



On the notion of persistence of excitation for linear switched systems

Mihály Petreczky, Laurent Bako

► To cite this version:

Mihály Petreczky, Laurent Bako. On the notion of persistence of excitation for linear switched systems. *Nonlinear Analysis: Hybrid Systems*, 2023, 48, pp.101308. 10.1016/j.nahs.2022.101308. hal-03945535

HAL Id: hal-03945535

<https://hal.science/hal-03945535>

Submitted on 18 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the notion of persistence of excitation for linear switched systems

Mihály Petreczky^a, Laurent Bako^b

^a*Mihály Petreczky (Corresponding author) is with Centre de Recherche en Informatique, Signal et Automatique de Lille (CRISTAL), France*

^b*Univ Lyon, Ecole Centrale Lyon, INSA Lyon, Université Claude Bernard Lyon 1, CNRS, Ampère, UMR 5005, 69130 Ecully, France*

Abstract

The paper formulates the concept of persistence of excitation for discrete-time linear switched systems, and provides sufficient conditions for an input signal to be persistently exciting. Persistence of excitation is formulated as a property of the input signal, and it is not tied to any specific identification algorithm. The results of the paper rely on realization theory and on the notion of Markov-parameters for linear switched systems.

1. Introduction

The paper formulates the concept of persistence of excitation for *discrete-time linear switched systems* (abbreviated by DTLSSs). DTLSSs are one of the simplest and best studied classes of discrete-time switched systems, [1].

A DTLSS is a discrete-time switched system, such that the continuous sub-system associated with each discrete state is linear. The switching signal is viewed as an external input, and all linear systems live on the same input-output- and state-space.

We define persistence of excitation for input signals. More precisely, we will call an input signal persistently exciting for an input-output map, if the response of the input-output map to that particular input determines the input-output map uniquely. In other words, the knowledge of the output response to a persistently exciting input should be sufficient to predict the response to **any** input.

Persistence of excitation is essential for system identification and adaptive control. Normally, in system identification the system of interest is tested only for one input sequence. One of the reason for this is that our notion of the system entails a fixed initial state. However, any experiment changes that particular initial state and it is in general not clear how to reset the system to a particular initial state. The objective is to find a system model based on the response to the chosen input. However, the knowledge of a model of the system (including the initial state) immediately implies that the response of the system to **any** input is known. Hence, intuitively it is clear that persistence of excitation of the input signal is a prerequisite for a successful identification of a model.

Note that persistence of excitation is a joint property of the input and of the input-output map. That is, a particular input might be persistently exciting for a particular system and might fail to be persistently exciting for another system. In fact, it is not a priori clear if any system admits a persistently exciting input. This calls for investigating classes of inputs which are persistently exciting for some broad classes of systems.

In the existing literature, persistence of excitation is often defined as a specific property of the measurements which is **sufficient** for the correctness of **some** identification algorithm. In contrast, in this paper we propose a definition of persistence of excitation which is **necessary** for the correctness of **any** identification algorithm. Obviously, the two approaches are complementary. In fact, we hope that the results of this paper can serve as a starting point to derive persistence of excitation conditions for specific identification algorithms.

URL: mihaly.petreczky@centralelille.fr (Mihály Petreczky), laurent.bako@ec-lyon.fr (Laurent Bako)

Contribution of the paper We define persistence of excitation for finite input sequences and persistence of excitation for infinite input sequences.

We show that for every input-output map which is realizable by a reversible DTLSS, there exists a finite input sequence which is persistently exciting for that particular input-output map. A reversible DTLSS is a DTLSS such that its continuous dynamics is invertible. Such systems arise naturally by sampling continuous-time systems. In addition, we define the class of reversible input-output maps and show that there is a finite input sequence which is persistently exciting for all the input-output maps of that class. Moreover, we present a procedure for constructing such an input sequence.

We show that there exists a class of infinite input sequences which are persistently exciting for all the input-output maps which are realizable by a *stable* DTLSS. The conditions which the input sequence must satisfy is that each subsequence occurs there infinitely often (i.e. the switching signal is rich enough) and that the continuous input is a colored noise. Hence, this result is consistent with the classical result for linear systems.

It might be appealing to interpret the conditions above as ones which ensure that one stays in every discrete mode long enough and the continuous input is persistently exciting in the classical sense. One could then try to identify the linear subsystems separately and merge the results. Unfortunately, such an interpretation is in general incorrect. The reason for this is that there exists a broad class of input-output maps which can be realized by a linear switched system but not by a switched system whose linear subsystems are minimal, [2]. The above scheme obviously would not work for such systems. In fact, for such systems one has to test the system's response not only for each discrete mode, but for each combination of discrete modes.

The main idea behind the definition of persistence of excitation and the subsequent results is as follows. From realization theory [2] we know that the knowledge of (finitely many) Markov-parameters of the input-output map is sufficient for computing a DTLSS realization of that map. Hence, if the response of the input-output map to a particular input allows us to compute the necessary Markov-parameters, then we can compute a DTLSS representation of that map. This can serve as a definition of persistence of excitation. We call an input sequence persistently exciting, if the Markov-parameters of the input-output map can be computed from the response of the map to that input. We call an infinite sequence input persistently exciting, if from a large enough finite initial part of the response one can compute an arbitrarily precise approximation of the Markov-parameters. Since the realization algorithm for DTLSS is continuous in the Markov-parameters, it means that a persistently exciting infinite input sequence allows the computation of an arbitrarily precise approximation of a DTLSS realizing the input-output map.

Motivation of the system class The class of DTLSSs is the simplest and perhaps the best studied class of discrete-time switched systems.

In addition to its practical relevance, it also serves as a convenient starting point for theoretical investigations. In particular, any piecewise-affine hybrid system can be viewed as a feedback interconnection of a DTLSS with an event generating device. Hence, identification of a piecewise-affine system is related to the problem of closed-loop identification of a DTLSS. For the latter, it is indispensable to have a good notion of persistence of excitation. For this reason, we believe that the results of the paper will be relevant not only for identification of DTLSSs, but also for identification of piecewise-affine hybrid systems with autonomous switching.

Related work Identification of hybrid systems is an active research area, with several significant contributions, without claiming completeness see [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 5, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22] and the references therein. While enormous progress was made in terms of efficient identification algorithms, the fundamental theoretical limitations and properties of these algorithms are still only partially understood.

In particular, there are few results on consistency of system identification algorithms, i.e., [3, 8, 23, 24, 20, 25, 19, 17], and the ones which are available tend to concentrate on switched ARX systems [3, 8, 23, 24, 20, 25, 19].

The papers [3, 8, 23, 24, 20, 25, 19] impose conditions on the measured data, i.e., inputs, outputs, (possibly switching signals), which are sufficient for consistency of some system identification algorithm. These conditions could be viewed as persistence of excitation conditions, and they are referred to as such in some of the cited papers. In particular, the papers [3, 8, 23, 24, 20, 25, 19] impose conditions on the measured data which ensures that there are no multiple switched ARX

models which fit the data. However, this approach cannot be extended to state-space representations, as they are never unique: application of a state-space isomorphism always yields a state-space representation which is different from the original one but generates exactly the same data. The paper [17] imposes a condition that certain matrices formed by the underlying hidden states are full rank.

The common drawback of the conditions of the papers [3, 8, 23, 24, 20, 25, 19, 17] is that these conditions do not depend only on the input and switching sequence, but also on the output and even the underlying unobserved state. However, it is not clear how to design the system identification experiment so that the generated data satisfies those conditions: even if the inputs can be chosen arbitrarily, there are no guarantees that the resulting output will satisfy those persistence of excitation conditions. That is, the definition of persistence of excitation proposed in the cited literature does not possess one important feature of the traditional definitions of persistence of excitation, namely, the existence of a suitable input guaranteeing that the data from the system identification experiment is persistently exciting. This means that the conditions of [3, 8, 23, 24, 20, 25, 19, 17] are not directly helpful for designing system identification experiments. In principle, it could not be excluded that in general it is not possible to design good system identification experiments which would result in persistently exciting data as defined in [3, 8, 23, 24, 20, 25, 19, 17].

Fortunately, the results of the paper show that this is not the case, and it is possible to design system identification experiments which generate rich enough data for identification. That is, the main contribution of the present paper with respect to the existing literature lies in proving formally that for linear switched state-space models it is possible to design a system identification experiment which is persistently exciting, i.e., which allows the design of consistent system identification algorithms. By designing a system identification experiment we mean choosing an input and a switching signal. In other words, we show that for any linear switched system satisfying some mild conditions, there exists a switching signal and an input which allows us to identify the system parameters. To the best of our knowledge, this result is new.

For linear systems, persistence of excitation has thoroughly been investigated, see for example [26, 27] and the references therein.

The paper is an extension of [28]. The main difference with respect to [28] is that the present paper contains detailed proofs and the presentation has been improved. A preliminary version of this paper appeared in the form of a technical report [29].

Outline of the paper §2 presents the formal definition of DTLSSs and it formulates the major system-theoretic concepts for this system class. §3 presents a brief overview of realization theory for DTLSSs. §4 presents the main contribution of the paper.

Notation 1. Denote by $\mathbb{N}$ the set of natural numbers including 0. The notation described below is standard in automata theory, see [30, 31]. Consider a set X which will be called the alphabet. Denote by X^* the set of finite sequences of elements of X . Finite sequences of elements of X are referred to as strings or words over X . Each non-empty word w is of the form $w = a_1 a_2 \cdots a_k$ for some $a_1, a_2, \dots, a_k \in X$. The element a_i is called the i th letter of w , for $i = 1, \dots, k$ and k is called the length of w . We denote by ϵ the empty sequence (word). The length of word w is denoted by $|w|$; note that $|\epsilon| = 0$. We denote by X^+ the set of non-empty words, i.e. $X^+ = X^* \setminus \{\epsilon\}$. We denote by wv the concatenation of word $w \in X^*$ with $v \in X^*$. For each $j = 1, \dots, m$, e_j is the j th unit vector of $\mathbb{R}^m$, i.e. $e_j = (\delta_{1,j}, \dots, \delta_{n,j})$, $\delta_{i,j}$ is the Kronecker symbol.

2. Linear switched systems

In this section we present the formal definition of DTLSSs along with a number of relevant system-theoretic concepts for DTLSSs.

Definition 1. Recall from [32] that a discrete-time linear switched system (abbreviated by DTLSS), is a discrete-time control system of the form

$$\Sigma \begin{cases} x_{t+1} &= A_{q_t} x_t + B_{q_t} u_t \\ y_t &= C_{q_t} x_t. \end{cases} \quad (1)$$

Here $Q = \{1, \dots, D\}$ is the finite set of discrete modes, D is a positive integer, $q_t \in Q$ is the switching signal, $u_t \in \mathbb{R}$ is the continuous input, $y_t \in \mathbb{R}^p$ is the output and $A_q \in \mathbb{R}^{n \times n}$, $B_q \in \mathbb{R}^{n \times m}$, $C_q \in \mathbb{R}^{p \times n}$ are the matrices of the linear system in mode $q \in Q$.

Notation 2. We will use $\Sigma = (n, \{A_q, B_q, C_q\}_{q \in Q})$ as a short-hand notation for a DTLSS Σ of the form (1).

Throughout the section, Σ denotes a DTLSS of the form (1). The inputs of Σ are the continuous inputs $\{u_t\}_{t=0}^\infty$ and the switching signal $\{q_t\}_{t=0}^\infty$. The state of the system at time t is x_t . Note that any switching signal is admissible and that the initial state is assumed to be zero. We use the following notation for the inputs of Σ .

Notation 3 (Hybrid inputs). Denote $\mathcal{U} = Q \times \mathbb{R}^m$.

We denote by $\mathcal{U}^*$ (resp. $\mathcal{U}^+$) the set of all finite (resp. finite and non-empty) sequences of elements of $\mathcal{U}$.

Notation 4 (Concatenation $\diamond$ of hybrid inputs). In order to avoid confusion with the notation for concatenation of finite sequences of discrete modes, we will use the symbol $\diamond$ to denote the concatenation of two sequences from $\mathcal{U}^*$. The concatenation operation on $\mathcal{U}^*$ is the same as the standard one used for finite sequences of the elements of an arbitrary set, see Notation 1.

In particular, any element w of $\mathcal{U}^+$ can be represented as the concatenation of some elements $(q_0, u_0), \dots, (q_t, u_t)$ of $\mathcal{U}$ for some $t \geq 0$, i.e.,

$$w = (q_0, u_0) \diamond (q_1, u_1) \diamond \dots \diamond (q_t, u_t) \quad (2)$$

The input w from (2) describes the scenario, when the discrete mode q_i and the continuous input u_i are fed to Σ at time i , for $i = 0, \dots, t$.

Next, we define the state and output of (1), if the latter is started from some initial state.

Definition 2 (State and output). Consider a state $x_{init} \in \mathbb{R}^n$. For any $w \in \mathcal{U}^+$ of the form (2), denote by $x_\Sigma(x_{init}, w)$ the state of Σ at time $t + 1$, and denote by $y_\Sigma(x_{init}, w)$ the output of Σ at time t , if Σ is started from x_{init} and the inputs $\{u_i\}_{i=0}^t$ and the discrete modes $\{q_i\}_{i=0}^t$ are fed to the system.

That is, $x_\Sigma(x_{init}, w)$ is defined recursively as follows; $x_\Sigma(x_{init}, \epsilon) = x_{init}$, and if $w = v \diamond (q, u)$ for some $(q, u) \in \mathcal{U}$, $v \in \mathcal{U}^*$, then

$$x_\Sigma(x_{init}, w) = A_q x_\Sigma(x_{init}, v) + B_q u.$$

If $w \in \mathcal{U}^+$ and $w = v \diamond (q, u)$, $(q, u) \in \mathcal{U}$, $v \in \mathcal{U}^*$, then

$$y_\Sigma(x_{init}, w) = C_q x_\Sigma(x_{init}, v).$$

We will call Σ *observable*, if for any two distinct initial states $x_1, x_2 \in \mathbb{R}^n$, $x_1 \neq x_2$, there exists an input $w \in \mathcal{U}^+$ (w being possibly dependent on x_1 and x_2), such that $y_\Sigma(x_1, w) \neq y_\Sigma(x_2, w)$. We say that Σ is *reachable from the initial state x_{init}* , if the reachable set $\{x_\Sigma(x_{init}, w) \mid w \in \mathcal{U}^*\}$ equals $\mathbb{R}^n$. We say that Σ is *span-reachable from the initial state x_{init}* , if the linear span of the reachable set $\{x_\Sigma(x_{init}, w) \mid w \in \mathcal{U}^*\}$ equals $\mathbb{R}^n$. Note that reachability from an initial state implies span-reachability. We say that Σ is *controllable*, if for any initial state x_{init} there exists an input $w = w(x_{init}) \in \mathcal{U}^+$, such that $x_\Sigma(x_{init}, w) = 0$, i.e., if any state of Σ can be driven to zero. From [1] it follows that if the matrix A_q is invertable for all $q \in Q$, then Σ is controllable, if and only if it is reachable from the initial state zero. Moreover, for such systems a state is either reachable from zero, or cannot be influenced by continuous inputs.

So far we have defined states and outputs generated from a certain initial state. In order to define the input-output behavior of a DTLSS, we have to either fix an initial state, or consider the input-output behavior induced by all initial states. The latter approach is more elegant, but it requires elements of the behavioral approach and the corresponding realization theory. Moreover,

this approach is less common for system identification, as typical system identification experiments involve output responses generated from one single initial state. For this reason, we will fix an initial state when defining the input-output behavior of a DTLSS.

In order to simplify the discussion, in the sequel we will consider DTLSSs with *zero initial state*. This assumption allows us to avoid technical and notational complications. In addition to simplicity, there are other arguments in favor of assuming zero initial state. First, from a practical point of view, this assumption can be made for stable DTLSSs without loss of generality, as for such systems the effect of the non-zero initial state on the measured output response will not be detectable in the long run. In turn, identification of unstable state-space representations is known to be problematic in practice, even for the linear case. Another argument in favor of zero initial state is that for DTLSSs for which the matrices A_q , $q \in Q$ are invertible controllability implies reachability from the zero initial state. More precisely, state components which are not reachable from the zero initial state cannot be influenced by continuous inputs. In turn, for controllable DTLSS, the input-output behavior from any non-zero initial state can be viewed as the shifted behavior of the input-output behavior induced by the zero initial state.

Most of the results of this paper hold only for DTLSSs with zero initial state. We will discuss possible extensions to the case of non-zero initial state, when applicable.

Definition 3 (Input-output map). *The map $f_\Sigma : \mathcal{U}^+ \rightarrow \mathbb{R}^p$, $\forall w \in \mathcal{U}^+ : f_\Sigma(w) = y_\Sigma(0, w)$, is called the input-output map of Σ .*

That is, the input-output map of Σ maps each sequence $w \in \mathcal{U}^+$ to the output generated by Σ under the hybrid input w , if started from the zero initial state. The definition above implies that the input-output behavior of a DTLSS can be formalized as a map

$$f : \mathcal{U}^+ \rightarrow \mathbb{R}^p. \quad (3)$$

The value $f(w)$ for w of the form (2) represents the output of the underlying black-box system at time t , if the continuous inputs $\{u_i\}_{i=0}^t$ and the switching sequence $\{q_i\}_{i=0}^t$ are fed to the system.

Next, we define when a general map f of the form (3) is adequately described by the DTLSS Σ , i.e. when Σ is a realization of f .

Definition 4 (Realization). *The DTLSS Σ is a realization of an input-output map f of the form (3), if f equals the input-output map of Σ , i.e. $f = f_\Sigma$.*

In order to be coherent with our choice of zero initial state, in the sequel we will use the following terminology: we will say that a DTLSS Σ is *reachable* respectively *span-reachable*, if it is reachable respectively span-reachable from the zero initial state.

Finally, in the sequel we will need the notions of dimension, minimality and isomorphism, which will be recalled below, for a more complete discussion on these concepts, see [2].

Definition 5 (Dimension). *The dimension of Σ , denoted by $\dim \Sigma$, is the dimension n of its state-space.*

Definition 6 (Minimality). *Let f be an input-output map. Then Σ is a minimal realization of f , if Σ is a realization of f , and for any DTLSS $\hat{\Sigma}$ which is a realization of f , $\dim \Sigma \leq \dim \hat{\Sigma}$.*

Definition 7 (DTLSS isomorphism). *Consider a DTLSS Σ_1 of the form (1) and a DTLSS $\hat{\Sigma} = (n, \{\hat{A}_q, \hat{B}_q, \hat{C}_q\}_{q \in Q})$. An invertible $n \times n$ matrix S is said to be an isomorphism from Σ to $\hat{\Sigma}$, if*

$$\forall q \in Q : S A_q S^{-1} = \hat{A}_q, \quad \hat{B}_q = S B_q, \quad C_q S^{-1} = \hat{C}_q.$$

If there exists an isomorphism from Σ to $\hat{\Sigma}$, then Σ and $\hat{\Sigma}$ are said to be isomorphic.

3. Overview of realization theory

Below we present an overview of the results on realization theory of DTLSSs along with the concept of Markov-parameters. For more details on the topic see [2]. In the sequel, Σ denotes a DTLSS of the form (1), and f denotes an input-output map $f : \mathcal{U}^+ \rightarrow \mathbb{R}^p$.

For our purposes the most important result is the one which states that a DTLSS realization of f can be computed from the Markov-parameters of f . In order to present this result, we need to define the Markov-parameters of f formally. Denote $Q^{k,*} = \{w \in Q^* \mid |w| \geq k\}$. Define the maps $S_j^f : Q^{2,*} \rightarrow \mathbb{R}^p$, $j = 1, \dots, m$ as follows; for any $v = \sigma_1 \dots \sigma_{|v|} \in Q^*$ with $\sigma_k \in Q$, and for any $q, q_0 \in Q$,

$$S_j^f(q_0 v q) = \begin{cases} f((q_0, e_j) \diamond (q, 0)) & \text{if } v = \epsilon \\ f((q_0, e_j) \diamond (\sigma_1, 0) \diamond \dots \diamond (\sigma_{|v|}, 0) \diamond (q, 0)) & \text{if } |v| \geq 1 \end{cases} \quad (4)$$

with $e_j \in \mathbb{R}^m$ is the vector with 1 as its j th entry and zero everywhere else. Define

$$S^f(v) = [S_1^f(v) \quad \dots \quad S_m^f(v)] \in \mathbb{R}^{p \times m}$$

for all $v \in Q^*$. The collection of values $\{S^f(v)\}_{v \in Q^*}$ is called the *Markov-parameters* of f . The functions S_j^f , $j = 1, \dots, m$ can be viewed as *input responses*. The interpretation of S_j^f will become more clear after we define the concept of a *generalized convolution representation*. Note that the values of the Markov-parameters can be obtained from the values of f .

Definition 8 (Convolution representation). *The input-output map f has a generalized convolution representation (abbreviated as **GCR**), if for all $w \in \mathcal{U}^+$ of the form (2), $f(w)$ can be expressed via the Markov-parameters of f as follows:*

$$f(w) = \sum_{k=0}^{t-1} S^f(q_k q_{k+1} \dots q_t) u_k$$

Remark 1. *If f has a **GCR**, then the Markov-parameters of f determine f uniquely.*

The motivation for introducing **GCRs** is that existence of a **GCR** is a necessary condition for realizability by DTLSSs. Moreover, if f is realizable by a DTLSS, then the Markov-parameters of f can be expressed as products of the matrices of its DTLSS realization. In order to formulate this result more precisely, we need the following notation.

Notation 5. *Consider the collection of $n \times n$ matrices A_σ , $\sigma \in X$. For any $w \in Q^*$, the $n \times n$ matrix A_w is defined as follows. If $w = \epsilon$, then A_ϵ is the identity matrix. If $w = \sigma_1 \sigma_2 \dots \sigma_k \in X^*$, $\sigma_1, \dots, \sigma_k \in X$, $k > 0$, then*

$$A_w = A_{\sigma_k} A_{\sigma_{k-1}} \dots A_{\sigma_1}. \quad (5)$$

Lemma 1. *The map f is realized by the DTLSS Σ if and only if f has a **GCR** and for all $v \in Q^*$, $q, q_0 \in Q$,*

$$S_j^f(q_0 v q) = C_q A_v B_{q_0} e_j, \quad j = 1, \dots, m. \quad (6)$$

Next, we define the concept of a Hankel-matrix. Similarly to the linear case, the entries of the Hankel-matrix are formed by the Markov parameters. For the definition of the Hankel-matrix of f , we will use lexicographical ordering on the set of sequences Q^* .

Remark 2 (Lexicographic ordering). *Recall that $Q = \{1, \dots, D\}$. We define a lexicographic ordering $\prec$ on Q^* as follows. For any $v, s \in Q^*$, $v \prec s$ if either $|v| < |s|$ or $0 < |v| = |s|$, $v \neq s$ and for some $l \in \{1, \dots, |s|\}$, $v_l < s_l$ with the usual ordering of integers and $v_i = s_i$ for $i = 1, \dots, l-1$. Here v_i and s_i denote the i th letter of v and s respectively. Note that $\prec$ is a complete ordering and $Q^* = \{v_1, v_2, \dots\}$ with $v_1 \prec v_2 \prec \dots$. Note that $v_1 = \epsilon$ and for all $i \in \mathbb{N}$, $q \in Q$, $v_i \prec v_i q$.*

In order to simplify the definition of a Hankel-matrix, we introduce the notion of a combined Markov-parameter.

Definition 9 (Combined Markov-parameters). *A combined Markov-parameter $M^f(v)$ of f indexed*

by the word $v \in Q^*$ is the following $pD \times Dm$ matrix

$$M^f(v) = \begin{bmatrix} S^f(1v1), & \dots, & S^f(Dv1) \\ S^f(1v2), & \dots, & S^f(Dv2) \\ \vdots & \dots & \vdots \\ S^f(1vD), & \dots, & S^f(DvD) \end{bmatrix} \quad (7)$$

Definition 10 (Hankel-matrix). Consider the lexicographic ordering $\prec$ of Q^* from Remark 2. Define the Hankel-matrix H_f of f as the following infinite matrix

$$H_f = \begin{bmatrix} M^f(v_1v_1) & M^f(v_2v_1) & \dots & M^f(v_kv_1) & \dots \\ M^f(v_1v_2) & M^f(v_2v_2) & \dots & M^f(v_kv_2) & \dots \\ M^f(v_1v_3) & M^f(v_2v_3) & \dots & M^f(v_kv_3) & \dots \\ \vdots & \vdots & \dots & \vdots & \dots \end{bmatrix},$$

i.e. the $pD \times (mD)$ block of H_f in the block row i and block column j equals the combined Markov-parameter $M^f(v_jv_i)$ of f . The rank of H_f , denoted by $\text{rank}H_f$, is the dimension of the linear span of its columns.

The main result on realization theory of DTLSSs can be stated as follows.

Theorem 1 ([2]). 1. The map f has a realization by a DTLSS if and only if f has a **GCR** and $\text{rank}H_f < +\infty$.
 2. A minimal DTLSS realization of f can be constructed from H_f and any minimal DTLSS realization of f has dimension $\text{rank}H_f$.
 3. A DTLSS Σ is a minimal realization of f if and only if Σ is span-reachable, observable and it is a realization of f . Any two DTLSSs which are minimal realizations of f are isomorphic.

Note that Theorem 1 shows that the knowledge of the Markov-parameters is necessary and sufficient for finding a state-space representation of f . In fact, similarly to the continuous-time case [33], we can even show that the knowledge of *finitely many* Markov-parameters is sufficient. This will be done by formulating a *realization algorithm* for DTLSSs, which computes a DTLSSs realization of f based on finitely many Markov-parameters of f .

In order to present the realization algorithm, we need the following notation.

Notation 6. Consider the lexicographic ordering $\prec$ of Q^* and recall that $Q^* = \{v_1, v_2, \dots\}$ where $v_1 \prec v_2 \dots$. Denote by $\mathbf{N}(L)$ the number of sequences from Q^* of length at most L . It then follows that $|v_i| \leq L$ if and only if $i \leq \mathbf{N}(L)$.

Definition 11 ($H_{f,L,M}$ sub-matrices of H_f). For $L, K \in \mathbb{N}$ define the integers $I_L = \mathbf{N}(L)pD$ and $J_K = \mathbf{N}(K)mD$. Denote by $H_{f,L,K}$ the following upper-left $I_L \times J_K$ sub-matrix of H_f ,

$$\begin{bmatrix} M^f(v_1v_1) & M^f(v_2v_1) & \dots & M^f(v_{\mathbf{N}(K)}v_1) \\ M^f(v_1v_2) & M^f(v_2v_2) & \dots & M^f(v_{\mathbf{N}(K)}v_2) \\ \vdots & \vdots & \dots & \vdots \\ M^f(v_1v_{\mathbf{N}(L)}) & M^f(v_2v_{\mathbf{N}(L)}) & \dots & M^f(v_{\mathbf{N}(K)}v_{\mathbf{N}(L)}) \end{bmatrix}.$$

Notice that the entries of $H_{f,L,K}$ are Markov-parameters indexed by words of length at most $L + K$, i.e. $H_{f,L,K}$ is uniquely determined by $\{M^f(v_i)\}_{i=1}^{\mathbf{N}(L+K)}$.

The promised realization algorithm is Algorithm 1, which takes as input the matrix $H_{f,N,N+1}$ and produces a DTLSS. Note that the knowledge of $H_{f,N,N+1}$ is equivalent to the knowledge of the finite sequence $\{M^f(v_i)\}_{i=1}^{\mathbf{N}(2N+1)}$ of Markov-parameters. The correctness of Algorithm 1 is stated below.

Theorem 2. If $\text{rank}H_{f,N,N} = \text{rank}H_f$, then Algorithm 1 returns a minimal realization Σ_N of f . The condition $\text{rank}H_{f,N,N} = \text{rank}H_f$ holds for a given N , if there exists a DTLSS realization Σ of f such that $\dim \Sigma \leq N + 1$.

The proof of Theorem 2 is completely analogous to its continuous-time counterpart [33]. *Theorem 2 implies that if f is realizable by a DTLSS, then a minimal DTLSS realization of f is computable from finitely many Markov-parameters, using Algorithm 1.* In fact, if f is realizable by a DTLSS of dimension n , then the first $\mathbf{N}(2n - 1)$ Markov-parameters $\{M^f(v_i)\}_{i=1}^{\mathbf{N}(2n-1)}$ uniquely determine f .

Algorithm 1

Inputs: Hankel-matrix $H_{f,N,N+1}$.

Output: DTLSS Σ_N

- 1: Let $n = \text{rank} H_{f,N,N+1}$. Choose a tuple of integers $(i_1, \dots, i_n)$ such that the columns of $H_{f,N,N+1}$ indexed by $i_1, \dots, i_n$ form a basis of $\text{Im} H_{f,N,N+1}$. Let $\mathbf{O}$ be $I_N \times n$ matrix formed by these linearly independent columns, i.e. the r th column of $\mathbf{O}$ equals the i_r th column of $H_{f,N,N+1}$. Let $\mathbf{R} \in \mathbb{R}^{n \times J_{N+1}}$ be the matrix, r th column of which is formed by coordinates of the r th column of $H_{f,N,N+1}$ with respect to the basis consisting of the columns $i_1, \dots, i_n$ of $H_{f,N,N+1}$, for every $r = 1, \dots, J_{N+1}$. It then follows that $H_{f,N,N+1} = \mathbf{O}\mathbf{R}$ and $\text{rank} \mathbf{R} = \text{rank} \mathbf{O} = n$.
- 2: Define $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times J_N}$ as the matrix formed by the first J_N columns of $\mathbf{R}$.
- 3: For each $q \in Q$, let $\mathbf{R}_q \in \mathbb{R}^{n \times J_N}$ be such that for each $i = 1, \dots, J_N$, the i th column of $\mathbf{R}_q$ equals the $r(i)$ th column of $\mathbf{R}$. Here $r(i) \in \{1, \dots, J_{N+1}\}$ is defined as follows. Consider the decomposition $i = (r - 1)mD + z$ for some $z = 1, \dots, mD$ and $r = 1, \dots, \mathbf{N}(N)$. Consider the word $v_r q$ and notice that $|v_r q| \leq N + 1$. Hence, $v_r q = v_d$ for some $d = 1, \dots, \mathbf{N}(N + 1)$. Then define $r(i)$ as $r(i) = (d - 1)mD + z$.
- 4: Construct Σ_N of the form (1) such that

$$\begin{aligned} [B_1, \dots, B_D] = \\ \text{the first } mD \text{ columns of } \mathbf{R} \end{aligned} \quad (8)$$

$$[C_1^T \ C_2^T \ \dots \ C_D^T]^T = \text{the first } pD \text{ rows of } \mathbf{O} \quad (9)$$

$$\forall q \in Q : A_q = \mathbf{R}_q \tilde{\mathbf{R}}^+ \quad (10)$$

where $\tilde{\mathbf{R}}^+$ is the Moore-Penrose pseudoinverse of $\tilde{\mathbf{R}}$.

- 5: Return Σ_N
-

The intuition behind Algorithm 1 is the following. The state-space of the DTLSS Σ_N returned by Algorithm 1 is an isomorphic copy of the space spanned by the columns of $H_{f,N,N}$. The isomorphism is determined by the matrix $\mathbf{R}$. The columns of B_q , $q \in Q$ are formed by the columns $(q - 1)mD + 1, \dots, qmD$ of the block-matrix

$$[M^f(v_1 v_1)^T \ \dots \ M^f(v_1 v_{\mathbf{N}(L)})^T]^T.$$

The rows of C_q , $q \in Q$ are formed by the rows $(q - 1)p + 1, \dots, pq$ of $H_{f,N,N+1}$. Finally, the matrix A_q , $q \in Q$ is the matrix of a shift-like operator, which maps a block-column $\{M^f(v_j v_i)\}_{i=1}^{\mathbf{N}(L)}$ of $H_{f,N,N}$ to the block-column $\{M^f(v_j q v_i)\}_{i=1}^{\mathbf{N}(L)}$ of $H_{f,N,N+1}$.

The intuition behind Algorithm 1 explains why Theorem 2 requires $\text{rank} H_{f,N,N} = \text{rank} H_f$, even though Algorithm 1 uses the matrix $H_{f,N,N+1}$. In a nutshell, $\text{rank} H_{f,N,N} = \text{rank} H_f$ implies that $\text{rank} H_{f,N,N+1} = \text{rank} H_{f,N,N} = \text{rank} H_f$. Since A_q is the matrix representation of the linear map which maps columns of $H_{f,N,N}$ to columns of $H_{f,N,N+1}$, the condition $\text{rank} H_{f,N,N+1} = \text{rank} H_{f,N,N}$ ensures that A_q is a well-defined square matrix. Moreover, the condition $\text{rank} H_{f,N,N+1} = \text{rank} H_{f,N,N} = \text{rank} H_f$ also ensures that A_q is the matrix representation of the linear map which maps a the block-column $\{M^f(v_j v_i)\}_{i=1}^\infty$ of the infinite Hankel-matrix H_f to the shifted block-column $\{M^f(v_j q v_i)\}_{i=1}^\infty$ of the same matrix. In turn, the latter is necessary for the result of Algorithm 1 to be a realization of f . Note that requiring only $\text{rank} H_{f,N,N+1} = \text{rank} H_f$ would not have guaranteed that $\text{rank} H_{f,N,N+1} = \text{rank} H_{f,N,N}$ holds, and thus the matrix representation of a linear map shifting columns of $H_{f,N,N}$ would not be a square matrix.

Note that the size of the matrix $H_{f,N,N+1}$ grows exponentially with N . This renders Algorithm

1 impractical for large N . There is a way to get around it, by using so called *selections*, introduced in [34, 35, 36].

We will call a finite subset $\alpha \subset Q^* \times Q \times \{1, \dots, p\}$ a *row selection* and a finite subset $\beta \subseteq Q^* \times Q \times \{1, \dots, m\}$ a *column selection*. For the SISO case, row and column selections can be thought of as subsets of $Q^* \times Q$. We will refer to row and column selections as *selections*, if it is clear from the context whether we mean row or column selections.

Selections will be used to define Hankel-like matrices, entries of which are Markov parameters.

Assume that α and β are row and column selections respectively and assume that α and β are both finite sets of cardinality n and κ respectively. Fix an enumeration of the elements of α and β as follows:

$$\alpha = \{(v_i, q_i, l_i)\}_{i=1}^n, \quad \beta = \{(w_j, \sigma_j, h_j)\}_{j=1}^\kappa. \quad (11)$$

Let us now define the matrix $\mathcal{H}_{f,\alpha,\beta} \in \mathbb{R}^{n \times \kappa}$ as follows:

$$[\mathcal{H}_{f,\alpha,\beta}]_{i,j} = (S^f(\sigma_j w_j v_i q_i))_{l_i, h_j} \quad i = 1, \dots, n, j = 1, \dots, \kappa. \quad (12)$$

If α and β have the same cardinality, then the matrix $H_{f,\alpha,\beta}$ is a square one. Intuitively, the rows of $\mathcal{H}_{f,\alpha,\beta}$ are indexed by the elements of α , and the columns by the elements of β .

In order to present the algorithm, we define the matrices $\mathcal{H}_{q,f,\alpha,\beta} \in \mathbb{R}^{n \times \kappa}$, $\mathcal{H}_{f,\alpha,q} \in \mathbb{R}^{n \times m}$ and $\mathcal{H}_{q,f,\beta} \in \mathbb{R}^{p \times \kappa}$:

$$[\mathcal{H}_{q,f,\alpha,\beta}]_{i,j} = (S^f(\sigma_j w_j q v_i q_i))_{l_i, h_j}, \quad i = 1, \dots, n, j = 1, \dots, \kappa, \quad (13)$$

$$[\mathcal{H}_{f,\alpha,q}]_{i,r} = (S^f(q v_i q_i))_{l_i, r}, \quad i = 1, \dots, n, \quad r = 1, \dots, m, \quad (14)$$

$$[\mathcal{H}_{q,f,\beta}]_{r,j} = (S^f(\sigma_j w_j q))_{r, h_j}, \quad j = 1, \dots, \kappa, r = 1, \dots, p \quad (15)$$

Algorithm 2 Realization algorithm for DTLSSs with selections

Inputs: selections α, β and matrices $\mathcal{H}_{f,\alpha,\beta}, \mathcal{H}_{q,f,\alpha,\beta}, \mathcal{H}_{f,\alpha,q}$ and $\mathcal{H}_{q,f,\beta}$, $q \in Q$.

- 1: Choose a tuple of integers $(i_1, \dots, i_{n_m})$ such that the columns of $H_{f,\alpha,\beta}$ which are indexed by $i_1, \dots, i_{n_m}$ form a basis of $\text{Im} H_{f,\alpha,\beta}$. Let O_{n_m} be $n \times n_m$ matrix formed by these linearly independent columns, i.e. the r th column of O_{n_m} equals the i_r th column of $H_{f,\alpha,\beta}$. Let $R_{n_m} = O_{n_m}^+ H_{f,\alpha,\beta}$, where $O_{n_m}^+$ is the Moore-Penrose inverse of O_{n_m} . It then follows that $H_{f,\alpha,\beta} = O_{n_m} R_{n_m}$ and O_{n_m} is full column rank, R_{n_m} is full **row rank**.
- 2: Define

$$\hat{A}_q = O_{n_m}^+ \mathcal{H}_{q,f,\alpha,\beta} R_{n_m}^+, \quad \hat{B}_q = O_{n_m}^+ \mathcal{H}_{f,\alpha,q}, \quad \hat{C}_q = \mathcal{H}_{q,f,\beta} R_{n_m}^+$$

where $O_{n_m}^+, R_{n_m}^+$ are the Moore-Penrose inverse of O_{n_m} and R_{n_m} respectively.

- 3: **return** $\hat{\Sigma} = (n_m, \{\hat{A}_q, \hat{B}_q, \hat{C}_q\}_{q \in Q})$.
-

Theorem 3 (Adapted from [35], realization algorithm with selections). *If n_m is the dimension of a minimal DTLSS realization of f , then the DTLSS $\hat{\Sigma}$ defined in Algorithm 2 is a minimal realization of f . Moreover, if n_m is the dimension of a minimal DTLSS realization of f , then there exists a pair of selections $\alpha \subseteq Q^* \times Q \times \{1, \dots, p\}$ and $\beta \subseteq Q^* \times Q \times \{1, \dots, m\}$, such that the cardinality of the sets α, β is n_m and $\text{rank} \mathcal{H}_{f,\alpha,\beta} = n_m = \text{rank} H_f$.*

In the sequel we will use the following notation.

Notation 7 ($Q^{\leq k}$). We denote by $Q^{\leq k}$ the set of all sequences from Q^* of length at most k .

Remark 3 (Relationship with Algorithm 1). *From Theorem 2 and [35] it follows that we can choose $\alpha = Q^{\leq N} \times Q \times \{1, \dots, p\}$ and $\beta = Q^{\leq N} \times Q \times \{1, \dots, m\}$, where N is any integer not smaller than the dimension of a minimal DTLSS realization of f . In this case, Algorithm 2 is equivalent to Algorithm 1, see [35, 37].*

Remark 4 (Markov parameters indexed by selections determine all Markov-parameters). *Theorem 2 and Theorem 3 imply that for any input-output map f which is realizable by a DTLSS, there exist a*

finite set $\Gamma \subseteq Q^*$ such that the knowledge of the finite number of Markov-parameters $\{S^f(v)\}_{v \in \Gamma}$ is sufficient to determine the entire input-output map f . This observation will be useful for computing persistently exciting inputs.

Indeed, from Theorem 2 it follows that if f is realizable by a DTLSS of dimension n , then we can choose Γ as

$$\Gamma := \Gamma_n = Q^{\leq 2n+3}, \quad (16)$$

i.e., Γ is the set of all words of length at most $2n + 3$. It then follows that $\{S^f(v)\}_{v \in \Gamma}$ form the entries of the Hankel-matrix $H_{f,n,n+1}$ and the latter can be used to compute a DTLSS realization of f using Algorithm 1.

The set Γ from (16) has $O(|Q|^{2n})$ elements, and hence its size is exponential in n . As we shall see later, the size of Γ determines the length of persistently exciting inputs. Therefore, this definition of Γ is not always practical for computing persistently exciting inputs.

This motivates us to formulate a refined definition of Γ using selections and Theorem 3. Consider selections α, β such that $\text{rank} H_{f,\alpha,\beta} = \text{rank} H_f$. By Theorem 3 such selections always exist. Let us define Γ as a function of α, β as follows

$$\begin{aligned} \Gamma = \Gamma(\alpha, \beta) = & \{rvq \mid r \in Q \text{ and } \exists q \in Q, \exists l = 1, \dots, p : (v, q, l) \in \alpha\} \cup \\ & \{q_0wr \mid r \in Q \text{ and } \exists q_0 \in Q, \exists h \in \{1, \dots, m\} : (w, q_0, h) \in \alpha\} \cup \\ & \{q_0wrvq \mid r \in Q \text{ and } \exists q_0, q \in Q, \exists l \in \{1, \dots, p\}, \exists h \in \{1, \dots, m\} : (v, q, l) \in \alpha, (w, q_0, h) \in \beta\} \cup \\ & \{q_0wvq \mid \exists q_0, q \in Q, \exists l \in \{1, \dots, p\}, \exists h \in \{1, \dots, m\} : (v, q, l) \in \alpha, (w, q_0, h) \in \beta\} \end{aligned} \quad (17)$$

Then the knowledge of $\{S^f(v)\}_{v \in \Gamma(\alpha,\beta)}$ is sufficient to construct the matrices $\mathcal{H}_{f,\alpha,\beta}, \mathcal{H}_{q,f,\alpha,\beta}, \mathcal{H}_{f,\alpha,q}$ and $\mathcal{H}_{q,f,\beta}$, $q \in Q$, and the latter matrices can be used by Algorithm 2 to compute a DTLSS realization of f . In turn, the latter realization is sufficient to determine all the Markov-parameters $\{S^f(v)\}_{v \in Q^*}$.

Note that α and β can be chosen to have n elements, where n is the dimension of a minimal DTLSS realization of f . That is, at most $(|Q| + 1)n^2 + 2n|Q|$ Markov parameters are sufficient to determine **all** the Markov-parameters of f .

In general, the choice of α and β depends on the input-output map f , especially if we would like these selections to contain as many elements as the dimension of a minimal DTLSS realization of f . However, if we do not put restrictions on the number of elements of α, β , then it is possible to choose α and β independently of f , namely, by choosing the selections $\alpha = Q^{\leq N} \times Q \times \{1, \dots, p\}$, $\beta = Q^{\leq N} \times Q \times \{1, \dots, m\}$ such that $N \geq n$, where n is the dimension of a minimal DTLSS realization of f . In this case, $\Gamma(\alpha, \beta)$ will be equal to Γ_N from (16), and the number of elements of α and β is exponential in N , i.e., it is $O(|Q|^N)$. However, the set of Markov-parameters which is required in order to determine all the Markov-parameters of f does not depend on f .

That is, we can determine all the Markov-parameters of f from a finite set of Markov parameters, and this set either

- depends on f but its cardinality is quadratic in n , or
- is independent of f , but then its cardinality is exponential in n ,

where n is the dimension of a minimal DTLSS realization of f .

4. Main results of the paper

The main idea behind our definition of persistence of excitation is as follows. The measured time series is persistently exciting, if from this time-series we can reconstruct the Markov-parameters of the underlying system. Note that by Theorem 2, it is enough to reconstruct *finitely many* Markov-parameters. This also means that our definition of persistence of excitation is also applicable to finite time series.

In order to present our main results, we will need some terminology.

Definition 12 (Output time-series). For any input-output map f and for any finite input sequence $w \in \mathcal{U}^+$ we denote by $\mathbf{O}(f, w)$ the output time series induced by f and w , i.e. if w is of the form (2), then $\mathbf{O}(f, w) = \{y_t\}_{t=0}^T$, such that $y_t = f((q_0, u_0) \diamond \dots \diamond (q_t, u_t))$ for all $t \leq T$.

Definition 13 (Persistence of excitation). The finite sequence $w \in \mathcal{U}^+$ is persistently exciting for the input-output map f , if it is possible to determine the Markov-parameters of f from the data $(w, \mathbf{O}(f, w))$.

Remark 5 (Interpretation). *Remark 4 allows the following interpretation of persistence of excitation defined above. If w is persistently exciting, then the Markov-parameters of f can be computed from the response of f to the prefixes of w . In particular, if f admits a DTLSS realization of dimension at most n , then there exists a finite subset $\Gamma \subseteq Q^*$ of the Markov-parameters such that $\{S^f(v)\}_{v \in \Gamma}$ determine all the other Markov-parameters, and $\{S^f(v)\}_{v \in \Gamma}$ can be computed from the data $(w, \mathbf{O}(f, w))$. The knowledge of $\{S^f(v)\}_{v \in \Gamma}$ is sufficient for computing a DTLSS realization of f . Hence, persistence of excitation of w for f means that Algorithm 1 or Algorithm 2 can serve as an identification algorithm for computing a DTLSS realization of f from the time-series $(w, \mathbf{O}(f, w))$. Note, however, that our definition does not depend on Algorithm 1 or Algorithm 2. Indeed, if there is any algorithm which can correctly find a DTLSS realization of f from $(w, \mathbf{O}(f, w))$, then according to our definition, w is persistently exciting, as the knowledge of a DTLSS realization allows us to compute all the Markov-parameters. Note that our definition of persistence of excitation involves only the inputs, but not the output response.*

So far we have defined the persistence of excitation for finite sequences of inputs. Next, we define the same notion for infinite sequences of inputs. To this end, we need the following notation.

Notation 8. We denote by $\mathcal{U}^\omega$ the set of infinite sequences of hybrid inputs. That is, any element $w \in \mathcal{U}^\omega$ can be interpreted as a time-series $w = \{(q_t, u_t)\}_{t=0}^\infty$. For each $N \in \mathbb{N}$, denote by w_N the sequence formed by the first N elements of w , i.e. $w_N = (q_0, u_0) \diamond \dots \diamond (q_N, u_N)$.

Definition 14 (Asymptotic persistence of excitation). An infinite sequence of inputs $w \in \mathcal{U}^\omega$ is called asymptotically persistently exciting for the input-output map f , if the following holds. For every sufficiently large N , we can compute from $(w_N, \mathbf{O}(f, w_N))$ asymptotic estimates of the Markov-parameters of f . More precisely, for $N \in \mathbb{N}$, we can compute from $(w_N, \mathbf{O}(f, w_N))$ some matrices $\{M_N^f(v)\}_{v \in Q^*}$ such that $\lim_{N \rightarrow \infty} M_N^f(v) = M^f(v)$ for all $v \in Q^*$. When clear from the context, we will use the term persistently exciting instead of asymptotically persistently exciting.

Remark 6 (Interpretation of asymptotic persistence of excitation). The interpretation of asymptotic persistence of excitation is that asymptotically persistently exciting inputs allow us to estimate a DTLSS realization of f with arbitrary accuracy. Indeed, assume that $w \in \mathcal{U}^\omega$ is asymptotically persistently exciting. Then for each N we can compute from the time-series $(w_N, \mathbf{O}(f, w_N))$ an approximation $\{M_N^f(v)\}_{v \in Q^*}$ of the Markov-parameters of f . Suppose that f is realizable by a DTLSS of dimension n and we know the indices $(i_1, \dots, i_n)$ of those columns of $H_{f, n-1, n}$ which form a basis of the column space of $H_{f, n-1, n}$. Let $H_{f, n-1, n}^N$ be the matrix which is constructed in the same way as $H_{f, n-1, n}$, but with $M_N^f(v)$ instead of the Markov-parameters $M^f(v)$. Since $M_N^f(v)$ converges to $M^f(v)$ for all $v \in Q^*$, we get that each entry of $H_{f, n-1, n}^N$ converges to the corresponding entry of $H_{f, n-1, n}$. Modify Algorithm 1 by fixing the choice of columns to $(i_1, \dots, i_n)$ in the first step. It is easy to see that the modified algorithm represents a continuous map from the input data (finite Hankel-matrix) to the output data (matrices of a DTLSS). For sufficiently large N , the columns of $H_{f, n-1, n}^N$ indexed by $(i_1, \dots, i_n)$ also represent a basis of the column space of $H_{f, n-1, n}^N$. If we apply the modified Algorithm 1 to the sequence of matrices $H_{f, n-1, n}^N$, we obtain a sequence of DTLSSs $\Sigma_{n, N}$ and the parameters of $\Sigma_{n, N}$ converge to the parameters of the DTLSS Σ which we would obtain from Algorithm 1 if we applied it to $H_{f, n-1, n}$. In particular, by choosing a sufficiently large N , the parameters of $\Sigma_{n, N}$ are sufficiently close to those of Σ .

A similar argument can be made by replacing Algorithm 1 by Algorithm 2, and by replacing $H_{f, n-1, n}$ with the matrices $\mathcal{H}_{f, \alpha, \beta}, \mathcal{H}_{q, f, \alpha, \alpha}, \mathcal{H}_{f, \alpha, q}, \mathcal{H}_{q, f, \beta, q} \in Q$ for suitable selections α, β . More precisely, assume that α, β are selections containing n_m elements, such that $\text{rank } H_{f, \alpha, \beta} = \text{rank } H_f = n_m$. In this case, O_{n_m} equals $H_{f, \alpha, \beta}, O_{n_m}^+$ is the inverse of $H_{f, \alpha, \beta}$, and R_{n_m} is the

identity matrix. It then follows that the matrices returned by Algorithm 2 are continuous functions of the entries of the matrices $\mathcal{H}_{f,\alpha,\beta}, \mathcal{H}_{q,f,\alpha,\alpha}, \mathcal{H}_{f,\alpha,q}, \mathcal{H}_{q,f,\beta}, q \in Q$.

Let $\mathcal{H}_{f,\alpha,\beta}^N, \mathcal{H}_{q,f,\alpha,\alpha}^N, \mathcal{H}_{f,\alpha,q}^N, \mathcal{H}_{q,f,\beta}^N, q \in Q$ be the matrices constructed by replacing $M^f(v)$ by $M_N^f(v)$ (note that the matrices $S^f(q_0 v q)$ used in (12)-(15) are blocks of $M^f(v)$, and hence the matrices defined in (12)-(15) are functions $\{M^f(v)\}_{v \in Q^*}$). Consider the DTLSS $\hat{\Sigma}_N$ returned by a version of Algorithm 2, where $\mathcal{H}_{f,\alpha,\beta}, \mathcal{H}_{q,f,\alpha,\alpha}, \mathcal{H}_{f,\alpha,q}, \mathcal{H}_{q,f,\beta}, q \in Q$ are replaced by $\mathcal{H}_{f,\alpha,\beta}^N, \mathcal{H}_{q,f,\alpha,\alpha}^N, \mathcal{H}_{f,\alpha,q}^N, \mathcal{H}_{q,f,\beta}^N, q \in Q$. Let $\hat{\Sigma}$ be the DTLSS returned by Algorithm 2 when applied to the true matrices $\mathcal{H}_{f,\alpha,\beta}, \mathcal{H}_{q,f,\alpha,\alpha}, \mathcal{H}_{f,\alpha,q}, \mathcal{H}_{q,f,\beta}, q \in Q$. Since for large enough N , the matrices $\mathcal{H}_{f,\alpha,\beta}^N, \mathcal{H}_{q,f,\alpha,\alpha}^N, \mathcal{H}_{f,\alpha,q}^N, \mathcal{H}_{q,f,\beta}^N, q \in Q$ are close to $\mathcal{H}_{f,\alpha,\beta}, \mathcal{H}_{q,f,\alpha,\alpha}, \mathcal{H}_{f,\alpha,q}, \mathcal{H}_{q,f,\beta}, q \in Q$, and the steps of Algorithm 2 are continuous in the elements of those matrices, the matrices of $\hat{\Sigma}_N$ will converge to those of $\hat{\Sigma}$ as $N \rightarrow \infty$. That is, $\hat{\Sigma}_N$ can be viewed as an approximate realization of f .

The discussion above highlights the difference between persistently exciting inputs and asymptotically persistently exciting inputs. Namely, by Remark 5 persistently exciting inputs allow us to reconstruct an *exact* model of the input-output map f from the output response of finite length. In contrast, as Remark 6 indicates, asymptotically persistently exciting inputs allow us to construct an *approximate* model of f from output responses of finite length, with the additional property that as the length of the output response converges to ∞ , the obtained approximate model converges to an exact model of f . That is, the two concepts are different.

Remark 7 (Recovering finitely many Markov-parameters). *The definition of (asymptotically) persistently exciting inputs involve the possibility of recovering all the Markov-parameters from the output response to that input. At a first glance, this may seem ill-posed, as from inputs of finite length one expects to be able to recover at best a finite number of Markov-parameters. However, Remark 5-6 clearly show that it is sufficient to recover only a finite number of Markov-parameters, as for input-output maps realizable by DTLSSs, a finite number of Markov-parameters determine all the Markov-parameters of the input-output map.*

We will show that for **every** reversible DTLSS there exists **some** input which is persistently exciting. In addition, we present a class of inputs which are persistently exciting of any input-output map f realizable by a stable DTLSS.

4.1. Persistently exciting input for specific systems

In this section we present results which state that for any input-output map f which is realizable by a **reversible** DTLSS, there exists a persistently exciting finite input.

Note that from (4) it follows that the Markov-parameters of f can be obtained from finitely many input-output data. However, the application of (4) implies evaluating the response of the system for different inputs, while started from a fixed initial state. In order to simulate this by evaluating the response of the system to one single input (which is then necessarily persistently exciting), one has to provide means to reset the system to its initial state. In order to be able to do so, we restrict attention to *reversible* DTLSSs.

Definition 15. A DTLSS Σ of the form (1) is reversible, if for every discrete mode $q \in Q$, the matrix A_q is invertible.

Reversible DTLSSs arise naturally when sampling continuous-time systems.

Theorem 4. Consider an input-output map f . Assume that f has a realization by a reversible DTLSS. Then there exists an input $w \in \mathcal{U}^+$ such that w is persistently exciting for f .

Proof. The main idea behind the proof of Theorem 4 is as follows. If f admits a DTLSS realization of dimension n , then by Remark 4, there exists a finite selections α, β and there exists a finite set $\Gamma(\alpha, \beta) \subseteq Q^*$ such that the Markov-parameters $\{S^f(v)\}_{v \in \Gamma(\alpha, \beta)}$ determine all the Markov-parameters of f uniquely. Hence, in order for a finite input w to be persistently exciting for f , it is sufficient that $\{S^f(v)\}_{v \in \Gamma(\alpha, \beta)}$ can be computed from the response $(w, \mathbf{O}(f, w))$.

Note that (4) implies that $\{S^f(v)\}_{v \in \Gamma(\alpha, \beta)}$ can be computed from the responses of f from finitely many inputs. More precisely, $\{S^f(v)\}_{v \in \Gamma(\alpha, \beta)}$ can be computed from $\{f(s) \mid s \in S(\alpha, \beta)\}$, where

$$S(\alpha, \beta) = \{(q_0, e_j) \diamond (\sigma_1, 0) \diamond \dots \diamond (\sigma_k, 0) \diamond (q, 0) \in \mathcal{U}^+ \mid q_0, q \in Q, \sigma_1, \dots, \sigma_k \in Q, q_0 \sigma_1 \dots \sigma_k q \in \Gamma(\alpha, \beta), k \geq 0, j = 1, \dots, m\}. \quad (18)$$

That is, for each $v \in \Gamma(\alpha, \beta)$, each column of $S^f(v)$ equals to $f(s)$ for some $s \in S(\alpha, \beta)$. Hence, if for each $s \in S(\alpha, \beta)$ there exists a prefix p of w such that $f(s) = f(p)$, then this w will be persistently exciting.

One way to construct such a w is to construct for each $s \in S(\alpha, \beta)$ an input $s^{-1} \in \mathcal{U}^+$ such that

$$\forall v \in \mathcal{U}^+ : f(s \diamond s^{-1} \diamond v) = f(v).$$

That is, the input s^{-1} neutralizes the effect of the input s . We defer the construction of the input s^{-1} to the end of the proof. Assume for the moment being that such inputs s^{-1} exist. Let $S(\alpha, \beta) = \{s_1, \dots, s_d\}$ be an enumeration of $S(\Gamma)$. Then it is easy to see that $f(s_1 \diamond s_1^{-1} \diamond s_2) = f(s_2)$, $f(s_1 \diamond s_1^{-1} \diamond s_2 \diamond s_2^{-1} \diamond s_3) = f(s_3)$, etc. Hence, if we define

$$w = s_1 \diamond s_1^{-1} \diamond \dots \diamond s_{d-1} \diamond s_{d-1}^{-1} \diamond s_d, \quad (19)$$

then each $f(s)$, $s \in S(\alpha, \beta)$ can be obtained as a response of f to a suitable prefix of w . Hence, w is persistently exciting.

It is left to show that s^{-1} exists. Consider a reversible realization Σ of f . Then the controllable set and reachable set of Σ coincide by [38]. Hence, from any reachable state x of Σ , there exists an input $w(x)$ such that $w(x)$ drives Σ from x to zero, i.e. $x_\Sigma(x, w(x)) = 0$. For each $s \in S$, let $x(s) = x_\Sigma(0, s)$ and define $s^{-1} = w(x(s))$ as the input which drives $x(s)$ back to the initial zero state. $\square$

It is easy to see that Theorem 4 can be extended to any input-output map which admits a controllable DTLSS realization. However, it is not clear if every input-output map which is realizable by a DTLSS is also realizable by a controllable DTLSS.

The proof of Theorem 4 can be used to formulate an algorithm for constructing a persistently exciting input. In order to present this algorithm, we will need to adapt from [1] the following procedure for constructing an input to drive a DTLSS from any state to zero.

Assume that Σ is a DTLSS of the form (1), and consider a finite column selection $\beta \subseteq Q^* \times Q \times \{1, \dots, m\}$

$$\beta = \{(w_i, q_i, j_i)\}_{i=1}^\kappa \quad (20)$$

such that the matrix

$$\mathcal{R}(\beta) = [A_{w_1} B_{q_1}, \quad A_{w_2} B_{q_2}, \quad \dots, \quad A_{w_\kappa} B_{q_\kappa}] \quad (21)$$

has rank n , i.e., $\text{rank} \mathcal{R}(\beta) = n$. In particular, from [34] it follows that if Σ is span-reachable, then there exists a set β of the form (20) such that $\text{rank} \mathcal{R}(\beta) = n$. Moreover, $\beta = Q^{\leq n-1} \times Q \times \{1, \dots, m\}$ is a valid choice for any span-reachable DTLSS. In fact, from [34] it follows that if $\text{rank} H_{f, \alpha, \beta} = \text{rank} H_f$ for some selections α, β , then $\text{rank} \mathcal{R}(\beta) = n$ for any minimal DTLSS realization Σ of f such that $\dim \Sigma = n$.

Let us define the switching sequences π_i , $i = 1, \dots, n$ recursively as follows:

$$\pi_1 = q_1 w_1, \quad \pi_{i+1} = \underbrace{\pi_i \dots \pi_i}_{(n+1)\text{-times}} q_{i+1} w_{i+1}, \quad i = 1, \dots, \kappa - 1 \quad (22)$$

From [39] it then follows that any state of Σ can be driven to 0 using the switching sequence π_κ and some continuous input. More precisely, assume that

$$\pi_\kappa = q_1 \dots q_L, \quad (23)$$

$q_1, \dots, q_L \in Q$ for $L > 0$. It then follows that for any $x \in \mathbb{R}^n$ there exists $u_1 \cdots u_L \in \mathbb{R}^m$, such that for

$$w(x) = (q_1, u_1) \diamond (q_2, u_2) \cdots (q_L, u_L) \quad (24)$$

$x_\Sigma(x, w(x)) = 0$. The input $u_1, \dots, u_L$ can be computed as follows: let us define the matrix

$$\mathcal{R}_{\pi_\kappa} = [A_{q_2 \cdots q_L} B_{q_1}, \quad A_{q_3 \cdots q_L} B_{q_2}, \quad \dots, \quad A_{q_L} B_{q_{L-1}}, \quad B_{q_L}] \quad (25)$$

From [38] it follows that $\text{rank} \mathcal{R}_{\pi_\kappa} = n$ and $u_1, \dots, u_L$ can be computed by solving the following linear least squares problem

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_L \end{bmatrix} = \underset{\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_L \in \mathbb{R}^m}{\text{argmin}} \|A_{q_1 \cdots q_L} x + \mathcal{R}_{\pi_\kappa} \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_L \end{bmatrix}\|_2^2 \quad (26)$$

Then using the proof of Theorem 4 yields the following algorithm for constructing a persistently exciting input and switching sequence. From the proof of Theorem 4 the following results follow.

Algorithm 3 **Input** a reachable DTLSS realization $\Sigma = (n, \{A_q, B_q, C_q \mid q \in Q\})$ of f , and selections α, β such that $\text{rank} H_{f, \alpha, \beta} = \text{rank} H_f$. **Output** **input** $w \in \mathcal{U}^+$.

- 1: Construct the sequence π_κ as defined in (22).
- 2: Consider $\Gamma(\alpha, \beta)$ from (17).
- 3: Fix an enumeration $\Gamma(\alpha, \beta) = \{\gamma_1, \dots, \gamma_R\}$. Then $S(\alpha, \beta)$ from (18) is of the form

$$S(\alpha, \beta) = \{s_1, \dots, s_d\}, \quad d = mR$$

where for every $i = 1, \dots, d$, if $i = m(k-1) + j$ for some $k = 1, \dots, R$, $j = 1, \dots, m$ and $\gamma_k = q_{0,k} q_{1,k} \cdots q_{L_k,k}$, $q_{0,k}, \dots, q_{L_k,k} \in Q$, $L_k \geq 1$,

$$s_i = (q_0, e_j) \diamond (q_{1,k}, 0) \diamond \cdots \diamond (q_{L_k}, 0) \quad (27)$$

- 4: For each $i = 1, \dots, d$, using (27), define

$$s_i^{-1} = w(x_i), \quad x_i = A_{q_{1,k} \cdots q_{L_k,k} q_1} B_{q_{0,k}} e_j \quad (28)$$

where $w(x_i)$ is as in (26) for $x = x_i$.

- 5: Construct w according to (19), i.e.,

$$w = s_1 \diamond s_1^{-1} \diamond \cdots \diamond s_{d-1} \diamond s_{d-1}^{-1} \diamond s_d$$

- 6: **return** w
-

Corollary 1 (Correctness of Algorithm 3). *Assume that the selections α, β are such that $\text{rank} H_{f, \alpha, \beta} = \text{rank} H_f$. The input w returned by Algorithm 3 is persistently exciting for f .*

Note that the Algorithm 3 requires the knowledge of selections α, β such that $\text{rank} H_{f, \alpha, \beta} = \text{rank} H_f$. This is not a restriction, as from Remark 3 it follows that we can always take $\alpha = Q^{\leq n} \times Q \times \{1, \dots, p\}$ and $\beta = Q^{\leq n} \times Q \times \{1, \dots, m\}$, where n is the dimension of a DTLSS realization of f .

Remark 8 (Complexity of the persistently exciting input). *If the selections α, β used in Algorithm 3 are chosen to have n elements and belong to $Q^{\leq n} \times Q \times \{1, \dots, p\}$ and $Q^{\leq n} \times Q \times \{1, \dots, m\}$ respectively, then the length of the persistently exciting input w returned by the Algorithm 3 is $O((n+1)^n \cdot (n^2+3) \cdot mR) = O((n+1)^{n+4})$, i.e., it is exponential in n . Indeed, the length of π_κ is at most $(n+1)^n$, and hence so is the length of s_i^{-1} . The length of each s_i is at most $n^2+3 \leq O((n+1)^2)$. Finally, the number R of elements of $\Gamma(\alpha, \beta)$ is at most $2(|Q| \cdot n) + (|Q|+1)n^2$, i.e., $O((n+1)^2)$. This means that the length of the necessary input grows rapidly with the state-space dimension of*

the underlying DTLSS. The memory complexity of Algorithm 3 is then also $O((n+1)^{n+4})$. The time complexity of Algorithm 3 is determined by the need for solving (26), which is polynomial in R and n , i.e., the time complexity of Algorithm 3 is polynomial in n .

Example 1. Consider the DTLSS of the form (1) such that $Q = \{1, 2\}$ and

$$\begin{aligned} A_1 &= \begin{bmatrix} 0.67032 & 0.22099 & 0.00000 \\ 0.00000 & 0.44933 & 0.00000 \\ 0.00000 & 0.00000 & 0.44933 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.44933 & 0.00000 & 0.00000 \\ 0.00000 & 0.72466 & 0.27534 \\ 0.00000 & 0.27534 & 0.72466 \end{bmatrix}, \\ B_1 &= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad C_1 = C_2 = [1 \quad 0 \quad 0] \end{aligned} \quad (29)$$

Consider $\alpha = \{(\epsilon, 1, 1), (1, 1, 1), (21, 1, 1)\}$ and $\beta = \{(\epsilon, 2, 1), (2, 1, 1), (21, 2, 1)\}$. Then by applying Algorithm 3 we get a persistently exciting input of length 864. In particular, the sequence π_n from (22) will be 222122122221222221 and its length is 18. The set Γ has 38 elements, the first couple of elements of $\Gamma(\alpha, \beta)$ are as follows:

$$\Gamma(\alpha, \beta) = \{11, 21, 211, 221, 121, 1211, 1221, 2211, 22111, \dots\}.$$

Then the first couple of inputs s_i from (27) are as follows: $s_1 = (1, 1) \diamond (1, 0)$, $s_2 = (2, 1) \diamond (1, 0)$, $\dots$, $s_9 = (2, 1) \diamond (2, 0) \cdot (1, 0) \diamond (1, 0) \diamond (1, 0)$. Then for instance s_1^{-1} from (28) is

$$\begin{aligned} s_1^{-1} &= (2, 0.00090) \diamond (2, -0.00090) \diamond (2, -0.00089) \diamond (1, -0.00399) \diamond (2, -0.00199) \diamond (2, -0.00199) \diamond \\ &(1, -0.00886) \diamond (2, -0.00443) \diamond (2, -0.00440) \diamond (2, -0.00434) \diamond (2, -0.00421) \diamond (1, -0.01924) \diamond \\ &(2, -0.00957) \diamond (2, -0.00885) \diamond (2, -0.00724) \diamond (2, -0.00364) \diamond (2, 0.00435) \diamond (1, 0.00007) \\ s_9^{-1} &= (2, -0.00009) \diamond (2, -0.00009) \diamond (2, -0.00009) \diamond (1, -0.00040) \diamond (2, -0.00020) \diamond (2, -0.00020) \diamond \\ &(1, -0.00089) \diamond (2, -0.00045) \cdot (2, -0.00044) \diamond (2, -0.00044) \diamond (2, -0.00042) \diamond (1, -0.00194) \diamond (2, -0.00097) \diamond \\ &(2, -0.00089) \diamond (2, -0.00073) \diamond (2, -0.00037) \diamond (2, 0.00044) \diamond (1, 0.00001) \end{aligned}$$

Then using (19), the persistently exciting input w is of the form

$$\begin{aligned} w &= (1, 1) \diamond (1, 0) (2, 0.00090) \diamond (2, -0.00090) \diamond (2, -0.00089) \diamond (1, -0.00399) \diamond (2, -0.00199) \diamond \\ &(2, -0.00199) \diamond (1, -0.00886) \diamond (2, -0.00443) \diamond (2, -0.00440) \diamond (2, -0.00434) \diamond (2, -0.00421) \diamond \\ &(1, -0.01924) \diamond (2, -0.00957) \diamond (2, -0.00885) \diamond (2, -0.00724) \diamond (2, -0.00364) \diamond (2, 0.00435) \diamond \\ &(1, 0.00007) \diamond (2, 1) \diamond (1, 0) \cdots (2, 1) \diamond (2, 0) \diamond (1, 0) \diamond (1, 0) \diamond (1, 0) \diamond (2, -0.00009) \diamond \\ &(2, -0.00009) \diamond (2, -0.00009) \diamond (1, -0.00040) \diamond (2, -0.00020) \diamond (2, -0.00020) \cdots (1, -0.00089) \diamond \\ &(2, -0.00045) \diamond (2, -0.00044) \diamond (2, -0.00044) \diamond (2, -0.00042) \diamond (1, -0.00194) \diamond (2, -0.00097) \diamond \\ &(2, -0.00089) \diamond (2, -0.00073) \diamond (2, -0.00037) \diamond (2, 0.00044) \diamond (1, 0.00001) \end{aligned}$$

By performing simulations we can see that $(w, \mathbf{O}(f, w))$ can be used to recover the Markov-parameters. For example, if $\mathbf{O}(f, w) = \{y_t\}_{t=0}^T$ where T is the length of w , then $y_1 = C_1 B_1 = 1 = S^f(11)$, $y_{21} = 1 = C_1 B_2 = 1 = S^f(21)$, etc., for example, $y_9 = S^f(22111) = C_1 A_1 A_1 A_2 B_2 = 0.27002$.

Note that the construction of the persistently exciting w from Theorem 4 requires the knowledge of a DTLSS realization of f . Moreover, the length of persistently exciting signal constructed in Theorem 4 can grow very fast with the number of states necessary to implement f . Below we present a subclass of input-output maps, for which the knowledge of a state-space representation is not required to construct a persistently exciting input.

Definition 16. Fix a map $\cdot^{-1} : \mathcal{U} \ni \alpha \mapsto \alpha^{-1} \in \mathcal{U}^+$. A input-output map f is said to be reversible with respect to the map $\cdot^{-1}$, if for all $\alpha \in \mathcal{U}$, $s, w \in \mathcal{U}^*$, $|s \diamond w| > 0$,

$$f(s \diamond \alpha \diamond \alpha^{-1} \diamond w) = f(sw).$$

Intuitively, f is reversible with respect to $\cdot^{-1}$, if the effect of any input $\alpha = (q, u)$ can be neutralized by the input α^{-1} . Such a property is not that uncommon, think for example of turning a valve on and off.

Remark 9 (Conditions for reversibility). *For example, if f has a realization by a DTLSS Σ of the form (1), and $Q = \{1, \dots, 2K\}$ such that for each $q \in \{1, \dots, K\}$, $A_q = A_{q+K}^{-1}$, $B_q = B_{q+K}$, then f is reversible and $(q, 0)^{-1} = (q + K, 0)$ and for $u \neq 0$, $(q, u)^{-1} = (q + K, 0) \diamond (q, -u) \diamond (q + K, 0)$,*

From the proof of Theorem 4, we obtain the following corollary.

Corollary 2. *If f is reversible with respect to $\cdot^{-1}$, then a persistently exciting input sequence w can be constructed for f . The construction does not require the knowledge of a DTLSS state-space realization of f . If the inputs α^{-1} from Definition 16 are computable from α , then the construction of w is effective.*

Proof of Corollary 2. The proof differs from that of Theorem 4 only in the definition of s^{-1} for each $s \in S$. More precisely, if f is reversible, then for each $s = (q_0, u_0) \diamond \dots \diamond (q_t, u_t) \in S$ define

$$s^{-1} = (q_t, u_t)^{-1} \diamond (q_{t-1}, u_{t-1})^{-1} \diamond \dots \diamond (q_0, u_0)^{-1} \quad (30)$$

□

The proof of Corollary 2 suggests the following algorithm for computing persistently exciting inputs for reversible input-output maps.

Algorithm 4 Computing persistently exciting inputs for reversible input-output maps

Input Selections α, β such that $\text{rank} H_{f, \alpha, \beta} = \text{rank} H_f$. **Output** **input** $w \in \mathcal{U}^+$

- 1: Consider $\Gamma(\alpha, \beta)$ from (17).
- 2: Repeat Step 3 of Algorithm 3, i.e., fix an enumeration $\Gamma(\alpha, \beta) = \{\gamma_1, \dots, \gamma_R\}$ and an enumeration of $S(\alpha, \beta) = \{s_1, \dots, s_d\}$ from (18), where $d = mR$ and for every $i = 1, \dots, d$, if $i = m(k-1) + j$ for some $k = 1, \dots, R$, $j = 1, \dots, m$ and s_i is as in (27).
- 3: For each $i = 1, \dots, d$, using (27), define s_i^{-1} as in (30) with s being replaced by s_i .
- 4: Construct w according to (19), i.e.,

$$w = s_1 \diamond s_1^{-1} \diamond \dots \diamond s_{d-1} \diamond s_{d-1}^{-1} \diamond s_d$$

5: **return** w

Remark 10 (Complexity of persistently exciting inputs for reversible input-output maps). *For reversible input-output maps the length of persistently exciting inputs and the computational complexity of computing it is much lower. Indeed, Algorithm 4 does not require the computation of the sequence π_κ from (22), length of which is $O((n+1)^{n+4})$. In fact, the length of s_i^{-1} constructed in Step 3 of Algorithm 4 will be the twice that of s_i . Therefore, if the selections α, β are chosen to be subsets of $Q^{\leq N} \times Q \times \{1, \dots, p\}$ and $Q^{\leq N} \times Q \times \{1, \dots, m\}$ respectively, n is the size of α , $\kappa \geq n$ is the size of β and $N \geq n$, then the length of w returned by the modified Algorithm 3 will be $O(N\kappa^2)$. Indeed, in this case the number of elements of $\Gamma(\alpha, \beta)$ is $O(\kappa^2)$ and their length is at most $2N$. In particular, for $N = \kappa = n$, the length of w is $O(n^3)$. This compares favorably with the length of w returned by the original Algorithm 3, which was $O((n+1)^{n+4})$. In particular, with such a choice of selections α, β , the time and memory complexity of the Algorithm 4 is polynomial and linear respectively in the state dimension.*

If the selections α and β are chosen to be $\alpha = Q^{\leq n} \times Q \times \{1, \dots, p\}$ and $\beta = Q^{\leq n} \times Q \times \{1, \dots, m\}$, then $\kappa = O(|Q|^n)$, and the length of the persistently exciting input returned by Algorithm 4 is $O(|Q|^{2n})$, i.e., it is still exponential in n . Then the storage complexity of Algorithm 4 is also exponential in n . However, this still compares favorably with the complexity of Algorithm 3.

The persistently exciting input constructed in the proof of Theorem 4 and Corollary 2 can be used to formulate a provenly system identification algorithm which finds an exact model based on data. For the sake of completeness we present the algorithm.

Algorithm 5 Identification algorithm using persistently exciting input

Inputs: Selections α, β such that $\text{rank} H_{f,\alpha,\beta} = \text{rank} H_f = n$ and the data $(w, \mathbf{O}(f, w))$ for some w of the form (19) such that $S(\alpha, \beta) = \{s_1, \dots, s_d\}$, where $S(\alpha, \beta)$ is as in (18).

Output: DTLSS $\hat{\Sigma}$.

- 1: Compute $\Gamma(\alpha, \beta)$ as in (17) and choose an enumeration $\Gamma(\alpha, \beta) = \{\gamma_1, \dots, \gamma_R\}$ and an indexing function $\phi : \{1, \dots, R\} \times \{1, \dots, m\} \rightarrow \{1, \dots, d\}$ such that

$$S^f(\gamma_k) = [f(s_{\phi(k,1)}) \quad f(s_{\phi(k,2)}) \quad \cdots \quad f(s_{\phi(k,m)})].$$

For $m = 1$; $\phi(i, 1) = i$ could be taken, for $m > 1$, $\phi(k, j) = m(k-1) + j$ could be taken.

- 2: Assume that $\mathbf{O}(f, w) = \{y_t\}_{t=0}^T$ and for all $k = 1, \dots, d$, recover the Markov-parameters $\{S^f(\gamma_i)\}_{i=1}^R$ from $\{y_t\}_{t=0}^T$ as follows:

$$\begin{aligned} S^f(\gamma_k) &= [y_{t(\phi(k,1))} \quad y_{t(\phi(k,2))} \quad \cdots \quad y_{t(\phi(k,m))}] \\ t(j) &= \sum_{i=1}^{j-1} (|s_i| + |s_i^{-1}|) + |s_j| - 1, \quad j = 1, \dots, d \end{aligned} \tag{31}$$

where $|s|$ denotes the length of a sequence of $s \in \mathcal{U}^+$, and $t(j) + 1$ equals the length of $s_1 \diamond s_1^{-1} \diamond \cdots \diamond s_{j-1} \diamond s_{j-1}^{-1} \diamond s_j$, i.e.,

$$y_{t(j)} = f(s_1 \diamond s_1^{-1} \diamond \cdots \diamond s_{j-1} \diamond s_{j-1}^{-1} \diamond s_j)$$

for all $j = 1, \dots, d$.

- 3: Construct the matrices $\mathcal{H}_{f,\alpha,\beta}$, $\mathcal{H}_{q,f,\alpha,\beta}$, $\mathcal{H}_{f,\alpha,q}$ and $\mathcal{H}_{q,f,\beta}$, $q \in Q$ from $\mathbf{O}(f, w) = \{y_t\}_{t=0}^T$ by using (31).
 - 4: Let $\hat{\Sigma}$ be the DTLSS returned by Algorithm 2 when applied to $\mathcal{H}_{f,\alpha,\beta}$, $\mathcal{H}_{q,f,\alpha,\beta}$, $\mathcal{H}_{f,\alpha,q}$ and $\mathcal{H}_{q,f,\beta}$.
 - 5: **return** $\hat{\Sigma}$
-

It is easy to see that the following holds.

Lemma 2 (Consistency of Algorithm 5). *The DTLSS returned by Algorithm 5 is a realization of f .*

In general, in order to compute the persistently exciting input w , which is necessary for Algorithm 5, the knowledge of the underlying system is required. However, Algorithm 4 for reversible input-output maps does not require the knowledge of the underlying system. Hence, Algorithm 4 in combination with Algorithm 5 is an example of a design of system identification experiment which allows exact identification of the underlying system, assuming that there is no noise. Extensions to the noisy case would require using stochastic realization theory for DTLSSs, see [40].

Example 2. Consider a DTLSS Σ of the form (1), where $Q = \{1, 2, 3, 4\}$ and A_i, B_i, C_i , $i = 1, 2$ are as in (29), and $A_3 = A_1^{-1}$ and $A_4 = A_2^{-1}$, $B_3 = B_1$, $B_4 = B_2$, $C_3 = C_1$, $C_4 = C_2$. Define the operation $\cdot^{-1} : \mathcal{U} \ni \alpha \mapsto \alpha^{-1} \in \mathcal{U}^+$ as follows: $(1, 0)^{-1} = (3, 0)$, $(2, 0)^{-1} = (4, 0)$, $(1, u)^{-1} = (3, 0) \diamond (1, -u) \diamond (1, 0)$, $(2, u)^{-1} = (4, 0) \diamond (2, -u) \diamond (4, 0)$. Let us consider the selection α, β as in Example 1. The set Γ contains 68 elements, for instance 11, 21, 211, 231, 241, 221, 121, 1211, 1221, 2211, 22111, etc. Then Algorithm 4 will return the input of length 645 whose first couple of elements are as follows

$$w = (1, 1) \diamond (1, 0) \diamond (3, 0) \diamond (3, 0) \diamond (1, -1) \diamond (3, 0) \diamond (2, 1) \diamond (1, 0) \diamond (3, 0) \diamond (4, 0) \diamond (2, -1) \diamond (4, 0) \cdots$$

The resulting input is persistently exciting. This can also be seen by simulation: if $\mathbf{O}(f, w) = \{y_t\}_{t=0}^T$ then the Markov-parameters which are necessary to identify f can be extracted from $\{y_t\}_{t=0}^T$: for instance $y_2 = C_1 B_1 = 1$, $y_6 = C_1 B_2$, $y_{65} = C_1 A_1 A_1 A_2 B_1 = 0.4493$, etc.

The resulting persistently exciting input can be used in combination with Algorithm 5 to obtain a DTLSS $\hat{\Sigma} = (n, \{\hat{A}_q, \hat{B}_q, \hat{C}_q\}_{q \in Q})$, such that $\hat{\Sigma}$ and Σ are related by an isomorphism $\mathcal{S}$, i.e.,

$\mathcal{S}A_q\mathcal{S}^{-1} = \hat{A}_q$, $\mathcal{S}B_q = \hat{B}_q$, $C_q\mathcal{S}^{-1} = \hat{C}_q$, where

$$\mathcal{S} = \begin{bmatrix} 1.93101 & -2.06335 & 2.93101 \\ -1.18326 & 0.34841 & 1.18326 \\ 9.56433 & 5.26680 & -9.56433 \end{bmatrix}.$$

In particular, the integers $t(j)$ from (31) are as follows $t(1) = 2$, $t(2) = 8$, $t(3) = 14$, etc., $t(10) = 65$, etc..

The same result can be obtained if we use Algorithm 5 for the persistently exciting input from Example 1.

Remark 11 (Extension to non-zero initial state). Note that realization theory and the notion of Markov-parameters can be extended to systems with non-zero initial states [2, 41, 42]. That is, the basic idea of the paper could in principle be extended to systems with non-zero initial state. More precisely, Theorem 4 can be extended to systems which are controllable to the initial state, i.e., for systems such that any state can be driven to the initial state for some input and switching signal. Corollary 2 can be repeated for systems with non-zero initial states, by using the extended definition of Markov-parameters [2, 41].

4.2. Universal asymptotically persistently exciting inputs

Next, we discuss classes of inputs which are asymptotically persistently exciting for all input-output maps realizable by DTLSSs.

Definition 17 (Persistence of excitation condition). An infinite input $w = \{(q_t, u_t)\}_{t=0}^\infty \in \mathcal{U}^\omega$ satisfies PE condition, if for any word $v \in Q^+$ and any integer $j \geq 1$, the limits below exist and satisfy the following conditions,

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N u_{t+j} u_t^T \chi(q_t q_{t+1} \cdots q_{t+|v|-1} = v) &= 0, \\ \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=j}^N u_{t-j} u_t^T \chi(q_{t-j} q_{t-j+1} \cdots q_{t-j+|v|-1} = v) &= 0, \\ \mathcal{R} &\stackrel{\text{def}}{=} \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N u_t u_t^T > 0, \\ \pi_v &\stackrel{\text{def}}{=} \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N \chi(q_t \cdots q_{t+|v|-1} = v) > 0, \\ \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N u_t u_t^T \chi(q_t \cdots q_{t+|v|-1} = v) &= \pi_v \mathcal{R}. \end{aligned}$$

where χ is the indicator function, i.e. $\chi(A) = 1$ if A holds and $\chi(A) = 0$ otherwise. Note that by $\mathcal{R} > 0$ we mean that $\mathcal{R}$ is a strictly positive definite $m \times m$ matrix.

Remark 12 (PE condition implies rich switching). Note that if $w \in \mathcal{U}^\omega$ satisfies the conditions of Definition 17, then the signal is rich enough, i.e. any sequence of discrete modes occurs in the switching signal infinitely often. Hence, our condition for persistence of excitation implies that the switching signal should be rich enough. This is consistent with many of the existing definitions of persistence of excitation for hybrid systems. The requirement that $\pi_v > 0$ for all $v \in Q^*$ is quite a strong one. At the end of this section we will discuss possible relaxations of this requirement.

Remark 13 (Relationship with stochastic processes). Fix a probability space $(\Omega, \mathcal{F}, P)$ and consider ergodic discrete-time stochastic processes $\mathbf{u}_t : \Omega \rightarrow \mathbb{R}^m$ and $\mathbf{q}_t : \Omega \rightarrow Q$ with values in $\mathbb{R}^m$ and Q respectively. In addition, assume the following.

- The processes $\mathbf{u}_t$ and $\mathbf{q}_t$ are independent (i.e. the σ -algebras generated by $\{\mathbf{u}_t\}_{t=0}^\infty$ and by $\{\mathbf{q}_t\}_{t=0}^\infty$ are independent).

- The stochastic process $\mathbf{u}_t$ is a colored noise, i.e. it is zero-mean, $\mathbf{u}_t$ and $\mathbf{u}_s$ are uncorrelated for $s \neq t$, and $E[\mathbf{u}_t \mathbf{u}_t^T] = \mathcal{R} > 0$, with $E[\cdot]$ denoting the expectation operator.
- For each $v \in Q^+$, $\pi_v = P(\mathbf{q}_t \cdots \mathbf{q}_{t+|v|-1} = v) > 0$.

It then follows that almost all sample paths of $\mathbf{u}_t, \mathbf{q}_t$ satisfy the PE condition of Definition 17. That is, there exists a set $A \in \mathcal{F}$, such that $P(A) = 0$ and for all $\omega \in \Omega \setminus A$, the sequence $w = \{(q_t, u_t) = (\mathbf{q}_t(\omega), \mathbf{u}_t(\omega))\}_{t=0}^\infty$ satisfies the PE condition.

Remark 14. If $\mathbf{u}_t$ is a white-noise Gaussian process and if the variables $\mathbf{q}_t$ are uniformly distributed over Q (i.e. $P(\mathbf{q}_t = q) = \frac{1}{|Q|}$ and are independent from each other and from $\{\mathbf{u}_s\}_{s=0}^\infty$, then $\mathbf{u}_t$ and $\mathbf{q}_t$ satisfy the conditions of Remark 13 and hence almost any sample path of $\mathbf{u}_t$ and $\mathbf{q}_t$ satisfies the PE condition of Definition 17.

This special case also provides a simple practical way to generate inputs which satisfy the PE conditions.

We will show that input sequences which satisfy the conditions of Definition 17 are asymptotically persistently exciting for a large class of input-output maps. The main idea behind the theorem is as follows. Consider a DTLSS Σ which is realization of f , and suppose we feed a stochastic input $\{\mathbf{q}_t, \mathbf{u}_t\}$ into Σ . Then the state $\mathbf{x}_t$ and the output response $\mathbf{y}_t$ of Σ will also be stochastic processes. Suppose that $\{\mathbf{q}_t, \mathbf{u}_t\}$ are stochastic processes which satisfy the conditions of Remark 13. It is easy to see that

$$\mathbf{y}_t = \sum_{k=0}^t C_{\mathbf{q}_t} A_{\mathbf{q}_{t-1}} \cdots A_{\mathbf{q}_{k+1}} B_{\mathbf{q}_k} \mathbf{u}_k.$$

and hence for all $r, q \in Q, v \in Q^*, |rvq| = t+1$,

$$\begin{aligned} E[\mathbf{y}_t \mathbf{u}_0^T \chi(\mathbf{q}_0 \cdots \mathbf{q}_t = rvq)] &= \\ \sum_{k=0}^t C_q A_v B_r E[\mathbf{u}_k \mathbf{u}_0^T \chi(\mathbf{q}_0 \cdots \mathbf{q}_t = rvq)] &= \\ C_q A_v B_r \mathcal{R} \pi_{rvq} = S^f(rvq) \mathcal{R} \pi_{rvq}. \end{aligned} \quad (32)$$

Hence, if we know the expectations $E[\mathbf{y}_t \mathbf{u}_0^T \chi(\mathbf{q}_0 \cdots \mathbf{q}_t = rvq)]$ for all $r, q \in Q, v \in Q^*, |rvq| = t+1, t > 0$, then we can find all the Markov-parameters of f , by the following formula

$$S^f(rvq) = E[\mathbf{y}_t \mathbf{u}_0^T \chi(\mathbf{q}_0 \cdots \mathbf{q}_{t+1} = rvq)] \mathcal{R}^{-1} \frac{1}{\pi_{rvq}}.$$

Hence, the problem of estimating the Markov-parameters reduces to estimating the expectations

$$E[\mathbf{y}_t \mathbf{u}_0^T \chi(\mathbf{q}_0 \cdots \mathbf{q}_t = rvq)]. \quad (33)$$

For practical purposes, the expectations in (33) have to be estimated from a sample-path of $\mathbf{y}_t, \mathbf{u}_t$ and $\mathbf{q}_t$. The most natural way to accomplish this is to use the formula

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=i}^N y_{i+t} u_i^T \chi(q_i \cdots q_{i+t} = rvq) \quad (34)$$

where y_t, u_t, q_t denote the value at time t of a sample-path of $\mathbf{y}_t, \mathbf{u}_t$ and $\mathbf{q}_t$ respectively. Note that y_t is in fact the output of Σ at time t , if the input $\{u_i\}_{i=0}^t$ and the switching signal $\{q_i\}_{i=0}^t$ are fed to the system.

The problem with estimating (33) by (34) is that the limit (34) may fail to exist or to converge to (33).

A particular case when (34) converges to (33) is when the process $(\mathbf{y}_t, \mathbf{u}_t, \mathbf{q}_t)$ is ergodic, [see \[43\] for the definition of ergodic processes](#). Then there exists a sample path (y_t, u_t, q_t) of $(\mathbf{y}_t, \mathbf{u}_t, \mathbf{q}_t)$ for which the limit in (34) equals the expectation (33) ; in fact ‘almost all’ sample paths will have

this property. This means that we can choose a suitable deterministic input sequence $\{u_t\}_{t=0}^\infty$ and a switching signal $\{q_t\}_{t=0}^\infty$, such that for the resulting output $\{y_t\}_{t=0}^\infty$, the limit (34) equals the expectation (33). That is, in that case the input $w = (q_0, u_0) \cdots (q_t, u_t) \cdots$ is asymptotically persistently exciting. However, proving ergodicity of $\mathbf{y}_t$ is not easy. In addition, even if $\mathbf{y}_t$ is ergodic, the particular choice of the deterministic input w for which (34) equals (33) might depend on the DTLSS itself.

For this reason, instead of using the concepts of ergodicity directly, we just show that for the input sequences w which satisfy the conditions of Definition 17, the corresponding output $\{y_t\}_{t=0}^\infty$ has the property that the limit (34) exists and it equals $S^f(rvq)\mathcal{R}\pi_{rvq}$, for any input-output map f which is realizable by a l_1 -stable DTLSS. This strategy allows us to use elementary techniques, while not compromising the practical relevance of the result.

In order to present the main result of this section, we have to define the notion of l_1 -stability of DTLSSs.

Definition 18 (Stability of DTLSSs). *A DTLSS Σ of the form (1) is called l_1 -stable, if for every $x \in \mathbb{R}^n$, the series $\sum_{v \in Q^*} \|A_v x\|_2$ is convergent.*

Remark 15 (Sufficient condition for stability). *If for all $q \in Q$, $\|A_q\|_2 < \frac{1}{|Q|}$, where $\|A_q\|_2$ is the matrix norm of A_q induced by the standard Euclidean norm, then Σ is l_1 -stable.*

Remark 16 (Asymptotic stability). *If Σ is l_1 -stable, then it is asymptotically stable, in the sense that if $s_i \in Q^*$, $i > 0$ is a sequence of words such that $\lim_{i \rightarrow \infty} |s_i| = +\infty$, then $\lim_{i \rightarrow \infty} A_{s_i} x = 0$ for all $x \in \mathbb{R}^n$.*

Intuitively it is clear why we have to restrict attention to stable systems. Recall that (4) allows us to compute the Markov-parameters of f from the responses of f to finitely many inputs. In order to obtain the response of f to several inputs from the response of f to one input, one has to find means to suppress the contribution of the current state of the system to future inputs. In §4.1 this was done by feeding inputs which drive the system back to the initial state. Unfortunately, the choice of such inputs depended on the system itself. By assuming stability, we can make sure that the effect of the past state will asymptotically diminish in time. Hence, by waiting long enough, we can approximately recover the response of f to any input.

Another intuitive explanation for assuming stability is that it is necessary for the stationarity, and hence ergodicity, of the output and state processes $\mathbf{y}_t$, $\mathbf{x}_t$.

Equipped with the definitions above, we can finally state the main result of the section.

Theorem 5 (Main result). *If w satisfies the PE conditions of Definition 17, then w is asymptotically persistently exciting for any input-output map f which admits a l_1 -stable DTLSS realization.*

The theorem above together with Remark 14 imply that white noise input and a binary noise switching signal are asymptotically persistently exciting. The proof of Theorem 5 relies on the following technical result.

Lemma 3. *Assume that Σ is a l_1 -stable DTLSS of the form (1), and assume that w satisfies the PE conditions. Let $\{y_t\}_{t=0}^\infty$ and $\{x_t\}_{t=0}^\infty$ be the output and state response of Σ to w , i.e. $y_t = f_\Sigma(w_t)$ and $x_t = x_\Sigma(0, w_t)$. Then for all $v, \beta \in Q^*$, $r, q \in Q$*

$$\begin{aligned} \pi_{rvq\beta} A_v B_r \mathcal{R} = \\ \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N x_{t+|v|+1} u_t^T \chi(t, rvq\beta) \end{aligned} \quad (35)$$

$$\begin{aligned} \pi_{rvq\beta} C_q A_v B_r \mathcal{R} = \\ \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N y_{t+|v|+1} u_t^T \chi(t, rvq\beta) \end{aligned} \quad (36)$$

Here we used the following notation: for all $s \in Q^+$,

$$\chi(t, s) = \begin{cases} 1 & \text{if } s = q_t q_{t+1} \cdots q_{t+|s|-1} \\ 0 & \text{otherwise} \end{cases}$$

Informally, Lemma 3 implies that if f is realizable by a l_1 -stable DTLSS, then the limit (34) equals (33). The proof of Lemma 3 can be found in Appendix A.

Proof of Theorem 5. For each t , denote by y_t the response of f to the first t elements of w , i.e. $y_t = f((q_0, u_0) \cdots (q_t, u_t))$. For each integer $N \in \mathbb{N}$ and for each word $v \in Q^*$, define the matrix $S_N(rvq)$ as

$$S_N(rvq) = \left(\frac{1}{N} \sum_{t=0}^N y_{t+|v|+1} u_t^T \chi(t, rvq) \right) \mathcal{R}^{-1} \frac{1}{\pi_{rvq}} \quad (37)$$

and define the matrix $M_N(v)$ by

$$\begin{bmatrix} S_N(1v1) & \cdots & S_N(Dv1) \\ \vdots & \ddots & \vdots \\ S_N(1vD) & \cdots & S_N(DvD) \end{bmatrix}$$

From Lemma 3 it follows that

$$\lim_{N \rightarrow \infty} S_N(rvq) = S^f(rvq) \quad (38)$$

and hence $\lim_{N \rightarrow \infty} M_N(v) = M^f(v)$. Hence, w is indeed asymptotically persistently exciting. $\square$

Remark 17 (Relaxation of PE condition). Assume that we restrict attention to input-output maps which are realizable by a minimal l_1 -stable DTLSS of dimension at n , and let f be such an input-output map. Furthermore, consider selections α, β such that $\text{rank} H_{f,\alpha,\beta} = \text{rank} H_f = n$ and α, β have exactly n elements. Recall that such selections always exist. It then follows that it is enough to estimate $\{S^f(v)\}_{v \in \Gamma(\alpha,\beta)}$, where $\Gamma(\alpha,\beta)$ is as in (16) in order to estimate all the Markov-parameters of f . In this case, one can replace the conditions of Definition 17, that $\pi_v > 0$ by the condition that $\pi_v > 0$ for all $v \in \Gamma(\alpha,\beta)$ and still obtain asymptotically persistently exciting inputs for f .

Indeed, consider now any $w \in \mathcal{U}^\omega$ which satisfies Definition 17 with the exception that $\pi_v > 0$ is required only for $v \in \Gamma(\alpha,\beta)$. Then Lemma 3 remains valid for this case (the proof remains literally the same) and from the proof of Theorem 5 we get that for all $v \in \Gamma(\alpha,\beta)$

$$S^f(v) = \lim_{N \rightarrow \infty} \left(\frac{1}{N} \sum_{t=0}^N y_{t+|v|-1} u_t^T \chi(t, v) \right) \mathcal{R}^{-1} \frac{1}{\pi_v} \quad (39)$$

Hence, $\{S^f(v)\}_{v \in \Gamma(\alpha,\beta)}$ can asymptotically be estimated from $(w_N, \mathbf{O}(f, w_N))$. Since the modified Algorithm 2 from Remark 6 determines a continuous map from $\{S^f(v)\}_{v \in \Gamma(\alpha,\beta)}$ to the other Markov-parameters of f , w is asymptotically persistently exciting for f .

The proof of Theorem 5 and the discussion of Remark 6 suggests the following system identification algorithm. Note that Algorithm 6 is an adaptation of the CRA method for LPV systems

Algorithm 6 Identification algorithm using PE condition

Inputs: Selections α, β such that $\text{rank} H_{f,\alpha,\beta} = \text{rank} H_f = n$, integer N , and the data $(w_N, \mathbf{O}(f, w_N))$ for some $w \in \mathcal{U}^+$ which satisfies the PE condition

Output: DTLSS Σ_N .

- 1: Construct the matrices $\mathcal{H}_{f,\alpha,\beta}^N$, $\mathcal{H}_{q,f,\alpha,\beta}^N$, $\mathcal{H}_{f,\alpha,q}^N$ and $\mathcal{H}_{q,f,\beta}^N$ $q \in Q$, by replacing in (12) – (15) every occurrence of $S^f(v)$ by $S_N(v)$ from (37).
 - 2: Let Σ_N be the DTLSS returned by Algorithm 2 when applied to $\mathcal{H}_{f,\alpha,\beta}^N$, $\mathcal{H}_{q,f,\alpha,\beta}^N$, $\mathcal{H}_{f,\alpha,q}^N$ and $\mathcal{H}_{q,f,\beta}^N$.
 - 3: **return** Σ_N
-

described in [35]. For linear systems, i.e., for $|Q| = 1$, Algorithm 2 becomes a version of the well known CCA subspace identification algorithms [44, 45].

Lemma 4 (Consistency of Algorithm 6). Assume that $\text{rank} H_{f,\alpha,\beta} = \text{rank} H_f = n$, α, β both contain exactly n elements, and w satisfies the PE condition. Let Σ_N returned by Algorithm 2 be of the form $\Sigma_N = (n_N, \{^N A_q, ^N B_q, ^N C_q\}_{q \in Q})$. Then $\lim_{N \rightarrow \infty} n_N = n$ and $\lim_{N \rightarrow \infty} (^N A_q, ^N B_q, ^N C_q) = (A_q, B_q, C_q)$, $q \in Q$ and the DTLSS $\Sigma = (n, \{A_q, B_q, C_q\}_{q \in Q})$ is a realization of f .

Proof of Lemma 4. As it was discussed in Remark 6, the matrices of the DTLSS returned by Algorithm 2 are continuous in the entries of $\mathcal{H}_{f,\alpha,\beta}^N$, $\mathcal{H}_{q,f,\alpha,\beta}^N$, $\mathcal{H}_{f,\alpha,q}^N$ and $\mathcal{H}_{q,f,\beta}^N$ $q \in Q$. By (39), the matrices $\mathcal{H}_{f,\alpha,\beta}^N$, $\mathcal{H}_{q,f,\alpha,\beta}^N$, $\mathcal{H}_{f,\alpha,q}^N$ and $\mathcal{H}_{q,f,\beta}^N$ $q \in Q$ converge to $\mathcal{H}_{f,\alpha,\beta}$, $\mathcal{H}_{q,f,\alpha,\beta}$, $\mathcal{H}_{f,\alpha,q}$ and $\mathcal{H}_{q,f,\beta}$ as $N \rightarrow \infty$. It then follows that for large enough N , $\text{rank} H_{f,\alpha,\beta}^N = \text{rank} H_{f,\alpha,\beta} = n$, and $\lim_{N \rightarrow \infty} ({}^N A_q, {}^N B_q, {}^N C_q) = (A_q, B_q, C_q)$ holds, where $\Sigma = (n, \{A_q, B_q, C_q\}_{q \in Q})$ is the DTLSS returned by Algorithm 2. The rest of the statement of the lemma follows from Theorem 3. $\square$

Example 3. Consider the input-output map f generated by the DTLSS Σ of the form (1), where $Q = \{1, 2\}$ and

$$A_1 = \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.2 & 0.2 \\ 0 & 0.2 & 0.2 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad C_1 = C_2 = [1 \quad 0 \quad 1].$$

Notice that this DTLSS is minimal, it satisfies the rank conditions of [2] for observability and span-reachability. However, the second linear subsystems is not minimal. In addition, A_1 and A_2 are not invertible. Hence, the results of Section 4.1 do not apply. However, $\|A_1\|_2 = 0.427 < 1/2$ and $\|A_2\|_2 = 0.4 < 1/2$, and hence by Remark 15 is minimal and l_1 -stable. Consider the selections $\alpha = \{(\epsilon, 1, 1), (1, 1, 1), (22, 1, 1)\}$ and $\beta = \{(\epsilon, 2, 1), (2, 1, 1), (1, 2, 1)\}$. Then the first couple of elements of $\Gamma(\alpha, \beta)$ are

$$21, 211, 2221, 121, 1211, 12221, \dots$$

By using random number generators, let us generate a time series $\{u_t, q_t\}_{t=0}^N$ such that $\{u_t\}_{t=0}^N$ is a sample of a uniformly distributed zero mean white noise process taking values in $[-0.5, 0.5]$ and $\{q_t\}_{t=0}^N$ is a sample path of a binary white noise process $\mathbf{q}_t$ taking values in $\{1, 2\}$, such that the probability of $\mathbf{q}_t = i$, $i = 1, 2$ is 0.5. We choose $N = 2 \cdot 10^7$. Let $\{y_t\}_{t=0}^N$ be the corresponding output response (we took the initial state as zero). If we estimate the Markov-parameters $S^f(v)$ for all $v \in \Gamma(\alpha, \beta)$ according to (37). The first couple of elements are as follows:

$$S_N(21) = 1.9995, \quad S_N(211) = 0.3999, \quad S_N(121) = 0.4033, \quad S_N(1211) = 0.1601$$

For comparison, the corresponding values of S^f are as follows

$$S^f(21) = 2, \quad S^f(211) = 0.4, \quad S^f(2111) = 0.4, \quad S^f(2211) = 0.16$$

In fact, we can use Algorithm 6 to compute a DTLSS from the simulated data. In this case, the matrices $\mathcal{H}_{f,\alpha,\beta}^N$, $\mathcal{H}_{q,f,\alpha,\beta}^N$, $\mathcal{H}_{f,\alpha,q}^N$ and $\mathcal{H}_{q,f,\beta}^N$ $q \in Q$, are as follows

$$\begin{aligned} \mathcal{H}_{f,\alpha,\beta}^N &= \begin{bmatrix} 1.9995 & 0.4021 & 0.3999 \\ 0.3999 & 0.1601 & 0.1174 \\ 0.0816 & 0.0655 & 0.0139 \end{bmatrix}, \quad \mathcal{H}_{1,f,\alpha,\beta}^N = \begin{bmatrix} 0.3999 & 0.1601 & 0.1174 \\ 0.1174 & 0.0466 & 0.0442 \\ 0.0139 & 0.0075 & 0.0066 \end{bmatrix}, \\ \mathcal{H}_{2,f,\alpha,\beta}^N &= \begin{bmatrix} 0.2026 & 0.1611 & 0.0395 \\ 0.0828 & 0.0643 & 0.0179 \\ 0.0323 & 0.0273 & 0.0094 \end{bmatrix}, \quad \mathcal{H}_{f,\alpha,1}^N = \begin{bmatrix} 2.0000 \\ 0.5989 \\ 0.1611 \end{bmatrix}, \quad \mathcal{H}_{f,\alpha,2}^N = \begin{bmatrix} 1.9995 \\ 0.3999 \\ 0.0816 \end{bmatrix} \\ \mathcal{H}_{1,f,\beta}^N &= \mathcal{H}_{2,f,\beta}^N = [2.0004 \quad 0.4002 \quad 0.3996] \end{aligned}$$

For comparison, the true matrices $\mathcal{H}_{f,\alpha,\beta}$, $\mathcal{H}_{q,f,\alpha,\beta}$, $\mathcal{H}_{f,\alpha,q}$ and $\mathcal{H}_{q,f,\beta}$ $q \in Q$ as as follows:

$$\begin{aligned}\mathcal{H}_{f,\alpha,\beta} &= \begin{bmatrix} 2.0000 & 0.4000 & 0.4000 \\ 0.4000 & 0.1600 & 0.1200 \\ 0.0800 & 0.0640 & 0.0160 \end{bmatrix}, \quad \mathcal{H}_{1,f,\alpha,\beta} = \begin{bmatrix} 0.4000 & 0.1600 & 0.1200 \\ 0.1200 & 0.0480 & 0.0480 \\ 0.0160 & 0.0064 & 0.0096 \end{bmatrix}, \\ \mathcal{H}_{2,f,\alpha,\beta} &= \begin{bmatrix} 0.2000 & 0.1600 & 0.0400 \\ 0.0800 & 0.0640 & 0.0160 \\ 0.0320 & 0.0256 & 0.0064 \end{bmatrix} \\ \mathcal{H}_{f,\alpha,1} &= \begin{bmatrix} 2.0000 \\ 0.6000 \\ 0.1600 \end{bmatrix}, \quad \mathcal{H}_{f,\alpha,2} = \begin{bmatrix} 2.0000 \\ 0.4000 \\ 0.0800 \end{bmatrix} \\ \mathcal{H}_{1,f,\beta} &= \mathcal{H}_{2,f,\beta} = [2.0000 \quad 0.4000 \quad 0.4000]\end{aligned}$$

The matrices of the DTLSS $\Sigma_N = (3, \{\hat{A}_q, \hat{B}_q, \hat{C}_q\}_{q \in Q})$ returned by Algorithm 6 can be computed by applying Algorithm 2 to $\mathcal{H}_{f,\alpha,\beta}^N$, $\mathcal{H}_{q,f,\alpha,\beta}^N$, $\mathcal{H}_{f,\alpha,q}^N$ and $\mathcal{H}_{q,f,\beta}^N$ $q \in Q$ instead of $\mathcal{H}_{f,\alpha,\beta}$, $\mathcal{H}_{q,f,\alpha,\beta}$, $\mathcal{H}_{f,\alpha,q}$ and $\mathcal{H}_{q,f,\beta}$ $q \in Q$ and they are as follows:

$$\begin{aligned}\hat{A}_1 &= \begin{bmatrix} -0.0000 & 0.0099 & -0.0392 \\ -0.0000 & 0.0341 & 0.0577 \\ 1.0000 & 0.3165 & 0.4313 \end{bmatrix}, \quad \hat{A}_2 = \begin{bmatrix} -0.0143 & 0.0032 & 0.0011 \\ 0.4930 & 0.4197 & 0.1550 \\ 0.0822 & -0.0355 & -0.0629 \end{bmatrix} \\ \hat{B}_1 &= \begin{bmatrix} 0.3189 \\ 1.7007 \\ 1.6965 \end{bmatrix}, \quad \hat{B}_2 = \begin{bmatrix} 1.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix} \\ \hat{C}_1 &= \hat{C}_2 = [2.0004 \quad 0.4002 \quad 0.3996]\end{aligned}$$

Simulations reveal that the output response of $\hat{\Sigma}_N$ is very close to Σ , the Best Fit Rate (BFR) is 99.5%. For illustration, let us consider the DTLSS $\bar{\Sigma} = (3, \{\bar{A}_q, \bar{B}_q, \bar{C}_q\}_{q \in Q})$ returned by the realization algorithm Algorithm 2:

$$\begin{aligned}\bar{A}_1 &= \begin{bmatrix} 0 & 0 & -0.04 \\ 0 & 0 & 0.1 \\ 1 & 0.4 & 0.4 \end{bmatrix}, \quad \bar{A}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0.4 & 0.1 \\ 0 & 0 & 0 \end{bmatrix} \quad \bar{B}_1 = \begin{bmatrix} 0.3333 \\ 1.6667 \\ 1.6667 \end{bmatrix} \quad \bar{B}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ \bar{C}_1 &= \bar{C}_2 = [2 \quad 0.4 \quad 0.4]\end{aligned}$$

Note that $\mathcal{S} = \begin{bmatrix} 0.3333 & -0.6667 & 0.6667 \\ -0.8333 & 1.6667 & 0.8333 \\ 1.6667 & 1.6667 & -1.6667 \end{bmatrix}$ an isomorphism from the original system Σ to $\bar{\Sigma}$.

It is easy to see that the parameters of $\hat{\Sigma}$ and $\bar{\Sigma}$ are numerically close, the maximal relative error $\max\{\frac{\|\hat{A}_q - \bar{A}_q\|_2}{\|\bar{A}_q\|_2}, \frac{\|\hat{B}_q - \bar{B}_q\|_2}{\|\bar{B}_q\|_2}, \frac{\|\hat{C}_q - \bar{C}_q\|_2}{\|\bar{C}_q\|_2} \mid q \in Q\}$ is 0.1839. Simulations revealed that the more data points are used, the closer the matrices $\mathcal{H}_{f,\alpha,\beta}^N$, $\mathcal{H}_{q,f,\alpha,\beta}^N$, $\mathcal{H}_{f,\alpha,q}^N$ and $\mathcal{H}_{q,f,\beta}^N$ $q \in Q$ are $\mathcal{H}_{f,\alpha,\beta}$, $\mathcal{H}_{q,f,\alpha,\beta}$, $\mathcal{H}_{f,\alpha,q}$ and $\mathcal{H}_{q,f,\beta}$, $q \in Q$. Moreover, by increasing N , the output responses of $\hat{\Sigma}_N$ get closer to those of Σ . In addition, the parameters of $\hat{\Sigma}_N$ get closer to that of $\bar{\Sigma}$. That is, Algorithm 6 behaves truly as an asymptotically consistent system identification algorithm. Note that for a good performance, we need a lot of data points. Whether this is an inherent feature of the algorithm or it is due to the choice of the example remains a topic of future research.

Remark 18 (Extension of PE condition to non-zero initial state). The results of this section rely on exponential asymptotic stability of the underlying system. For such systems, the effect of the initial state on the output decays exponentially fast. That is, the effect of the initial state is not visible in the long run. For this reason, it is not clear how the derived results could be extended to systems with non-zero initial state.

Remark 19 (Length of PE input required for identifying the system with a desired accuracy). From the definition of asymptotic persistence of excitation it follows that the longer is the input

sequence, the more accurate are the estimates of the Markov parameters. In turn, more accurate estimates of the Markov-parameters lead to less error in the identified system. However, in general, for a given N , the accuracy of the system identification algorithm depends on two factors: the rate of convergence of the estimate (37) to the true Markov-parameter and the degree of robustness of Algorithm 2. None of these questions are easy to answer and they are left for future research. As the results of Example 3 indicate, in order to have a satisfactory accuracy, Algorithm 6 may require long input sequences. Whether this is inherent to the problem of identification of switched systems or can be improved by proposing better identification algorithms, remains a topic of future research.

5. Conclusions

We defined persistence of excitation for input signals of linear switched systems. We showed existence of persistently exciting input sequences and we identified several classes of input signals which are persistently exciting.

Future work includes finding less restrictive conditions for persistence of excitation and extending the obtained results to other classes of hybrid systems.

Appendix A. Technical proofs

The proof of Lemma 3 relies on the following result.

Lemma 5. *With the notation and assumptions of Lemma 3, for all $v \in Q^+$,*

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N x_t u_t^T \chi(t, v) = 0$$

The intuition behind Lemma 5 is as follows. Each x_t is a linear combination of inputs $u_0, \dots, u_{t-1}$. Hence, $\frac{1}{N} \sum_{t=0}^N x_t u_t^T$ can be expressed as linear combination of terms $\frac{1}{N} \sum_{t=k}^N u_{t-k} u_t^T \chi(t, s)$ for some $s \in Q^*$, $k = 1, \dots, N$. Since each such term converges to 0 as $N \rightarrow \infty$, intuitively their linear combination should converge to 0 as well. Unfortunately, the number of summands of the above increases with N . In order to deal with this difficulty a technique similar to the M -test for double series has to be used. The assumption that Σ is l_1 -stable is required for this technique to work.

Proof of Lemma 3. We start with the proof of (35). The proof goes by induction on the length of v .

If $v = \epsilon$, then

$$\begin{aligned} \frac{1}{N} \sum_{t=0}^N x_{t+1} u_t^T \chi(t, r\beta) &= \\ \frac{1}{N} \sum_{t=0}^N (A_{q_t} x_t + B_{q_t} u_t) u_t^T \chi(t, r\beta) &= \\ \frac{1}{N} \sum_{t=0}^N A_{q_t} x_t u_t^T \chi(t, r\beta) + \frac{1}{N} \sum_{t=0}^N B_{q_t} u_t u_t^T \chi(t, r\beta). \end{aligned} \tag{A.1}$$

Notice $A_{q_t} x_t u_t^T \chi(t, r\beta) = A_r x_t u_t^T \chi(t, r\beta)$ and $B_{q_t} u_t u_t^T \chi(t, r\beta) = B_r u_t u_t^T \chi(t, r\beta)$. Hence,

$$\begin{aligned} \frac{1}{N} \sum_{t=0}^N A_{q_t} x_t u_t^T \chi(t, r\beta) + \frac{1}{N} \sum_{t=0}^N B_{q_t} u_t u_t^T \chi(t, r\beta) &= \\ A_r \left(\frac{1}{N} \sum_{t=0}^N x_t u_t^T \chi(t, r\beta) \right) + B_r \left(\frac{1}{N} \sum_{t=0}^N u_t u_t^T \chi(t, r\beta) \right) \end{aligned} \tag{A.2}$$

From the assumptions on w it follows that

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N u_t u_t^T \chi(t, r\beta) = \mathcal{R} \pi_{r\beta}$$

Hence, from the PE conditions and Lemma 5 we get that

$$\begin{aligned} & \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N x_{t+1} u_t^T \chi(t, r\beta) = \\ & A_r \left(\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^n x_t u_t^T \chi(t, r\beta) \right) + \\ & + B_r \left(\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^n u_t u_t^T \chi(t, r\beta) \right) = \\ & A_r 0 + B_r \mathcal{R} \pi_{r\beta} = \pi_{r\beta} B_r \mathcal{R}, \end{aligned}$$

i.e. (35) holds.

Assume that (35) holds for all words of length at most L , and assume that $v = wq$, $|w| = L$ for some $w \in Q^*$ and $q \in Q$. Then by the induction hypothesis and the assumptions on w

$$\begin{aligned} & \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N x_{t+L+2} u_t^T \chi(t, rwq\beta) = \\ & \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N A_q x_{t+L+1} u_t^T \chi(t, rwq\beta) + \\ & + \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N B_q u_{t+L+1} u_t^T \chi(t, rwq\beta) = \\ & = A_q A_w B_r \pi_{rwq\beta} + B_r 0 = A_{wq} B_r \mathcal{R} \pi_{rwq\beta}. \end{aligned} \tag{A.3}$$

Finally, we prove (36). Notice that

$$y_{t+|v|+2} u_t^T \chi(q, t, rvq\beta) = C_q x_{t+|v|+2} u_t^T \chi(t, rvq\beta)$$

and hence by applying (35),

$$\begin{aligned} & \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N y_{t+|v|+2} u_t^T \chi(t, rvq\beta) = \\ & C_q \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N x_{t+|v|+2} u_t^T \chi(t, rvq\beta) = \\ & C_q A_v B_r \mathcal{R} \pi_{rvq\beta}. \end{aligned}$$

□

Proof of Lemma 5. Notice that

$$\begin{aligned}
& \sum_{t=1}^N x_t u_t^T \chi(t, v) = \\
& \sum_{t=1}^N \sum_{j=1}^{\textcolor{red}{t}} A_{q_{t-1}} \cdots A_{q_j} B_{q_{j-1}} u_{j-1} u_t^T \chi(t, v) = \\
& \sum_{k=1}^{\textcolor{red}{N}} \left(\sum_{t=k}^N A_{q_{t-1}} \cdots A_{q_{t-k+1}} B_{q_{t-k}} u_{t-k} u_t^T \chi(t, v) \right) = \\
& \sum_{r \in Q} \sum_{k=0}^{N-1} \sum_{|s|=k} A_s B_r \sum_{t=k+1}^N u_{t-k-1} u_t^T \chi(t-k-1, rsv) = \\
& \sum_{i=1}^{\mathbf{N}(N)} \sum_{r \in Q} A_{v_i} B_r \sum_{t=|v_i|+1}^N u_{t-|v_i|-1} u_t^T \chi(t-|v_i|-1, rv_i v).
\end{aligned}$$

In the last step we used the lexicographic ordering of Q^* from Remark 2. It then follows that

$$\begin{aligned}
& \frac{1}{N} \sum_{t=1}^N x_t u_t^T \chi(t, v) = \\
& \sum_{r \in Q} \sum_{i=1}^{\mathbf{N}(N)} A_{v_i} B_r \frac{1}{N} \sum_{t=|v_i|+1}^N u_{t-|v_i|-1} u_t^T \chi(t-|v_i|-1, rv_i v).
\end{aligned}$$

Define

$$\begin{aligned}
b_{i,N}^r &= \frac{1}{N} \sum_{t=|v_i|+1}^N u_{t-|v_i|-1} u_t^T \chi(t-|v_i|-1, rv_i v) \\
a_{i,N}^r &= A_{v_i} B_r b_{i,N}^r.
\end{aligned}$$

Then the statement of the lemma can be shown by showing that for all $r \in Q$,

$$\lim_{N \rightarrow \infty} \sum_{i=1}^{\mathbf{N}(N)} a_{i,N}^r = 0.$$

To this end, notice from the PE conditions that

$$\begin{aligned}
& \lim_{N \rightarrow \infty} a_{i,N}^r = \\
& A_{v_i} B_r \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=|v_i|+1}^N u_{t-k-1} u_t^T \chi(t-k-1, rv_i v) = 0.
\end{aligned}$$

Moreover, for a fixed N and i , we can get the following estimate

$$\|a_{i,N}^r\|_2 \leq \|A_{v_i} B_r\|_2 \|b_{i,N}^r\|_2.$$

If we can show that $\|b_{i,N}^r\|_2$ is bounded by a number K , then we get that

$$\|a_{i,N}^r\|_2 \leq \|A_{v_i} B_r\|_2 K.$$

The latter inequality is already sufficient to finish the proof. Indeed, let $D_i^r = \|A_{v_i} B_r\|_2 K$ and

notice from the l_1 -stability assumption on the realization Σ that

$$\sum_{i=1}^{\infty} D_i^r = K \sum_{v \in Q^*} \|A_v B_r\|_2$$

is convergent. Hence, we get that for every $\epsilon > 0$ there exists a I_ϵ such that

$$\sum_{i=I_\epsilon+1}^{\infty} D_i^r < \epsilon/2.$$

For every $N > I_\epsilon$,

$$\begin{aligned} \left\| \sum_{i=1}^{\mathbf{N}(N)} a_{i,N}^r \right\|_2 &= \left\| \sum_{i=1}^{I_\epsilon} a_{i,N}^r + \sum_{i=I_\epsilon+1}^{\mathbf{N}(N)} a_{i,N}^r \right\|_2 \leq \\ &\sum_{i=1}^{I_\epsilon} \|a_{i,N}^r\|_2 + \sum_{i=I_\epsilon+1}^{\mathbf{N}(N)} D_i^r < \sum_{i=1}^{I_\epsilon} \|a_{i,N}^r\|_2 + \epsilon/2. \end{aligned}$$

Since $\lim_{N \rightarrow \infty} a_{i,N}^r = 0$, there exists $N_\epsilon \in \mathbb{N}$ such that for all $N > N_\epsilon$, $\|a_{i,N}^r\|_2 < \frac{\epsilon}{2I_\epsilon}$. Define $\hat{N}_\epsilon$ to be an integer such that $\hat{N}_\epsilon > N_\epsilon$ and $\mathbf{N}(\hat{N}_\epsilon) > I_\epsilon$. Then for every $N > \hat{N}_\epsilon$, $\mathbf{N}(N) \geq \mathbf{N}(\hat{N}_\epsilon) > I_\epsilon$ and

$$\begin{aligned} \left\| \sum_{i=1}^{\mathbf{N}(N)} a_{i,N}^r \right\|_2 &\leq \sum_{i=1}^{I_\epsilon} \|a_{i,N}^r\|_2 + \epsilon/2 < \\ I_\epsilon \frac{\epsilon}{2I_\epsilon} + \epsilon/2 &= \epsilon/2 + \epsilon/2 = \epsilon. \end{aligned}$$

In other words, $\lim_{N \rightarrow 0} \sum_{i=1}^{\mathbf{N}(N)} a_{i,N}^r = 0$.

It is left to show that $\|b_{i,N}^r\|_2 \leq K$ for some $K > 0$ and for all $i = 1, 2, \dots, r \in Q$.

$$\begin{aligned} \|b_{i,N}^r\|_2 &\leq \left\| \frac{1}{N} \sum_{t=|v_i|+1}^N u_{t-|v_i|-1} u_t^T \chi(t-|v_i|-1, rv_i v) \right\|_2 \leq \\ &\left\| \frac{1}{N} \sum_{t=|v_i|+1}^N u_{t-|v_i|-1} u_t^T \chi(t-|v_i|-1, rv_i v) \right\|_F = \\ &\left[\sum_{i,j=1}^m \frac{1}{N^2} \left\{ \sum_{t=|v_i|+1}^N (u_{t-|v_i|-1})_i \chi(t-|v_i|-1, rv_i v) (u_t)_j \right\}^2 \right]^{1/2}. \end{aligned} \tag{A.4}$$

where $\|\cdot\|_F$ denotes the matrix Frobenius-norm, and $\|\cdot\|_2$ denotes the matrix norm induced by the Euclidean norm. The application of the Cauchy-Schwartz inequality to $(\sum_{t=|v_i|+1}^N (u_{t-|v_i|-1})_i \chi(t-|v_i|-1, rv_i v) (u_t)_j)^2$ leads to

$$\begin{aligned} &\left[\sum_{t=|v_i|+1}^N (u_{t-|v_i|-1})_i \chi(t-|v_i|-1, rv_i v) (u_t^T)_j \right]^2 \leq \\ &\left(\sum_{t=|v_i|+1}^N (u_{t-|v_i|-1})_i^2 \chi(t-|v_i|-1, rv_i v) \right) \left(\sum_{t=|v_i|+1}^N (u_t)_j^2 \right). \end{aligned} \tag{A.5}$$

Notice that $(u_{t-|v_i|-1})_i^2 \chi(t-|v_i|-1, rv_i v) \leq (u_{t-|v_i|-1})_i^2$, since $\chi(t-|v_i|-1, rv_i v) \in [0, 1]$. Hence,

$$\begin{aligned} & \sum_{t=|v_i|+1}^N (u_{t-|v_i|-1})_i^2 \chi(t-|v_i|-1, rv_i v) \leq \\ & \leq \sum_{t=|v_i|+1}^N (u_{t-|v_i|-1})_i^2 \leq \sum_{t=0}^N (u_t)_i^2. \end{aligned}$$

Similarly,

$$\sum_{t=|v_i|+1}^N (u_t)_j^2 \leq \sum_{t=0}^N (u_t)_j^2.$$

Combining these remarks with (A.5), we obtain

$$\begin{aligned} & \left[\frac{1}{N^2} \sum_{t=|v_i|+1}^N (u_{t-|v_i|-1})_i \chi(t-|v_i|-1, rv_i v) (u_t^T)_j \right]^2 \\ & \leq \left(\frac{1}{N} \sum_{t=0}^N (u_t)_i^2 \right) \left(\frac{1}{N} \sum_{t=0}^N (u_t)_j^2 \right). \end{aligned} \tag{A.6}$$

Notice that $\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=0}^N (u_t)_i^2 = \mathcal{R}_{ii}$ and hence $\frac{1}{N} \sum_{t=0}^N (u_t)_i^2$ is bounded from above by some positive number K_i . Using this fact and by substituting (A.6) into (A.4), we obtain

$$\|b_{i,N}^r\|_2 \leq \left(\sum_{i,j=1}^m K_i K_j \right)^{1/2}.$$

Hence, if we set $K = \sum_{i,j=1}^m K_i K_j$, then $\|b_{i,N}^r\|_2 \leq K$, which is what had to be shown. $\square$

References

- [1] Z. Sun, S. S. Ge, Switched linear systems : control and design, Springer, London, 2005.
- [2] M. Petreczky, L. Bako, v. J. Schuppen, Realization theory for discrete-time linear switched systems, Automatica 49 (11) (2013) 3337–3344.
- [3] R. Vidal, A. Chiuso, S. Sastry, Observability and identifiability of jump linear systems, in: Proc. IEEE Conf. Dec. and Control, 2002, pp. 3614 – 3619.
- [4] Y. Ma, R. Vidal, Identification of deterministic switched arx systems via identification of algebraic varieties, in: Hybrid Systems: Computation and Control, Vol. 3414 of Lecture Notes in Computer Science, 2005, pp. 449 – 465.
- [5] Y. Hashambhoy, R. Vidal, Recursive identification of switched ARX models with unknown number of models and unknown orders, in: IEEE Conference on Decision and Control, 2005.
- [6] Y. Ma, R. Vidal, A closed form solution to the identification of hybrid ARX models via the identification of algebraic varieties, in: Hybrid Systems: Computation and Control, 2005, pp. 449–465.
- [7] S. Paoletti, A. Juloski, G. Ferrari-Trecate, R. Vidal, Identification of hybrid systems: A tutorial, European Journal of Control 13 (2-3) (2007) 242 – 260.
- [8] R. Vidal, Recursive identification of switched ARX systems, Automatica 44 (9) (2008) 2274 – 2287.
- [9] A. Juloski, W. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, J. Niessen, Comparison of four procedures for the identification of hybrid systems, in: Hybrid Systems: Computation and Control, Vol. 3414 of LNCS, Springer-Verlag, Berlin, 2005, pp. 354–369.

- [10] G. Ferrari-Trecate, M. Musellu, D. Liberati, M. Morari, A clustering technique for the identification of piecewise-affine systems, *Automatica* 39 (2003) 205–217.
- [11] A. Juloski, S. Weiland, M. Heemels, A bayesian approach to identification of hybrid systems, in: *Proceedings of 43rd IEEE Conference on Decision and Control*, 2004.
- [12] A. Juloski, Observer design and identification methods for hybrid systems: Theory and experiments, Ph.D. thesis, Eindhoven University of Technology (2004).
- [13] L. Bako, G. Mercère, R. Vidal, S. Lecoeuche., Identification of switched linear state space models without minimum dwell time, in: *IFAC Symposium on System Identification*, Saint Malo, France, 2009.
- [14] L. Bako, G. Mercère, S. Lecoeuche, Online structured subspace identification with application to switched linear systems, *International Journal of Control* 82 (2009) 1496–1515.
- [15] J. Roll, A. Bemporad, L. Ljung, Identification of piecewise affine systems via mixed-integer programming, *Automatica* 40 (1) (2004) 37–50.
- [16] E. Fox, Bayesian nonparametric learning of complex dynamical phenomena, Ph.D. thesis, MIT, Cambridge, MA (2009).
- [17] V. Verdult, M. Verhaegen, Subspace identification of piecewise linear systems, in: *Proc.Conf. Decision and Control*, 2004.
- [18] S. Paoletti, J. Roll, A. Garulli, A. Vicino, Input/output realization of piecewise affine state space models, in: *46th IEEE Conf. on Dec. and Control*, 2007.
- [19] L. Bako, Identification of switched linear systems via sparse optimization, *Automatica* doi:10.1016/j.automatica.2011.01.036 (2011).
- [20] L. Bako, Analysis of the least sum-of-minimums estimator for switched systems, *IEEE Transactions on Automatic Control* 66 (8) (2021) 3733–3740.
- [21] F. Lauer, G. Bloch, Hybrid System Identification: Theory and Algorithms for Learning Switching Models, Vol. 478 of *Lecture Notes in Control and Information Sciences*, Springer International Publishing, 2019.
- [22] V. Breschi, D. Piga, A. Bemporad, Learning hybrid models with logical and continuous dynamics via multiclass linear separation, 2016, pp. 353–358.
- [23] R. Vidal, S. Sastry, A. Chiuso, Observability of linear hybrid systems, in: *Hybrid Systems: Computation and Control*, 2003.
- [24] A. Hiskens, Identifiability of hybrid system models, in: *Proceedings of the IEEE International Conference on Control Applications*, Anchorage, AK, 2000.
- [25] R. Vidal, S. Soatto, Y. Ma, S. Sastry, An algebraic geometric approach to the identification of a class of linear hybrid systems, in: *IEEE Conference on Decision and Control*, 2003.
- [26] L. Ljung, *System Identification: Theory for the user* (2nd Ed.), PTR Prentice Hall., Upper Saddle River, USA, 1999.
- [27] J. C. Willems, P. Rapisarda, I. Markovsky, B. L. D. Moor, A note on persistency of excitation, *Systems & Control Letters* 54 (4) (2005) 325 – 329.
- [28] M. Petreczky, L. Bako, On the notion of persistence of excitation for linear switched systems, in: *50th IEEE Conference on Decision and Control (CDC)*, 2011.
- [29] M. Petreczky, L. Bako, On the notion of persistence of excitation for linear switched systems, Tech. Rep. arXiv:1103.1349v1, ArXiv (2011).
- [30] F. Gécseg, I. Peák, Algebraic theory of automata, Akadémiai Kiadó, Budapest, 1972.

- [31] S. Eilenberg, Automata, Languages and Machines, Academic Press, New York, London, 1974.
- [32] M. Petreczky, L. Bako, J. van Schuppen, Identifiability of discrete-time linear switched systems, in: Hybrid Systems: Computation and Control, ACM, 2010.
- [33] M. Petreczky, J. H. van Schuppen, Partial-realization of linear switched systems: A formal power series approach, Automatica 47 (10) (2011) 2177–2184.
- [34] M. Bastug, M. Petreczky, R. Wisniewski, J. Leth, Model reduction by moment matching for linear switched systems, IEEE Transactions on Automatic Control 61 (2016) 3422–3437.
- [35] P. Cox, R. Tóth, M. Petreczky, Towards efficient maximum likelihood estimation of lpv-ss models, Automatica 97 (2018) 392–403.
- [36] M. Petreczky, R. Peeters, Spaces of nonlinear and hybrid systems representable by recognizable formal power series, in: Proceedings 19th International Symposium on Mathematical Theory of Networks and Systems, 2010, pp. 1051–1058.
- [37] M. Bastug, Model reduction of linear switched systems and lpv state-space models, Ph.D. thesis, Aalborg University (2016).
- [38] S. Ge, Z. Sun, T. Lee, Reachability and controllability of switched linear discrete-time systems, IEEE Trans. Automatic Control 46 (9) (2001) 1437 – 1441.
- [39] Z. Ji, G. Feng, X. Guo, A constructive approach to reachability realization of discrete-time switched linear systems, Systems & Control Letters 56 (11) (2007) 669–677.
- [40] M. Petreczky, R. Vidal, Realization theory for a class of stochastic bilinear systems, IEEE Transactions on Automatic Control 63 (1) (2017) 69–84.
- [41] M. Petreczky, Realization theory of linear and bilinear switched systems: A formal power series approach: Part i, ESAIM Control, Optimization and Calculus of Variations DOI 10.1051/cocv/2010014 (2010).
- [42] J. van den Hof, System theory and system identification of compartmental systems, Ph.D. thesis, University of Groningen (1996).
- [43] P. Billingsley, Probability and measure, Wiley, 1986.
- [44] T. Katayama, Subspace Methods for System Identification, Springer-Verlag, 2005.
- [45] P. van Overschee, B. D. Moor, Subspace Identification for Linear Systems, Kluwer Academic Publishers, 1996.