Model Reduction of Linear Switched Systems by Restricting Discrete Dynamics

Bastug, Mert; Petreczky, Mihaly; Wisniewski, Rafal; Leth, John-Josef

Published in:
Annual Conference on Decision and Control, 2014 IEEE 53rd

DOI (link to publication from Publisher):
10.1109/CDC.2014.7040079

Publication date:
2014

Document Version
Accepted author manuscript, peer reviewed version

Model Reduction of Linear Switched Systems by Restricting Discrete Dynamics

Mert Baştuğ[1,2], Mihály Petreczky[2], Rafael Wisniewski[1] and John Leth[1]

***Abstract*— We present a procedure for reducing the number of continuous states of discrete-time linear switched systems, such that the reduced system has the same behavior as the original system for a subset of switching sequences. The proposed method is expected to be useful for abstraction based control synthesis methods for hybrid systems.**

I. INTRODUCTION

A discrete-time linear switched system [11], [18] (abbreviated by DTLSS) is a discrete-time hybrid system of the form

$$\Sigma \begin{cases} x(t+1) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) \text{ and } x(0) = x_0 \\ y(t) = C_{\sigma(t)}x(t), \end{cases} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the continuous state, $y(t) \in \mathbb{R}^p$ the continuous output, $u(t) \in \mathbb{R}^m$ is the continuous input, $\sigma(t) \in Q = \{1, \ldots, D\}$, $D > 0$ is the discrete state (switching signal). A_q, B_q, C_q are matrices of suitable dimension for $q \in Q$. A more rigorous definition of DTLSSs will be presented later on. For the purposes of this paper, $u(t)$ and $\sigma(t)$ will be viewed as externally generated signals.

Contribution of the paper: Consider a discrete-time linear switched system Σ of the form (1), and a set L which describes the admissible set of switching sequences. In this paper, we will present an algorithm for computing another DTLSS

$$\bar{\Sigma} \begin{cases} \bar{x}(t+1) = \bar{A}_{\sigma(t)}\bar{x}(t) + \bar{B}_{\sigma(t)}u(t) \text{ and } \bar{x}(0) = \bar{x}_0 \\ \bar{y}(t) = \bar{C}_{\sigma(t)}\bar{x}(t) \end{cases} \tag{2}$$

such that for any switching sequence $\sigma(0)\cdots\sigma(t) = q_0 \cdots q_t \in L$ and continuous inputs $u(0), \ldots, u(t-1)$, the output at time t of (1) equals the output of (2), i.e., $y(t) = \bar{y}(t)$ and the number of state variables of (2) is smaller than that of (1). In short, for any sequence of discrete states from L, the input-output behaviors of Σ and $\bar{\Sigma}$ coincide and the size of $\bar{\Sigma}$ is smaller.

Motivation: Realistic plant models of industrial interest tend to be quite large and in general, the smaller is the plant model, the smaller is the resulting controller and the computational complexity of the control synthesis or verification algorithm. This is especially apparent for hybrid systems, since in this case, the computational complexity of control or verification algorithms is often exponential in the number of continuous states [19]. The particular model reduction problem formulated in this paper was motivated by the observation that in many instances, we are interested in the behavior of the model only for certain switching sequences.

DTLSSs with switching constraints occur naturally in a large number of applications. Such systems arise for example when the supervisory logic of the switching law is (partially) fixed. Note that verification or control synthesis of DTLSSs can be computationally demanding, especially if the properties or control objectives of interest are discrete [5]. The results of the paper could be useful for verification or control of such systems, if the properties of interest or the control objectives depend only on the input-output behavior. In this case, we could replace the original DTLSS Σ by the reduced order DTLSS $\bar{\Sigma}$ whose input-output behavior for all the admissible switching sequences coincides with that of Σ. We can then perform verification or control synthesis for $\bar{\Sigma}$ instead of Σ. If $\bar{\Sigma}$ satisfies the desired input-output properties, then so does Σ. The same applies for piecewise-affine hybrid systems, which can be modelled as a feedback interconnection of a linear switched system of the form (1) with a discrete event generator ϕ which generates the next discrete state based on the past discrete states and past outputs [6], [20]. Likewise, assume that it is desired to design a control law for Σ which ensures that the switching signal generated by the closed-loop system belongs to a certain prefix closed set L. Such problems arise in various settings for hybrid systems [19]. Again, this problem is solvable for Σ if and only if it is solvable for $\bar{\Sigma}$, and the controller which solves this problem for $\bar{\Sigma}$ also solves it for Σ.

Related work: Results on realization theory of linear switched systems with constrained switching appeared in [14]. However, [14] does not yield a model reduction algorithm, see Remark 1 in [2] for a detailed discussion. The algorithm presented in this paper bears a close resemblance to the moment matching method of [1], but its result and its scope of application are different. The subject of model reduction for hybrid and switched systems was addressed in several papers [3], [23], [12], [4], [8], [21], [22], [7], [9], [10], [13], [17]. However, none of them deals with the problem addressed in this paper.

Outline: In Section II, we fix the notation and terminology of the paper. In Section III, we present the formal definition and main properties of DTLSSs . In Section IV we give the precise problem statement. In Section V, we recall the con-

[1]Department of Electronic Systems, Automation and Control, Aalborg University, 9220 Aalborg, Denmark mertb@es.aau.dk, raf@es.aau.dk, jjl@es.aau.dk

[2]Department of Computer Science and Automatic Control (UR Informatique et Automatique), École des Mines de Douai, 59508 Douai, France mihaly.petreczky@mines-douai.fr

This work was partially supported by ESTIREZ project of Region Nord-Pas de Calais, France.

cept of Markov parameters, and we present the fundamental theorem and corollaries which form the basis of the model reduction by moment matching procedure. The algorithm itself is stated in Section VI in detail. Finally, in Section VII the algorithm is illustrated on a numerical example.

II. Preliminaries: notation and terminology

Denote by $\mathbb{N}$ the set of natural numbers including 0.

Consider a non-empty set Q which will be called the *alphabet*. Denote by Q^* the set of finite sequences of elements of Q. The elements of Q^* are called *strings* or *words* over Q, and any set $L \subseteq Q^*$ is called a *language* over Q. Each non-empty word w is of the form $w = q_1 q_2 \cdots q_k$ for some $q_1, q_2, \cdots, q_k \in Q$, $k > 0$. In the following, if a word w is stated as $w = q_1 q_2 \cdots q_k$, it will be assumed that $q_1, q_2, \ldots, q_k \in Q$. The element q_i is called the *ith letter of* w, for $i = 1, 2, \ldots, k$ and k is called the *length* of w. The *empty sequence (word)* is denoted by ε. The length of word w is denoted by $|w|$; specifically, $|\varepsilon| = 0$. The set of non-empty words is denoted by Q^+, i.e., $Q^+ = Q^* \backslash \{\varepsilon\}$. The set of words of length $k \in \mathbb{N}$ is denoted by Q^k. The *concatenation* of word $w \in Q^*$ with $v \in Q^*$ is denoted by wv: if $v = v_1 v_2 \cdots v_k$, and $w = w_1 w_2 \cdots w_m$, $k > 0, m > 0$, then $vw = v_1 v_2 \cdots v_k w_1 w_2 \cdots w_m$. If $v = \varepsilon$, then $wv = w$; if $w = \varepsilon$, then $wv = v$.

If Q has a finite number of elements, say D, it will be identified with its index set, that is $Q = \{1, 2, \cdots, D\}$.

III. Linear switched systems

In this section, we present the formal definition of linear switched systems and recall a number of relevant definitions. We follow the presentation of [14], [16].

Definition 1 (DTLSS): A discrete-time linear switched system (DTLSS) is a tuple

$$\Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) | q \in Q\}, x_0) \tag{3}$$

where $Q = \{1, \cdots, D\}$, $D > 0$, called the set of discrete modes, $A_q \in \mathbb{R}^{n \times n}$, $B_q \in \mathbb{R}^{n \times m}$, $C_q \in \mathbb{R}^{p \times n}$ are the matrices of the linear system in mode $q \in Q$, and x_0 is the initial state. The number n is called the *dimension (order) of* Σ and will sometimes be denoted by $\dim(\Sigma)$.

Notation 1: In the sequel, we use the following notation and terminology: The state space $X = \mathbb{R}^n$, the output space $Y = \mathbb{R}^p$, and the input space $U = \mathbb{R}^m$. We will write $\overline{U^+ \times Q^+} = \{(u, \sigma) \in U^+ \times Q^+ \mid |u| = |\sigma|\}$, and $\sigma(t)$ for the $t+1$th element q_t of a sequence $\sigma = q_1 q_2 \cdots q_{|\sigma|} \in Q^+$ (the same comment applies to the elements of U^+, X^+ and Y^+).

Throughout the paper, Σ denotes a DTLSS of the form (1).

Definition 2 (Solution): A *solution* of the DTLSS Σ at the initial state $x_0 \in X$ and relative to the pair $(u, \sigma) \in \overline{U^+ \times Q^+}$ is a pair $(x, y) \in X^+ \times Y^+$, $|x| = |\sigma| + 1, |y| = |\sigma|$ satisfying

$$\begin{aligned} x(t+1) &= A_{\sigma(t)} x(t) + B_{\sigma(t)} u(t),\ x(0) = x_0 \\ y(t) &= C_{\sigma(t)} x(t), \end{aligned} \tag{4}$$

for $t = 0, 1, \ldots, |\sigma| - 1$.

We shall call u the control input, σ the switching sequence, x the state trajectory, and y the output trajectory. Note that the pair $(u, \sigma) \in \overline{U^+ \times Q^+}$ can be considered as an input to the DTLSS.

Definition 3 (Input-state and input-output maps): The *input-state* map X_{Σ, x_0} and *input-output* map Y_{Σ, x_0} for the DTLSS Σ, induced by the initial state $x_0 \in X$, are the maps

$$\begin{aligned} \overline{U^+ \times Q^+} &\to X^+; (u, \sigma) \mapsto X_{\Sigma, x_0}(u, \sigma) = x, \\ \overline{U^+ \times Q^+} &\to Y^+;\ (u, \sigma) \mapsto Y_{\Sigma, x_0}(u, \sigma) = y, \end{aligned}$$

where (x, y) is the solution of Σ at x_0 relative to (u, σ).

Next, we present the basic system theoretic concepts for DTLSSs . The input-output behavior of a DTLSS realization can be formalized as a map

$$f : \overline{U^+ \times Q^+} \to Y^+. \tag{5}$$

The value $f(u, \sigma)$ represents the output of the underlying (black-box) system. This system may or may not admit a description by a DTLSS. Next, we define when a DTLSS describes (realizes) a map of the form (5).

The DTLSS Σ of the form (1) is a *realization* of an input-output map f of the form (5), if f is the input-output map of Σ which corresponds to some initial state x_0, i.e., $f = Y_{\Sigma, x_0}$. The map Y_{Σ, x_0} will be referred to as the *input-output map of* Σ, and it will be denoted by Y_Σ. The following discussion is valid only for realizable input-output maps.

We say that the DTLSSs Σ_1 and Σ_2 are *equivalent* if $Y_{\Sigma_1} = Y_{\Sigma_2}$. The DTLSS Σ_{m} is said to be a *minimal* realization of f, if Σ_{m} is a realization of f, and for any DTLSS Σ such that Σ is a realization of f, $\dim(\Sigma_{\mathrm{m}}) \leq \dim(\Sigma)$. In [14], it is stated that a DTLSS realization Σ is minimal if and only if it is span-reachable and observable. See [14] for detailed definitions of span-reachability and observability for LSSs.

IV. Model reduction by restricting the set of admissible sequences of discrete modes

In this section, we state formally the problem of restricting the discrete dynamics of the DTLSS .

Definition 4: A non-deterministic finite state automaton (NDFA) is a tuple $\mathscr{A} = (S, Q, \{\to_q\}_{q \in Q}, F, s_0)$ such that

1) S is the finite state set,
2) $F \subseteq S$ is the set of accepting (final) states,
3) $\to_q \subseteq S \times S$ is the state transition relation labelled by $q \in Q$,
4) $s_0 \in S$ is the initial state.

For every $v \in Q^*$, define $\to_v$ inductively as follows: $\to_\varepsilon = \{(s, s) \mid s \in S\}$ and $\to_{vq} = \{(s_1, s_2) \in S \times S \mid \exists s_3 \in S : (s_1, s_3) \in \to_v$ and $(s_3, s_2) \in \to_q\}$ for all $q \in Q$. We denote the fact $(s_1, s_2) \in \to_v$ by $s_1 \to_v s_2$. Define the language $L(\mathscr{A})$ accepted by $\mathscr{A}$ as

$$L(\mathscr{A}) = \{v \in Q^* \mid \exists s \in F : s_0 \to_v s\}.$$

Recall that a language $L \subseteq Q^*$ is regular, if there exists an NDFA $\mathscr{A}$ such that $L = L(\mathscr{A})$. In this case, we say that $\mathscr{A}$ *accepts* or *generates* L. We say that $\mathscr{A}$ is *co-reachable*, if from any state a final state can be reached, i.e., for any $s \in S$,

there exists $v \in Q^*$ and $s_f \in F$ such that $s \to_v s_f$. It is well-known that if $\mathscr{A}$ accepts L, then we can always compute an NDFA $\mathscr{A}_{co-r}$ from $\mathscr{A}$ such that $\mathscr{A}_{co-r}$ accepts L and it is co-reachable. Hence, without loss of generality, in this paper we will consider only co-reachable NDFAs.

Definition 5 (L-realization and L-equivalence): Consider an input-output map f and a DTLSS Σ. Let $L \subseteq Q^+$. We will say that Σ is an L-realization of f, if for every $u \in U^+$, and every $\sigma \in L$ such that $|u| = |\sigma|$,

$$Y_\Sigma(u,\sigma)(|\sigma|-1) = f(u,\sigma)(|\sigma|-1), \tag{6}$$

i.e., the final value of Y_Σ and f agrees for all $(u,\sigma) \in U^+ \times L$, $|\sigma| = |u|$. Note that a Q^+-realization is precisely a realization. We will say that two DTLSS Σ_1 and Σ_2 are L-equivalent, if Σ_2 is an L-realization of Y_{Σ_1} (or equivalently if Σ_1 is an L-realization of Y_{Σ_2}).

Problem 1 (Model reduction preserving L-equivalence): Consider a minimal DTLSS Σ and let $L \subseteq Q^+$ be a regular language. Find a DTLSS $\bar{\Sigma}$ such that $\dim(\bar{\Sigma}) < \dim(\Sigma)$ and, Σ and $\bar{\Sigma}$ are L-equivalent.

V. Model reduction algorithm: preliminaries

In order to present the model reduction algorithm and its proof of correctness, we need to recall the following definitions from [15].

Definition 6 (Convolution representation): The input-output map f has a *generalized convolution representation (abbreviated as **GCR**)*, if there exist maps $S_0^f : Q^+ \to \mathbb{R}^p$, $S^f : Q^+ \to \mathbb{R}^{p\times m}$, such that $S^f(q) = 0$ if $q \in Q$ and

$$f(u,\sigma)(t) = S_0^f(q_0 q_1 \cdots q_t) + \sum_{k=0}^{t-1} S^f(q_k q_{k+1} \cdots q_t) u_k$$

for all $(u,\sigma) \in \overline{U^+ \times Q^+}$, $0 \le t \le |\sigma|-1$ with $\sigma = q_0 q_1 \cdots q_{|\sigma|-1}$.

By a slight abuse of the terminology adopted in [15], we will call the maps $\{S^f, S_0^f\}$ the *Markov parameters* of f. Notice that if f has a **GCR**, then the Markov-parameters of f determine f uniquely. In other words, the Markov-parameters of f and g are equal if and only if f and g are the same input-output map, i.e. $S_0^f = S_0^g$ and $S^f = S^g$ if and only if $f = g$.

In the sequel, we will use the fact that Markov parameters can be expressed via the matrices of a state-space representation. In order to present this relationship, we introduce the following notation.

Notation 2: Let $w = q_1 q_2 \cdots q_k \in Q^*$, $k > 0$ and $A_{q_i} \in \mathbb{R}^{n\times n}$, $i = 1, \cdots, k$. Then the matrix A_w is defined as

$$A_w = A_{q_k} A_{q_{k-1}} \cdots A_{q_1}. \tag{7}$$

If $w = \varepsilon$, then A_ε is the identity matrix.

Lemma 1 ([15]): The map f is realized by the DTLSS Σ if and only if f has a **GCR** and for all $v \in Q^*$, $q, q_0 \in Q$,

$$S^f(q_0 v q) = C_q A_v B_{q_0} \quad \text{and} \quad S_0^f(vq) = C_q A_v x_0. \tag{8}$$

We will extend Lemma 1 to characterize the fact that Σ is an L realization of f in terms of Markov parameters. To this end, we need the following notation.

Notation 3 (Prefix and suffix of L): Let the prefix $(L)_*$ and suffix ${}_*(L)$ of a language L be defined as follows: $(L)_* = \{s \in Q^* \mid \exists w \in Q^* : sw \in L\}$, and ${}_*(L) = \{s \in Q^* \mid \exists w \in Q^* : ws \in L\}$. In addition, let the 1-prefix $(L)_1$ and 1-suffix ${}_1(L)$ of a language L be defined as follows: $(L)_1 = \{s \in Q^* \mid \exists q \in Q : sq \in L\}$ and ${}_1(L) = \{s \in Q^* \mid \exists q \in Q : qs \in L\}$. A language L is said to be prefix (resp. suffix) closed if $(L)_* = L$ (resp. ${}_*(L) = L$).

See [2] for an example which further illustrates this notation. Note that if L is regular, then so are $(L)_*$, ${}_*(L)$, $(L)_1$, ${}_1(L)$. Moreover NDFAs accepting these languages can easily be computed from an NDFA which accepts L.

The proof of Lemma 1 can be extended to prove the following result, which will be central for our further analysis.

Lemma 2: Assume f has a GCR. The DTLSS Σ is an L-realization of f, if and only if for all $v \in Q^*$, $q_0, q \in Q$

$$\begin{aligned} vq \in L &\implies S_0^f(vq) = C_q A_v x_0 \\ q_0 v q \in {}_*(L) &\implies S^f(q_0 v q) = C_q A_v B_{q_0}. \end{aligned} \tag{9}$$

Lemma 1 implies the following important corollary.

Corollary 1: Σ is L-equivalent to $\bar{\Sigma} = (p, m, r, Q, \{(\bar{A}_q, \bar{B}_q, \bar{C}_q) | q \in Q\}, \bar{x}_0)$ if and only if

(i) $\forall q \in Q, v \in Q^* : vq \in L \implies C_q A_v x_0 = \bar{C}_q \bar{A}_v \bar{x}_0$.

(ii) $\forall q, q_0 \in Q, v \in Q^* : q_0 v q \in {}_*(L) \implies C_q A_v B_{q_0} = \bar{C}_q \bar{A}_v \bar{B}_{q_0}$.

That is, in order to find a DTLSS $\bar{\Sigma}$ which is an L-equivalent realization of Σ, it is sufficient to find a DTLSS $\bar{\Sigma}$ which satisfies parts (i) and (ii) of Corollary 1. Intuitively, the conditions (i) and (ii) mean that certain Markov parameters of the input-output maps of $\bar{\Sigma}$ match the corresponding Markov parameters of the input-output map of Σ. Note that L need not be finite, and hence we might have to match an infinite number of Markov parameters. Relying on the intuition of [1], the matching of the Markov parameters can be achieved either by restricting Σ to the set of all states which are reachable along switching sequences from L, or by eliminating those states which are not observable for switching sequences from L. Remarkably, these two approaches are each other's dual.

Below we will formalize this intuition. To this end we use the following notation

Notation 4: In the sequel, the image and kernel of a real matrix M are denoted by $\mathrm{im}(M)$ and $\ker(M)$ respectively. In addition, the $n \times n$ identity matrix is denoted by I_n.

Definition 7 (L-reachability space): For a DTLSS Σ and $L \subseteq Q^*$, define the L-reachability space $\mathscr{R}_L(\Sigma)$ as follows:

$$\begin{aligned} \mathscr{R}_L(\Sigma) = \mathrm{span}\Big\{ &\{A_v x_0 \mid v \in Q^*, v \in ((L)_1)_*\} \cup \\ &\{\mathrm{im}(A_v B_{q_0}) \mid v \in Q^*, q_0 \in Q, q_0 v \in ({}_*((L)_1))_*\} \Big\}. \end{aligned} \tag{10}$$

Whenever Σ is clear from the context, we will denote $\mathscr{R}_L(\Sigma)$ by $\mathscr{R}_L$.

Recall that according to Notation 3

$$\begin{aligned} ((L)_1)_* &= \{s \in Q^* \mid \exists v \in Q^*, \hat{q} \in Q : sv\hat{q} \in L\}, \\ ({}_*((L)_1))_* &= \{s \in Q^* \mid \exists v_1, v_2 \in Q^*, \hat{q} \in Q : v_1 s v_2 \hat{q} \in L\}. \end{aligned} \tag{11}$$

Intuitively, the L-reachability space $\mathscr{R}_L(\Sigma)$ of a DTLSS Σ is the space consisting of all the states $x \in X$ which are reachable from x_0 with some continous input and some switching sequence from L. It follows from [14], [18] that Σ is span-reachable if and only if $\dim(\mathscr{R}_{Q^*}) = n$.

Definition 8 (L-unobservability subspace): For a DTLSS Σ, and $L \subseteq Q^*$, define the L-unobservability subspace as

$$\mathscr{O}_L(\Sigma) = \bigcap_{v \in Q^*, q \in Q, vq \in {}_*(L)} \ker(C_q A_v). \tag{12}$$

If Σ is clear from the context, we will denote $\mathscr{O}_L(\Sigma)$ by $\mathscr{O}_L$. Recall that according to Notation 3,

$${}_*(L) = \{s \in Q^* \mid \exists v \in Q^* : vs \in L\}. \tag{13}$$

Intuitively, the L-unobservability space $\mathscr{O}_L(\Sigma)$ is the set of all those states which remain unobservable under switching sequences from L.

From [18], it follows that Σ is observable if and only if $\mathscr{O}_{Q^*} = \{0\}$. Note that L-unobservability space is not defined in a totally "symmetric" way to the L-reachability space, i.e., subscript of the intersection sign in Equation (12) is not $vq \in {}_*({}_1({}_*(L)))$. See Remarks 2 and 3 in [2] for further discussion.

We are now ready to present two results which are central to the model reduction algorithm to be presented in the next section.

Lemma 3: Let $\Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) | q \in Q\}, x_0)$ be a DTLSS and $L \subseteq Q^*$. Let $\dim(\mathscr{R}_L(\Sigma)) = r$ and $P \in \mathbb{R}^{n \times r}$ be a full column rank matrix such that

$$\mathscr{R}_L(\Sigma) = \mathrm{im}(P).$$

Let $\bar{\Sigma} = (p, m, r, Q, \{(\bar{A}_q, \bar{B}_q, \bar{C}_q) | q \in Q\}, \bar{x}_0)$ be the DTLSS defined by

$$\bar{A}_q = P^{-1} A_q P,\ \bar{B}_q = P^{-1} B_q,\ \bar{C}_q = C_q P,\ \bar{x}_0 = P^{-1} x_0,$$

where P^{-1} is a left inverse of P. Then $\bar{\Sigma}$ and Σ are L-equivalent.

Proof: See [2]. ■

That is, Lemma 3 says that if we find a matrix representation of the L-reachability space, then we can compute a reduced order DTLSS which is an L-realization of Σ. By similar arguments we also obtain:

Lemma 4: Let $\Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) | q \in Q\}, x_0)$ be a DTLSS and let $L \subseteq Q^*$. Let $\mathrm{codim}(\mathscr{O}_L(\Sigma)) = r$ and $W \in \mathbb{R}^{r \times n}$ be a full row rank matrix such that

$$\mathscr{O}_L(\Sigma) = \ker(W).$$

Let $\bar{\Sigma} = (p, m, r, Q, \{(\bar{A}_q, \bar{B}_q, \bar{C}_q) | q \in Q\}, \bar{x}_0)$ be the DTLSS defined by

$$\bar{A}_q = W A_q W^{-1},\ \bar{B}_q = W B_q,\ \bar{C}_q = C_q W^{-1},\ \bar{x}_0 = W x_0.$$

where W^{-1} is a right inverse of W. Then $\bar{\Sigma}$ is an L-equivalent to Σ.

The Lemmas 3 and 4 form the basis of the model reduction algorithm to be presented in the next section.

VI. Model reduction algorithm

In this section, we present an algorithm (Algorithm 3) for solving Problem 1. The proposed algorithm relies on computing the matrices P and W which satisfy the conditions of Lemma 3 and Lemma 4 respectively. In order to compute these matrices, we will formulate alternative definitions of L-reachability/unobservability spaces. To this end, for matrices G, H of suitable dimensions and for a regular language $K \subseteq Q^*$ define the sets $\mathscr{R}_K(G)$ and $\mathscr{O}_K(G)$ as follows;

$$\mathscr{R}_K(G) = \mathrm{span}\{\mathrm{im}(A_v G) \mid v \in K\}$$
$$\mathscr{O}_K(H) = \bigcap_{v \in K} \ker(H A_v).$$

Then the L-reachability space of Σ can be written as

$$\mathscr{R}_L = \mathscr{R}_{((L)_1)_*}(x_0) + \sum_{q \in Q} \mathscr{R}_{({}^q(K))}(B_q), \tag{14}$$

where $((L)_1)_*$ is defined as in (11), and

$${}^q(K) = \{s \in Q^* \mid \exists v_1, v_2 \in Q^*, \hat{q} \in Q, v_1 q s v_2 \hat{q} \in L\}. \tag{15}$$

In (14), $+$ and $\sum$ denote sums of subspaces, i.e. if $\mathscr{W}, \mathscr{V}$ are two linear subspaces of $\mathbb{R}^n$, then $\mathscr{W} + \mathscr{V} = \{a + b \mid a \in \mathscr{W}, b \in \mathscr{W}\}$. Similarly, if $\{\mathscr{W}_i\}_{i \in I}$ is a finite family of linear subspaces of $\mathbb{R}^n$, then $\sum_{i \in I} \mathscr{W}_i = \{\sum_{i \in I} a_i \mid a_i \in \mathscr{W}_i, i \in I\}$.

The L-unobservability space can be written as

$$\mathscr{O}_L = \bigcap_{q \in Q} \mathscr{O}_{((K)^q)}(C_q), \tag{16}$$

where

$$(K)^q = \{s \in Q^* \mid \exists v \in Q^*, vsq \in L\}. \tag{17}$$

Note that if L is regular, then ${}^q(K)$ and $(K)^q$, $q \in Q$, and $((L)_1)_*$ are also regular and NDFA's accepting ${}^q(K)$, $(K)^q$, $q \in Q$, and $((L)_1)_*$ can easily be computed from an NDFA accepting L. From (14) and (16) it follows that in order to compute the matrix P in Lemma 3 or W in Lemma 4, it is enough to compute representations of the subspaces $\mathscr{R}_K(G)$ and $\mathscr{O}_K(H)$ for various choices of K, G and H. The corresponding algorithms are presented in Algorithm 1 and Algorithm 2. There, we used the following notation.

*Notation 5 (**orth**):* For a matrix M, $\mathbf{orth}(M)$ denotes the matrix U whose columns form an orthogonal basis of $\mathrm{im}(M)$.

Lemma 5 (Correctness of Algorithm 1 – Algorithm 2): Algorithm 1 computes $\mathscr{R}_K(G)$ and Algorithm 2 computes $\mathscr{O}_K(H)$.

Proof: See [2]. ■

Notice that the computational complexities of Algorithm 1 and Algorithm 2 are polynomial in n, even though the spaces of $\mathscr{R}_L$ (resp. $\mathscr{O}_L$) might be generated by images (resp. kernels) of exponentially many matrices.

Using Algorithm 1 and 2, we can state Algorithm 3 for solving Problem 1. The matrices P and W computed in Algorithm 3 satisfy the conditions of Lemma 3 and Lemma 4 respectively. Lemma 3 – 4 then imply the following corollary.

Corollary 2 (Correctness of Algorithm 3): The LSS $\bar{\Sigma}$ returned by Algorithm 3 is a solution of Problem 1, i.e. $\bar{\Sigma}$ is L-equivalent to Σ and $\dim(\bar{\Sigma}) \leq \dim(\Sigma)$.

Algorithm 1 Calculate a matrix representation of $\mathscr{R}_K(G)$,
Inputs: $(\{A_q\}_{q\in Q}, G)$ and $\hat{\mathscr{A}} = (S, \{\rightarrow_q\}_{q\in Q}, F, s_0)$ such that $L(\hat{\mathscr{A}}) = K$, $F = \{s_{f_1}, \cdots s_{f_k}\}$, $k \geq 1$ and $\hat{\mathscr{A}}$ is co-reachable.
Outputs: $\hat{P} \in \mathbb{R}^{n\times \hat{r}}$ such that $\hat{P}^{\mathrm{T}}\hat{P} = I_{\hat{r}}$, $\mathrm{rank}(\hat{P}) = \hat{r}$, $\mathrm{im}(\hat{P}) = \mathscr{R}_K(G)$.

$\forall s \in S\backslash\{s_0\} : P_s := 0.$
$P_{s_0} := \mathbf{orth}(G).$
$\mathrm{flag} = 0.$
while $\mathrm{flag} = 0$ **do**
  $\forall s \in S : P_s^{old} := P_s$
  for $s \in S$ **do**
    $W_s := P_s$
    for $q \in Q, s' \in S : s' \rightarrow_q s$ **do**
      $W_s := \begin{bmatrix} W_s, & A_q P_{s'}^{old} \end{bmatrix}$
    end for
    $P_s := \mathbf{orth}(W_s)$
  end for
  if $\forall s \in S : \mathrm{rank}(P_s) = \mathrm{rank}(P_s^{old})$ **then**
    $\mathrm{flag} = 1.$
  end if
end while
return $\hat{P} = \mathbf{orth}\left(\begin{bmatrix} P_{s_{f_1}} & \cdots & P_{s_{f_k}} \end{bmatrix}\right).$

Algorithm 2 Calculate a matrix representation of $\mathscr{O}_K(H)$,
Inputs: $(\{A_q\}_{q\in Q}, H)$ and $\hat{\mathscr{A}} = (S, \{\rightarrow_q\}_{q\in Q}, F, s_0)$ such that $L(\hat{\mathscr{A}}) = K$, $F = \{s_{f_1}, \cdots s_{f_k}\}$, $k \geq 1$ and $\hat{\mathscr{A}}$ is co-reachable.
Outputs: $\hat{W} \in \mathbb{R}^{\hat{r}\times n}$ such that $\hat{W}\hat{W}^{\mathrm{T}} = I_{\hat{r}}$, $\mathrm{rank}(\hat{W}) = \hat{r}$, $\ker(\hat{W}) = \mathscr{O}_K(H)$.

$\forall s \in S\backslash F : W_s := 0.$
$\forall s \in F : W_s^{\mathrm{T}} := \mathbf{orth}(H^{\mathrm{T}}).$
$\mathrm{flag} = 0.$
while $\mathrm{flag} = 0$ **do**
  $\forall s \in S : W_s^{old} := W_s$
  for $s \in S$ **do**
    $P_s := W_s$
    for $q \in Q, s' \in S : s \rightarrow_q s'$ **do**
      $P_s := \begin{bmatrix} P_s \\ W_{s'}^{old} A_q \end{bmatrix}$
    end for
    $W_s^{\mathrm{T}} := \mathbf{orth}(P_s^{\mathrm{T}})$
  end for
  if $\forall s \in S : \mathrm{rank}(W_s) = \mathrm{rank}(W_s^{old})$ **then**
    $\mathrm{flag} = 1.$
  end if
end while
return $\hat{W} = W_{s_0}.$

VII. NUMERICAL EXAMPLE

In this section, the model reduction method for DTLSSs with restricted discrete dynamics will be illustrated by a numerical example. The data used for the example and corresponding MATLAB code are available from https://kom.aau.dk/~mertb/. The example specifically illustrates the use of Lemma 4 and Algorithm 3. For an example where Lemma 3 is used, see [2]. The NDFA which generates the set of admissible switching sequences is defined as the tuple $\mathscr{A} = (S, \{\rightarrow_q\}_{q\in Q}, F, s_0)$ where $S = \{s_0, s_1, s_f\}$, $\rightarrow_1 = \{(s_0, s_1)\}$, $\rightarrow_2 = \{(s_1, s_f)\}$, $\rightarrow_3 = \{(s_f, s_0)\}$ and $F = \{s_f\}$.

Algorithm 3 Reduction for DTLSSs
Inputs: $\Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) | q \in Q\}, x_0)$ and $\mathscr{A} = (S, \{\rightarrow_q\}_{q\in Q}, F, s_0)$ such that $L(\mathscr{A}) = L$, $F = \{s_{f_1}, \cdots s_{f_k}\}$, $k \geq 1$ and $\mathscr{A}$ is co-reachable.
Output: $\bar{\Sigma} = (p, m, r, Q, \{(\bar{A}_q, \bar{B}_q, \bar{C}_q) | q \in Q\}, \bar{x}_0)$.

Compute a co-reachable NDFA $\hat{\mathscr{A}}_r$ from $\mathscr{A}$ such that $L(\hat{\mathscr{A}}_r) = ((L)_1)_*$, where $((L)_1)_*$ is as in (11).
Use Algorithm 1 with inputs $(\{A_q\}_{q\in Q}, x_0)$ and NDFA $\hat{\mathscr{A}}_r$. Store the output $\hat{P}$ as $P_{x_0} := \hat{P}$.
for $q \in Q$ **do**
  Compute a co-reachable NDFA $\hat{\mathscr{A}}_{r,q}$ from $\mathscr{A}$ such that $L(\hat{\mathscr{A}}_{r,q}) = {}^q(K)$, where ${}^q(K)$ is as in (15).
  Use Algorithm 1 with inputs $(\{A_q\}_{q\in Q}, B_q)$ and NDFA $\hat{\mathscr{A}}_{r,q}$ Store the output $\hat{P}$ as $P_q := \hat{P}$.
end for
$P = \mathbf{orth}(\begin{bmatrix} P_{x_0} & P_1 & \cdots & P_D \end{bmatrix})$
for $q \in Q$ **do**
  Compute a co-reachable NDFA $\hat{\mathscr{A}}_{o,q}$ from $\mathscr{A}$, such that $L(\hat{\mathscr{A}}_{o,q}) = (K)^q$, where $(K)^q$ is as in (17).
  Use Algorithm 2 with inputs $(\{A_q\}_{q\in Q}, C_q)$ and NDFA $\hat{\mathscr{A}}_{o,q}$. Store the output $\hat{W}$ as $W_q := \hat{W}$.
end for
$W^{\mathrm{T}} = \mathbf{orth}(\begin{bmatrix} W_1^{\mathrm{T}} & \cdots & W_D^{\mathrm{T}} \end{bmatrix})$
if $\mathrm{rank}(P) < \mathrm{rank}(W)$ **then**
  Let $r = \mathrm{rank}(P)$, P^{-1} be a left inverse of P and set
$$\bar{A}_q = P^{-1}A_qP,\ \bar{C}_q = C_qP,\ \bar{B}_q = P^{-1}B_q,\ \bar{x}_0 = P^{-1}x_0.$$
end if
if $\mathrm{rank}(P) \geq \mathrm{rank}(W)$ **then**
  Let $r = \mathrm{rank}(W)$ and let W^{-1} be a right inverse of W. Set
$$\bar{A}_q = WA_qW^{-1},\ \bar{C}_q = C_qW^{-1},\ \bar{B}_q = WB_q,\ \bar{x}_0 = Wx_0.$$
end if
return $\bar{\Sigma} = (p, m, r, Q, \{(\bar{A}_q, \bar{B}_q, \bar{C}_q) | q \in Q\}, \bar{x}_0)$.

Observe that the language L accepted by the NDFA $\mathscr{A}$ is the set $L = \{12, 12312, \ldots\}$ and it can also be represented by the regular expression $L = (123)^*12$. As stated in Definition 4, the labels of the edges represent the discrete mode indices of the DTLSS. The model reduction procedure is applied to the single input - single output (SISO) DTLSS Σ of order $n = 7$ with $Q = \{1, 2, 3\}$ and

```
A1 = [[0, 1, 0, 0, 0, 0, 0]; randn(6,7)]
A2 = [randn(2,7); [0, 0, 0, 0, 0, 0, 0]; randn(4,7)]
A3 = [randn(1,7); [0, 0, 1, 0, 0, 0, 0]; randn(5,7)]
B1 = randn(7,1), B2 = randn(7,1), B3 = randn(7,1)
C1 = randn(1,7), C2 = [1, 0, 0, 0, 0, 0, 0], C3 = randn(1,7).
```

Applying Algorithm 3 to this DTLSS whose admissible switching sequences are generated by this NDFA, yields a reduced order system $\bar{\Sigma}$ of order $r = 3$, whose output values are the same as the original system Σ for the instances when the NDFA reaches a final state. Note that this corresponds precisely to an L-realization in the sense of Definition 5 (the last outputs of Σ and $\bar{\Sigma}$ are the same for all the switching sequences generated by the governing NDFA, i.e., for all

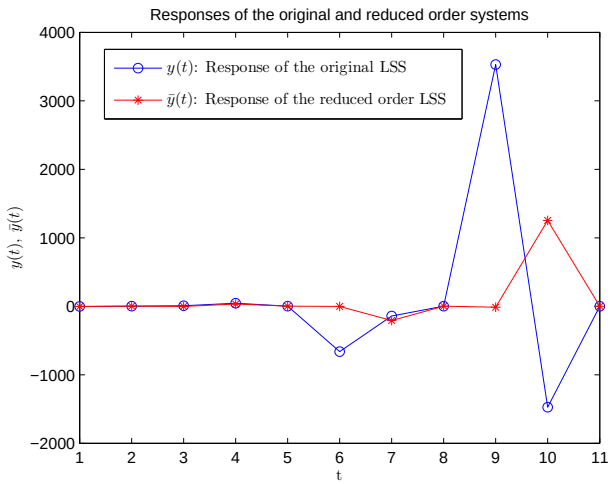

Fig. 1. The responses of the original DTLSS Σ of order 7 and the reduced order DTLSS $\bar{\Sigma}$ of order 3 acquired by Algorithm 3 for the switching sequence in (18). Note that only the second and fifth element of the output sequence is equal for Σ and $\bar{\Sigma}$ in first five elements, others look the same as a result of the scaling.

$\sigma \in L(\mathscr{A})$). In this example, the algorithm makes use of Lemma 4 and constructs the W matrix. The matrix $W \in \mathbb{R}^{r\times n}$ computed is $W = \left[\begin{array}{c|c} I_3 & \mathbf{0} \end{array}\right]$. The outputs of Σ and $\bar{\Sigma}$ for the switching sequence (18) of length 11 generated by the NDFA $\mathscr{A}$ are given in Figure 1 for comparison.

$$\sigma = 12312312312. \tag{18}$$

Note that the resulting DTLSS is merely an L-realization of Σ and nothing more, i.e., its output coincides with the output of Σ for only the instances corresponding to the final states of the NDFA, see Remark 3 in [2] for further discussion. This fact is visible from Figure 1, where it can be seen that output corresponding to the final state s_f of the NDFA coincides for Σ and $\bar{\Sigma}$ (Observe that for all switching sequences generated by $\mathscr{A}$ ending with the label 2, the output values of Σ and $\bar{\Sigma}$ are the same). The input sequence of length 11 used in the simulation is generated by the function `randn`. Finally, note that the DTLSS Σ is minimal [2] (note that the definition of minimality for linear switched systems are made by considering *all* possible switching sequences in Q^* [15]), whereas for the switching sequences restricted by the NDFA $\mathscr{A}$, it turns out 4 states are disposable in this case. In fact, this is the main idea of the paper.

VIII. CONCLUSIONS

A model reduction method for discrete time linear switched systems whose discrete dynamics are restricted by switching sequences comprising a regular language is presented. The method is essentially a moment matching type of model reduction method, which focuses on matching the Markov parameters of a DTLSS related to the specific switching sequences generated by a nondeterministic finite state automaton. Possible future research directions include expanding the method for continuous time case, and approximating the input/output behavior of the original system rather than exactly matching it, and formulating the presented algorithms in terms of bisimulation instead of input-output equivalence.

REFERENCES

[1] M. Bastug, M. Petreczky, R. Wisniewski, and J. Leth. Model reduction by moment matching for linear switched systems. *Proceedings of the American Control Conference*, 2014.

[2] M. Bastug, M. Petreczky, R. Wisniewski, and J. Leth. Model reduction of linear switched systems by restricting discrete dynamics. Technical Report arXiv:1409.1444, ArXiv, 2014. Available at http://arxiv.org/abs/1409.1444.

[3] A. Birouche, J Guilet, B. Mourillon, and M Basset. Gramian based approach to model order-reduction for discrete-time switched linear systems. In *Proc. Mediterranean Conference on Control and Automation*, 2010.

[4] Y. Chahlaoui. Model reduction of hybrid switched systems. In *Proceeding of the 4th Conference on Trends in Applied Mathematics in Tunisia, Algeria and Morocco, May 4-8, Kenitra, Morocco*, 2009.

[5] Mircea Lazar Calin Belta Ebru Aydin Gol, Xu Chu Ding. Finite bisimulations for switched linear systems. In *IEEE Conference on Decision and Control (CDC) 2012, Maui, Hawaii*, 2012.

[6] Goran Frehse. Phaver: algorithmic verification of hybrid systems past hytech. *International Journal on Software Tools for Technology Transfer*, 10(3):263–279, 2008.

[7] H. Gao, J. Lam, and C. Wang. Model simplification for switched hybrid systems. *Systems & Control Letters*, 55:1015–1021, 2006.

[8] C.G.J.M. Habets and J. H. van Schuppen. Reduction of affine systems on polytopes. In *International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[9] G. Kotsalis, A. Megretski, and M. A. Dahleh. Balanced truncation for a class of stochastic jump linear systems and model reduction of hidden Markov models. *IEEE Transactions on Automatic Control*, 53(11), 2008.

[10] G. Kotsalis and A. Rantzer. Balanced truncation for discrete-time Markov jump linear systems. *IEEE Transactions on Automatic Control*, 55(11), 2010.

[11] D. Liberzon. *Switching in Systems and Control*. Birkhäuser, Boston, MA, 2003.

[12] E. Mazzi, A.S. Vincentelli, A. Balluchi, and A. Bicchi. Hybrid system model reduction. In *IEEE International conference on Decision and Control*, 2008.

[13] N. Monshizadeh, H. Trentelman, and M. Camlibel. A simultaneous balanced truncation approach to model reduction of switched linear systems. *Automatic Control, IEEE Transactions on*, PP(99):1, 2012.

[14] M. Petreczky. Realization theory for linear and bilinear switched systems: formal power series approach - part i: realization theory of linear switched systems. *ESAIM Control, Optimization and Caluculus of Variations*, 17:410–445, 2011.

[15] M. Petreczky, L. Bako, and J. H. van Schuppen. Realization theory of discrete-time linear switched systems. *Automatica*, 49:3337–3344, November 2013.

[16] M. Petreczky, R. Wisniewski, and J. Leth. Balanced truncation for linear switched systems. *Nonlinear Analysis: Hybrid Systems*, 10:4–20, November 2013.

[17] H.R. Shaker and R. Wisniewski. Generalized gramian framework for model/controller order reduction of switched systems. *International Journal of Systems Science, in press*, 2011.

[18] Z. Sun and S. S. Ge. *Switched linear systems : control and design*. Springer, London, 2005.

[19] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer-Verlag, 2009.

[20] B. Yordanov and C. Belta. Formal analysis of discrete-time piecewise affine systems. *Automatic Control, IEEE Transactions on*, 55(12):2834–2840, Dec 2010.

[21] L. Zhang, E. Boukas, and P. Shi. Mu-dependent model reduction for uncertain discrete-time switched linear systems with average dwell time. *International Journal of Control*, 82(2):378– 388, 2009.

[22] L. Zhang and P. Shi. Model reduction for switched lpv systems with average dwell time. *IEEE Transactions on Automatic Control*, 53:2443–2448, 2008.

[23] L. Zhang, P. Shi, E. Boukas, and C. Wang. Model reduction for uncertain switched linear discrete-time systems. *Automatica*, 44(11):2944 – 2949, 2008.