whose column vectors are vectors that extend* S_1 *to a basis*
16: $\quad S \leftarrow (S_1 \quad S_2)$
17: $\quad I_1 \leftarrow \begin{pmatrix} \mathbf{0}_i & \mathbf{1}_{n-i} \end{pmatrix}$
18: $\quad I_2 \leftarrow \begin{pmatrix} \mathbf{0}_i \\ \mathbf{1}_{n-i} \end{pmatrix}$
19: $\quad S^{-1} \leftarrow$ *inverse of* S
20: $\quad$ ***for*** $l \leftarrow 1$ ***to*** L ***do***
21: $\quad\quad T_l \leftarrow I_1 S^{-1} Z_l S I_2$
22: $\quad$ ***end for***
23: ***end for***
24: ***return*** S

The following comments are in order:

- Line 1 to Line 5 correspond to the initialization. Matrices T_l ($l = 1, \ldots, L$) play the role of the auxiliary variables and are initialized at the beginning with the matrices Z_l.

- At most $n-1$ successive loops from Line 7 to Line 23 are performed as stressed by Corollary 1.

- Line 8 corresponds to the first step of a given loop i. It consists in finding out an eigenvector v_i which is common to the matrices T_l ($l = 1, \ldots, L$). Consequently, in the first loop, v_1 is a common eigenvector of the matrices Z_l. It is worth pointing out that if this step fails in the loop i, it means that the matrices T_l ($l = 1, \ldots, L$) do not have any common eigenvector v_i and the algorithm stops. From Levitsky's theorem stating a necessary and sufficient condition, it can be concluded that the matrices Z_l do not generate a nilpotent semigroup and Theorem 3 is not fulfilled.

- Line 14 describes the fact that the final change of basis S is built column after column. Each new vector v_i carried out in the loop v_i is added (actually after a change of basis notified at Line 13) resulting in a matrix $S_1 = (v_1 \cdots v_i)$. When $i = n$ then $S = (S_1 \ S_2)$, S_1 and S_2 resulting from the loop $n-1$.

- At Line 15, S_1 must be extended to a basis. By extension, it is meant a set of $n-i$ vectors w_j so that $S = (S_1 \ w_1 \cdots w_{n-i})$ is full rank. The matrix S_2 is precisely $S_2 = (w_1 \cdots w_{n-i})$.

- Line 21 performs the current change of basis S to the matrices Z_l. The multiplications by I_1 and I_2 merely correspond to the extraction of a square matrix of dimension $n-i$ from the matrix $S^{-1}Z_lS$ that is, the first i rows and columns of $S^{-1}Z_lS$ are removed. A new set of matrices T_l is thereby obtained. A new loop can restart from Line 7.

Such an algorithm is quite general. However, the determination of a common eigenvector at Line 8 and the extension to a basis at Line 15 can be particularized to our special context.

3.3.1. Determination of a common eigenvector

According to Remark 1, a necessary condition for the set $\mathcal{Q}$ to generate a nilpotent semigroup is that all the matrices Z_l are nilpotent, that is all their eigenvalues are zero. The matrices T_l $(l = 1, \ldots, L)$ are initialized to Z_l $(l = 1, \ldots, L)$ and they are updated at each loop i through a change of basis S at Line 21 which preserves the eigenvalues. Hence, the eigenvalues of the T_l's are all zero whatever the loop $i \geq 0$ is. Consequently, for any v_i, $(i = 1, \ldots, n)$ and $\forall T_l \in \mathcal{T}$, $T_l v_i = 0$ holds. Hence, v_i is a non zero solution of

$$Rv_i = \mathbf{0} \text{ with } R = \begin{pmatrix} T_1 \\ \vdots \\ T_L \end{pmatrix} \tag{16}$$

As a result, v_i is a non zero vector of the null space of R denoted $\ker(R)$.

3.3.2. Extension to a basis

At Line 15, S_1 must be extended to a basis. We must thereby find out a set of $n-i$ vectors w_j so that $S = \begin{pmatrix} S_1 & w_1 & \cdots & w_{n-i} \end{pmatrix}$ is full rank.

It can be obtained by determining a basis of the kernel of the transpose of S_1. In other words, $S_2 = (w_1 \cdots w_{n-i})$ can be any basis of $\ker(S_1')$ where the symbol $'$ stands for transposition.

3.4. Complexity

All the operations can easily be performed by software involving the usual built-in functions. As an example, we give the corresponding Matlab source in Listing 1 of the Appendix. The complexity of the condition (8) in Theorem 1 is $O(J^{r+K}Kn^3)$. The problem lies in that the complexity is exponential with respect to the number of matrices K involved in the product, which can be large. Let us estimate the computational cost of the flat output characterization approach stated in Theorem 3 and so based on nilpotent semigroups. To this end, let us examine Algorithm 1. Considering a given loop, the most complex operations are performed at Lines 8, 15, 19 and 21. Lines 8 and 15 consist in determining the kernel of a matrix. It is usually based on a singular value decomposition of a $\mathbb{R}^{a\times b}$ matrix for which the usual algorithms have complexity with $O(4ab^2+8b^3)$ exist. Line 19 which involves a matrix inversion can resort to algorithms with complexity $O(n^3)$. Line 21 is a change of basis. The multiplications by I_1 and I_2 can be omitted since the effect is merely to extract a square matrix of dimension $n-i$ from the matrix $S^{-1}Z_lS$ so it can be done much more efficiently than by a matrix multiplication. Therefore, the two operations to be considered are the two matrix multiplications of Z_l by S and S^{-1}. Matrix multiplications have complexity at most $O(n^3)$. Therefore, the complexity of Lines 20 to 22 is $O(Ln^3)$ (or $O(J^{r+1}n^3)$ if we consider the upper bounding $L \leq J^{r+1}$ due to the consideration of the multiset). This part of the code is the one with the largest complexity. The operations are repeated over at most n loops (the maximum class of nilpotency according to Corollary 1). Therefore, the global complexity of Algorithm 1 related to Theorem 3 is $O(J^{r+1}n^4)$. It is clearly an improvement insofar as the complexity is no longer exponential with respect to the parameter K. Let us notice that the exponential complexity with respect to the inherent delay r is structural and cannot be circumvented.

As it turns out, the results can be interestingly extended to LPV systems as shown in the following.

4. Extension to LPV systems

The flat output characterization based on nilpotent semigroups for LPV systems is valuable for two major reasons. First, flat output characterization of LPV systems has never been addressed so far in the literature. Secondly, the characterization through Theorem 1 cannot conveniently be done numerically for LPV systems since that requires checking an infinite number of possible products. Indeed, σ taking values in a continuum, the number of sequences $\{\sigma\}$ in Theorem 1 or sequences $\{\sigma'\}$ in Theorem 2 would be infinite. The system (1) can be viewed as an LPV system as soon as we consider that the switching rule σ is replaced by a function which takes values in a continuum. If so, the sets $\mathcal{J}$, $\mathcal{H}$, $\Gamma(\sigma'(k))$, $\mathcal{Q}$ must be considered as uncountable sets. The $r-$delay inverse system (5), the auxiliary system (9) together with the mapping ϕ still make sense as soon as σ' is considered as a function, similarly to σ, taking values in a continuum. Besides and most importantly, it turns out that both the semigroups (Definition 3) and the nilpotent semigroups (Definition 4) are still well defined for an uncountable set $\mathcal{S}$. As a result, Theorem 3 still applies.

On the other hand, Levitsky's Theorem, which allows for checking whether Theorem 3 is fulfilled, applies for any semigroup, including semigroups with infinite cardinality, which is especially the case for LPV systems. It is recalled that Levitsky's theorem asserts that "$\mathcal{Q}$ generates a nilpotent semigroup if the matrices of $\mathcal{Q}$ can be simultaneously triangularized". The point is that we cannot perform a simultaneous triangularization of an infinite number of matrices. However, such a problem can be handled by noticing that it is equivalent to checking whether the matrix $Q_{\sigma'(k)}$ can be triangularized with a change of basis that does not depend on $\sigma'(k)$. From this consideration and by combining Theorem 3 and Theorem 4, the following theorem holds for characterizing flat outputs of LPV systems

Theorem 5. *If the matrix $Q_{\sigma'(k)}$ of the auxiliary system* (9) *can be triangularized independently of $\sigma'(k)$, then y_k is a flat output.*

Algorithm 1 still applies up to some minor modifications. The loops at Line 2 and Line 20 can be removed or equivalently, L can be set to $L = 1$. Besides, the determination of a common eigenvector at Line 8, that is the search for a non zero vector v_i of $\ker(R)$ as explained in Subsection 3.3.1 turns into the search for a non zero vector v_i of $\ker(T_1)$ (since $L = 1$ and so $R = T_1$) independent of $\sigma'(k)$. Line 8 has to be replaced by

8: $v_i \leftarrow$ One eigenvector of T_1 independent of $\sigma'(k)$

5. Illustrative examples

5.1. Example 1: practical example

We consider the dynamics of a vehicle[2] described by the usual nonlinear state-space equations [14].

$$\begin{aligned}\dot{x}_1(t) &= x_2(t),\\ \dot{x}_2(t) &= \frac{u(t)C_{mot}}{mR} - \frac{1}{2}\frac{\rho C_x S}{m}x_2(t)^2\end{aligned} \tag{17}$$

m is the mass of the vehicle. x_1 and x_2 denote respectively the position and the velocity of the vehicle. C_{mot} is the torque over the wheel delivered by the motor. The input $u(t) \in [0;1]$ is the duty cycle of the driving period during which the engine is on. Therefore $u(t)C_{mot}$ is the average torque over the wheel. R is the radius of the wheel. S is the frontal area of the vehicle. The aerodynamic drag coefficient C_x and the air density ρ characterize the aerodynamic frictions. The nonlinear discrete-time state space model with state variables $x_k^{(1)}$ and $x_k^{(2)}$ denoting respectively the position and the velocity at time k read

$$x_{k+1}^{(1)} = x_k^{(1)} + T_k x_k^{(2)} \tag{18}$$

$$x_{k+1}^{(2)} = x_k^{(2)} + T_k\frac{C_{mot}u_k}{mR} - T_k\frac{1}{2}\frac{\rho C_x S}{m}(x_k^{(2)})^2 \tag{19}$$

In such an embedded control application, the sampling period is a strictly positive time-varying quantity that can be scheduled in order to reduce the processor load.

The equations (19) can be rewritten as (1). The state space matrices read

$$A_{\sigma(k)} = \begin{pmatrix} 1 & \sigma_k^{(1)} \\ 0 & \sigma_k^{(2)} \end{pmatrix},\ B_{\sigma(k)} = \begin{pmatrix} 0 \\ \sigma_k^{(3)} \end{pmatrix},\ D_{\sigma(k)} = 0$$

with $\sigma_k^{(1)} = T_k$, $\sigma_k^{(2)} = 1 - T_k\frac{1}{2}\frac{\rho C_x S}{m}x_k^{(2)}$ and $\sigma_k^{(3)} = T_k\frac{C_{mot}}{mR}$. The sampling period T_k and the velocity lies in a bounded range so that $\sigma(k) = (\sigma_k^{(1)}, \sigma_k^{(2)}, \sigma_k^{(3)})$

[2]It is actually the prototype developed by the Research Center for Automatic Control of Nancy which is annually involved in the European Shell Eco-Marathon race in the category Plug-in (battery) category

lies in a hypercube $\Sigma = \Sigma_1 \times \Sigma_2 \times \Sigma_3$. The range of Σ_i ($i = 1, 2, 3$) is determined with respect to the minimum and maximum value of the sampling period T_k and of the velocity $x_k^{(2)}$. Whether the sets Σ_1 and Σ_3 are uncountable or countable ones according to the sampling period scheduling strategy, the system can be viewed as an LPV system or a mixed switched linear/LPV system.

We consider first the velocity $x^{(2)}$ as the measured output. Thus,

$$C_{\sigma(k)} = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

The system being a SISO one and since, for all $\sigma(k) \in \Sigma$, $D_{\sigma(k)} = 0$ and for all $(\sigma(k), \sigma(k+1)) \in \Sigma^2$, it holds that $C_{\sigma(k+1)} B_{\sigma(k)} = \sigma_k^{(3)}$ which clearly never vanishes, the inherent delay is $r = 1$. The set $\mathcal{Q}$ of the dynamical matrices of the left-inverse system (5) are the matrices $Q_{\sigma'(k)} = P_{\sigma(k:k+1)} = A_{\sigma(k)} - B_{\sigma(k)} (\sigma_k^{(3)})^{-1} C_{\sigma(k+1)} A_{\sigma(k)}$. It turns out that the set $\mathcal{Q}$ reduces to one matrix

$$Q_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Since Q_1 is not nilpotent, neither Theorem 5 nor Theorem 2 are fulfilled and y_k is not a flat output.

Now, we consider the position $x^{(1)}$ as the measured output. Thus,

$$C_{\sigma(k)} = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

In such a case, y_k is a flat output. Indeed, Definition 2 is satisfied with (7) which reads

$$\begin{cases} x_k^{(1)} &= y_k \\ x_k^{(2)} &= (T_k)^{-1}(y_{k+1} - y_k) \\ u_k &= (T_k \frac{C_{mot}}{mR})^{-1}\big((T_{k+1})^{-1}(y_{k+2} - y_{k+1}) - (T_k)^{-1}(y_{k+1} - y_k) \\ & \quad + T_k \frac{1}{2} \frac{\rho C_x S}{m}((T_k)^{-1}(y_{k+1} - y_k))^2\big) \end{cases}$$

Let us notice that even for a small system, directly applying the definition is not trivial and becomes prohibitive when the dimension grows up. The system being a SISO one, from $D_{\sigma(k)} = 0, C_{\sigma(k+1)} B_{\sigma(k)} = 0$ and $C_{\sigma(k+2)} A_{\sigma(k+1)} B_{\sigma(k)} = \sigma_{k+1}^{(1)} \cdot \sigma_k^{(3)} = T_{k+1} T_k \frac{C_{mot}}{mR}$ which clearly never vanishes, it is inferred that the inherent delay is $r = 2$. The set $\mathcal{Q}$ of the dynamical matrices of the left-inverse system (5) are the matrices $Q_{\sigma'(k)} = P_{\sigma(k:k+1)} = A_{\sigma(k)} - B_{\sigma(k)}C_{\sigma(k+1)}A_{\sigma(k+1)}A_{\sigma(k)}$ which read

$$Q_{\sigma'(k)} = \begin{pmatrix} 1 & \sigma_k^{(1)} \\ -(\sigma_{k+1}^{(1)})^{-1} & -(\sigma_{k+1}^{(1)})^{-1}\sigma_k^{(1)} \end{pmatrix}$$

If $\sigma_k^{(1)}$ takes values in a continuum, the set $\mathcal{Q}$ is uncountable. As it turns out, Algorithm 1 fails and so Theorem 5 is not fulfilled. And yet, as proved above, y_k is a flat output. By the way, Theorem 2 is fulfilled, (10) being checked here by working out the product symbolically. This is explained by the fact that condition (15) of Proposition 1 is not satisfied. Indeed, let $\sigma^*(k)$, $\sigma^*(k+1)$ and $\sigma^*(k+2)$ be respectively the actual modes at times k, $k+1$ and $k+2$. The set $\Gamma(\sigma'(k))$ of feasible transitions from the mode $\sigma'(k)$ to the mode $\sigma'(k+1)$ is the set $\{\sigma(k+1) = \sigma^*(k+1), \sigma(k+2) = \sigma^*(k+2), \sigma(k+3) \in \Sigma\}$. Thus, $Q_{\sigma'(k+1)}$ cannot be any matrix of $\mathcal{Q}$ since $\sigma_{k+1}^{(1)}$ is imposed in $Q_{\sigma'(k+1)}$. Finally, let us consider the situation when the sampling period is constant. Thus, it holds that $\sigma_k^{(1)} = \sigma_{k+1}^{(1)} = \sigma^*$ for all k. In such a case, the matrix $Q_{\sigma'(k)}$ reads

$$Q_{\sigma'(k)} = \begin{pmatrix} 1 & \sigma^* \\ -(\sigma^*)^{-1} & -1 \end{pmatrix}$$

Hence, the set $\mathcal{Q}$ generates a nilpotent semigroup with class of nilpotency $t = 2$ and we infer that y_k is a flat output for the LPV system (19), $\sigma_k^{(2)}$ still being time-varying.

5.2. Further example: a switching system

Consider the switched linear system of the form (1). The dimension is $n = 4$, the switching rule σ, not detailed here, is assumed to deliver arbitrary sequences and the number of modes is $J = 3$. According to the mode, the state space matrices numerically read

$$A_1 = \begin{pmatrix} -1 & -0.5 & -0.5 & 0 \\ 1 & 1.5 & 1.5 & 0 \\ 1 & 0.5 & 0.5 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} -1 & -1 & -1 & -0.5 \\ 1 & 2 & 2 & 0.5 \\ 1 & 1 & 1 & 1.5 \\ 1 & 0 & 1 & 0 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} -1 & -2.5 & -2.5 & -2 \\ 1 & 3.5 & 3.5 & 2 \\ 1 & 2.5 & 2.5 & 3 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$B_1 = B_2 = B_3 = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}^T$$

$$C_1 = C_2 = C_3 = \begin{pmatrix} 2 & 1 & 1 & 0 \end{pmatrix}$$

$$D_1 = D_2 = D_3 = 0$$

For SISO systems, the inherent delay r can be inferred by merely figuring out that $D_i = 0$ for all $i \in \{1, 2, 3\}$, $C_i B_j = 0$ for all $(i, j) \in \{1, 2, 3\}^2$ and $C_i A_j B_l \neq 0$ for all $(i, j, l) \in \{1, 2, 3\}^3$. Therefore, $r = 2$. Let us derive the corresponding auxiliary system as defined in Section 2.2. To this end, we must define the mapping ϕ. The number of possible sequences over any interval of time $[k : k + r]$ is $J^{r+1} = 3^{2+1} = 27$ and

$$\begin{array}{ll} \phi(\{1,1,1\}) = 1 & \cdots \\ \phi(\{1,1,2\}) = 2 & \phi(\{3,3,1\}) = 25 \\ \phi(\{1,1,3\}) = 3 & \phi(\{3,3,2\}) = 26 \\ \cdots & \phi(\{3,3,3\}) = 27 \end{array}$$

The set Γ of feasible transitions obeys

$$\begin{array}{ll} \Gamma(1) = \{1,2,3\} & \cdots \\ \Gamma(2) = \{4,5,6\} & \Gamma(26) = \{23,24,25\} \\ \cdots & \Gamma(27) = \{25,26,27\} \end{array}$$

It turns out that $\mathcal{Q}$ has only $L = 3$ distinct matrices denoted Z_l $(l = 1, 2, 3)$ and thus is viewed here as a multiset. The matrices numerically read

$$Z_1 = \begin{pmatrix} -1 & -0.5 & -0.5 & 0 \\ 1 & 1.5 & 1.5 & 0 \\ 1 & 0.5 & 0.5 & 1 \\ -2 & -2 & -2 & -1 \end{pmatrix},$$

$$Z_2 = \begin{pmatrix} -1 & -1 & -1 & -0.5 \\ 1 & 2 & 2 & 0.5 \\ 1 & 1 & 1 & 1.5 \\ -2 & -3 & -3 & -2 \end{pmatrix},$$

$$Z_3 = \begin{pmatrix} -1 & -2.5 & -2.5 & -2 \\ 1 & 3.5 & 3.5 & 2 \\ 1 & 2.5 & 2.5 & 3 \\ -2 & -6 & -6 & -5 \end{pmatrix}$$

It holds that

$$\mu(\Gamma(1)) = \cdots = \mu(\Gamma(27)) = \mathcal{Q}$$

and so, Proposition 1 is fulfilled.
Finally, it turns out that Algorithm 1 succeeds and returns the following change of basis S

$$S = \begin{pmatrix} 0 & 0.3780 & -0.9258 & 0 \\ 0.7071 & -0.3780 & -0.1543 & 0.5774 \\ -0.7071 & -0.3780 & -0.1543 & 0.5774 \\ 0 & 0.7559 & 0.3086 & 0.5774 \end{pmatrix}$$

As a consequence, based on Theorem 3 and Theorem 4, we conclude that y_k is a flat output.
To end up with this example, let us compare the complexities. The complexity of the condition of Theorem 2 would be $O(J^{r+K}Kn^3) = O(3^{2+3}3 \cdot 3^3) = O(1594323)$ while the complexity of the approach derived from Theorem 3 is $O(J^{r+1}n^4) = O(3^{2+1}3^4) = O(2187)$.

5.3. Further example: an LPV system

In this example, we consider (1) as an LPV system where $\sigma(k)$ takes values in a continuum.

The state space matrices read

$$A_{\sigma(k)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \sigma_k^{(1)} & \sigma_k^{(2)} & 1 & 0 \end{pmatrix},\ B_{\sigma(k)} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

$$C_{\sigma(k)} = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix},\ D_{\sigma(k)} = 0$$

Since it is a SISO system, the inherent delay r can be inferred by merely figuring out that for all $\sigma(k)$, $D_{\sigma(k)} = 0$ and $C_{\sigma(k+1)}B_{\sigma(k)} = 1 \neq 0$. Therefore, the inherent delay is $r = 1$. The matrix $Q_{\sigma'(k)} = P_{\sigma(k:k+1)} = A_{\sigma(k)} - BCA_{\sigma(k)}$ reads

$$Q_{\sigma'(k)} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ \sigma_k^{(1)} & \sigma_k^{(2)} & 1 & 0 \end{pmatrix}$$

It turns out that Theorem 5 is fulfilled with a triangularization basis S which numerically reads

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

6. Conclusion

We have provided a condition to characterize flat outputs of switched linear discrete-time systems. The condition is based on the notion of nilpotent semigroups. A complete and tractable recursive algorithm has been provided to perform the characterization. It stops after a finite number of steps bounded by the dimension of the system. Next, an extension to LPV discrete-time systems has been done. This paper gives a serious alternative to an existing computationally prohibitive approach which was merely derived from the definition of flatness.

Appendix A. Matlab code

Listing 1: Matlab code that corresponds to Algorithm 1. The command `null` is used to obtain the kernel (null space) of a matrix

```
function S=GetSimultaneousTrig(Z)
    [m,n,L]=size(Z);
    S2=eye(n);
    k=0;
    T=Z;
    S1=zeros(m,0);
    for i=1:n-1
        %Determining a common eigenvector
        clear R;
```

```
        R=T(:,:,1);
        for l=2:L
            R=vertcat(R,T(:,:,l));
        end
        commoneigenvectors=null(R);
        if(length(commoneigenvectors)==0)
            display('No basis exists.');
            break;
        end
        %Constructing the change of basis
        v=commoneigenvectors(:,1);
        vkp1=S2*v;
        S1=[S1,vkp1];
        %Extending the set of column vectors of S1 into
        %a basis
        S2=null(S1');
        %Preparing the matrices T for the next iteration
        S=[S1,S2];
        Sinv=inv(S);
        clear T
        for l=1:L
           tmp =Sinv*Z(:,:,l)*S;
           T(:,:,l)=tmp(i+1:n,i+1:n);
        end
    end
end
```

References

[1] 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.

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

[3] J. Levine and D. V. Nguyen. Flat output characterization for linear systems using polynomial matrices. *Systems and control Letters*, 48:69–75, 2003.

[4] K. Sako. 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.

[5] R. W. Brockett and M. D. Mesarovic. The reproducibility of multivariable systems. *J. Math. Anal. Appl.*, 11:548–563, July 1965.

[6] L. Vu and D. Liberzon. Invertibility of switched linear systems. *Automatica*, 44(4):949–958, 2008.

[7] 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.

[8] G. Millérioux and J. Daafouz. *Proc. of the 10th International Conference on Hybrid Systems: Computation and Control (HSCC'07)*, volume 44, chapter Invertibility and Flatness of Switched Linear Discrete-time Systems, pages 714–717. Springer Verlag, Pisa, Italy, April 2007.

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

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

[11] R. M. Hirschorn. Invertibility of nonlinear control systems. *SIAM J. Control and Optimization*, 17(2), March 1979.

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

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

[14] L. Guzzella and A. Sciarretta. *Vehicle propulsion systems : introduction to modeling and optimization.* Springer Berlin, 2005.