Model Reduction of Linear Switched Systems by Restricting Discrete Dynamics

Mert Baştuğ¹,², Mihály Petreczky², Rafael Wisniewski¹ and John Leth¹

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], [19] (abbreviated by DTLSS) is a discrete-time hybrid system of the form

\[
\begin{aligned}
\sum x(t+1) &= A_{\sigma(t)} x(t) + B_{\sigma(t)} u(t) \quad \text{and} \quad x(0) = x_0 \\
y(t) &= C_{\sigma(t)} x(t),
\end{aligned}
\]

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, \( \sigma(t) \in \{1, \ldots, D\} \) 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 \( \Sigma \) 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

\[
\begin{aligned}
\sum \bar{x}(t+1) &= \bar{A}_{\sigma(t)} \bar{x}(t) + \bar{B}_{\sigma(t)} u(t) \quad \text{and} \quad \bar{x}(0) = \bar{x}_0 \\
\bar{y}(t) &= \bar{C}_{\sigma(t)} \bar{x}(t)
\end{aligned}
\]

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 behavior of \( \Sigma \) 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 [20]. 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. To illustrate this point, we will consider a number of simple scenarios where the results of the paper could potentially be useful.

(1) Control and verification of DTLSSs with switching constraints. 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 \( \Sigma \) by the reduced order DTLSS \( \bar{\Sigma} \) whose input-output behavior for all the admissible switching sequences coincides with that of \( \Sigma \). We can then perform verification or control synthesis for \( \bar{\Sigma} \) instead of \( \Sigma \). If \( \Sigma \) satisfies the desired input-output properties, then so does \( \bar{\Sigma} \). Likewise, if the composition of \( \bar{\Sigma} \) with a controller meets the control objectives, then the composition of this controller with \( \Sigma \) meets them too.

(2) Piecewise-affine hybrid systems. Consider a piecewise-linear hybrid system \( H \) [2], [21]. Such systems can often be modelled as a feedback interconnection of a linear switched system \( \Sigma \) of the form (1) with a discrete event generator \( \phi \), which generates the next discrete state based on the past discrete states and past outputs. As a consequence, the solutions of \( H \) corresponds to the solutions \( \{q_i, x_i, y_i\}_{i=0}^\infty \) of (1) with \( q_0 = \phi(\{y_0, q_0\}_{i=0}^{t-1}) \). A simple example of such a system is \( q_i = \phi(y_{i-1}), \ t > 0, \) and \( q_0 \) is fixed, where \( \phi \) is a piecewise affine map. Often, it is desired to verify if the system is safe, i.e., that the sequences of discrete modes generated by the system \( H \) belong to a certain set of safe sequences \( L \) for all (some) continuous input signals. Consider now another piecewise-affine hybrid system \( \bar{H} \) obtained by interconnecting the discrete event generator \( \phi \) with a reduced order DTLSS \( \bar{\Sigma} \), such that the input-output behavior of \( \bar{\Sigma} \) coincides with that of \( \Sigma \) for all the switching sequences from \( L \). If \( L \) is prefix closed, then \( H \) is safe if and only if \( \bar{H} \) is safe, and hence it is sufficient to perform safety analysis on \( \bar{H} \). Since the number of continuous states of \( \bar{H} \) is smaller than that of \( H \), it is easier to do verification for \( \bar{H} \) than for the original model. Note that verification of

1Department of Electronic Systems, Automation and Control, Aalborg University, 9220 Aalborg, Denmark mertb@es.aau.dk, raf@es.aau.dk, jjl@es.aau.dk
2Department of Computer Science and Automatic Control (UR Informatique et Automatique), Ecole des Mines de Douai, 59508 Douai, France
mihaly.petreczky@mines-douai.fr
piecewise-affine hybrid systems has high (in certain cases exponential) computational complexity, [6], [22]. Likewise, assume that it is desired to design a control law for $H$ 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 [20].

Again, this problem is solvable for $H$ if and only if it is solvable for $\tilde{H}$, and the controller which solves this problem for $\tilde{H}$ also solves it for $H$.

**Related work** Results on realization theory of linear switched systems with constrained switching appeared in [15]. However, [15] does not yield a model reduction algorithm, see Remark 11 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], [25], [12], [4], [8], [23], [24], [7], [9], [10], [13], [18]. 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 concept 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 some numerical examples.

**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 $w = q_1q_2 \cdots q_k$ for some $q_1, q_2, \ldots, q_k \in Q$, $k > 0$. In the following, if a word $w$ is stated as $w = q_1q_2 \cdots q_k$, it will be assumed that $q_1, q_2, \ldots, q_k \in Q$. The element $q_i$ is called the $i$th letter of $w$, for $i = 1, 2, \ldots, k$ and $k$ is called the length of $w$.

The **empty sequence** (word) is denoted by $\varepsilon$. The length of word $w$ is denoted by $|w|$; note that $|\varepsilon| = 0$. The set of non-empty words is denoted by $Q^+$, i.e., $Q^+ = Q^+ \setminus \{\varepsilon\}$. The set of words of length $k \in \mathbb{N}$ is denoted by $Q^k$. The concatenation of word $w \in Q^k$ with $v \in Q^k$ is denoted by $wv$: if $v = v_1v_2 \cdots v_m$ and $w = w_1w_2 \cdots w_m$, $k > 0, m > 0$, then $wv = v_1v_2 \cdots v_mw_1w_2 \cdots w_m$. If $v = \varepsilon$, then $wv = w$; if $w = \varepsilon$, then $vw = v$.

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

**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 [15], [17].

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

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

(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 $\Sigma$ 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 $U^+ \times Q^+ = \{(u, \sigma) \in U^+ \times Q^+ | |u| = |\sigma|\}$, and $(t)$ for the $t$th element of a sequence $q = q_1q_2 \cdots q_{|\sigma|} \in Q^+$ (the same comment applies to the elements of $U^+, X^+$ and $Y^+$).

Throughout the paper, $\Sigma$ denotes a DTLSS of the form (1).

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

$$x(t + 1) = A_{\sigma(t)}(t)x(t) + B_{\sigma(t)}(t)u(t), x(0) = x_0$$

(4)

$$y(t) = C_{\sigma(t)}(t)x(t),$$

for $t = 0, 1, \ldots, |\sigma| - 1$. We shall call $u$ the control input, $\sigma$ the switching sequence, $x$ the state trajectory, and $y$ the output trajectory. Note that the pair $(u, \sigma) \in U^+ \times Q^+$ can be considered as an input to the DTLSS.

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

$$U^+ \times Q^+ \rightarrow X^+: (u, \sigma) \mapsto X_{\Sigma, x_0}(u, \sigma) = x,$$

(5)

$$U^+ \times Q^+ \rightarrow Y^+: (u, \sigma) \mapsto Y_{\Sigma, x_0}(u, \sigma) = y,$$

where $(x, y)$ is the solution of $\Sigma$ at $x_0$ relative to $(u, \sigma)$.

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

$$f : U^+ \times Q^+ \rightarrow Y^.$$
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 $(S, Q, \{\rightarrow_q\}_{q \in Q}, F, s_0)$ such that:
1. $S$ is the finite state set,
2. $F \subseteq S$ is the set of accepting (final) states,
3. $\rightarrow_q \subseteq S \times S$ 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 $\rightarrow_v$ inductively as follows: $\rightarrow_e = \{(s, s) \mid s \in S\}$ and $\rightarrow_q = \{(s_1, s_2) \in S \times S \mid \exists s_3 \in S : (s_1, s_3) \rightarrow_v (s_3, s_2) \rightarrow_q\}$ for all $q \in Q$. We denote the fact $(s_1, s_2) \rightarrow_v$ by $s_1 \rightarrow_v s_2$. The fact that there exists $s_2$ such that $s_1 \rightarrow_v s_2$ is denoted by $s_1 \rightarrow_v$. Define the language $L(\mathcal{A})$ accepted by $\mathcal{A}$ as

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

Recall that a language $L \subseteq Q^*$ is regular, if there exists an NDFA $\mathcal{A}$ such that $L = L(\mathcal{A})$. In this case, we say that $\mathcal{A}$ accepts or generates $L$. We say that $\mathcal{A}$ 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 $s \rightarrow_v s_f$. It is well-known that if $\mathcal{A}$ accepts $L$, then we can always compute an NDFA $\mathcal{A}_{co-r}$ from $\mathcal{A}$ such that $\mathcal{A}_{co-r}$ accepts $L$ and it is co-reachable. Hence, without loss of generality, in this paper we will consider only co-reachable NFAs.

Definition 5 (L-realization and L-equivalence): Consider an input-output map $f$ and a DTLSS $\Sigma$. Let $L \subseteq Q^+$. We will say that $\Sigma$ is an $L$-realization of $f$, if for every $u \in U^+$, and every $\sigma \in L$ such that $|u| = |\sigma|$, $Y_\Sigma(u, \sigma)(|\sigma| - 1) = f(u, \sigma)(|\sigma| - 1), \quad (6)$

i.e., the final value of $Y_\Sigma$ 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 $\Sigma_1$ and $\Sigma_2$ are L-equivalent, if $\Sigma_2$ is an $L$-realization of $Y_{\Sigma_1}$ (or equivalently if $\Sigma_1$ is an $L$-realization of $Y_{\Sigma_2}$).

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

Remark 1: The problem of finding an $L$-realization of $f$ was already addressed in [15], [14] for the continuous time case. There, it was shown that if $\Sigma$ is a realization of $f$ and $M$ is a number which depends on the cardinality of the state-space of a deterministic finite state automaton accepting $L$, then it is possible to find a $\Sigma'$ such that $\Sigma'$ is an $L$-realization of $f$ and

$$\dim \Sigma' \leq M \dim \Sigma. \quad (7)$$

This result may also be extended for the discrete time case in a similar way. However, as [7] shows, the obtained $L$-realization can even be of higher dimension than the original system.

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 [16].

Definition 6 (Convolution representation): The input-output map $f$ has a generalized convolution representation (abbreviated as GCR), if there exist maps $S'_0 : Q^+ \rightarrow \mathbb{R}^p$, $S_f : Q^+ \rightarrow \mathbb{R}^{p \times m}$, such that $S_f(0) = 0$ if $q \in Q$ and $f(u, \sigma)(t) = S'_0(q_0q_1 \cdots q_t) + \sum_{k=0}^{t-1} S_f(q_kq_{k+1} \cdots q_t)u_k$

for all $(u, \sigma) \in U^+ \times Q^+$, $t \leq |\sigma|$ with $\sigma = q_0q_1 \cdots q_{|\sigma|}$. By a slight abuse of the terminology adopted in [16], we will call the maps $(S'_0, S_f)$ 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. $S'_0 = S'_0$ and $S_f = S_f$ 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 $w = q_0q_2 \cdots q_k \in Q^+$, $k > 0$ and $A_{q_i} \in \mathbb{R}^{n \times n}$, $i = 1, \cdots, k$. Then the matrix $A_w$ is defined as

$$A_w = A_{q_k}A_{q_{k-1}} \cdots A_{q_1}. \quad (8)$$

If $w = e$, then $A_e$ is the identity matrix.

Lemma 1 ([16]): The map $f$ is realized by the DTLSS $\Sigma$ if and only if $f$ has a GCR and for all $v \in Q^+$, $q, q_0 \in Q$,

$$S'_0(qq_0vq) = C_qA_vB_{q_0} \quad \text{and} \quad S'_0(vq) = C_qA_vx_0. \quad (9)$$

We will extend Lemma 11 to characterize the fact that $\Sigma$ 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 $(L)_1$ and suffix $(L)_s$ of a language $L$ be defined as follows: $(L)_1 = \{s \in Q^* \mid \exists w \in Q^* : sw \in L\}$, and $(L)_s = \{s \in Q^* \mid \exists w \in Q^* : sw \in L\}$. In addition, let the $1$-prefix $(L)_1$ and $1$-suffix $(L)_s$ of a language $L$ be defined as follows: $(L)_1 = \{s \in Q^* \mid \exists q \in Q : sq \in L\}$, and $(L)_s = \{s \in Q^* \mid \exists q \in Q : qs \in L\}$. A language $L$ is said to be prefix (resp. suffix) closed if $(L)_1 = L$ (resp. $(L)_s = L$).

Example 1: Let the language $L$ be defined as $L = (123)^{\dagger}12 = \{12, 12312, 12312312, \ldots\}$. Then with the notation stated above, the following languages can be defined as follows:

$$(L)_1 = \{\varepsilon, 2, 12, 312, 2312, 12312, \ldots\}$$

$$(L)_s = \{\varepsilon, 1, 312, 1231, \ldots\}$$

$$((L)_1)_s = \{\varepsilon, 1, 2, 3, 12, 23, 31, 123, 231, 312, \ldots\}$$

$$((L)_s)_1 = \{1, 1231, 1231231, \ldots\}$$

Note that if $L$ is regular, then so are $(L)_s$. $(L)_1$, $(L), (L)_s$. Moreover NDFAs accepting these languages can easily be computed from an N DFA 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 DTLLS $\Sigma$ is an $L$-realization of $f$, if and only if for all $v \in Q^*$, $q_0, q \in Q$

\[
q_0vq \in \mathcal{L} \iff S^f(q_0vq) = C_qA_vx_0
\] (10)

**Lemma 1** implies the following important corollary.

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

(i) $\forall q \in Q, v \in Q^* : q_0v \in \mathcal{L} \implies C_qA_vx_0 = C_q\tilde{A}_\tilde{v}\tilde{x}_0$.

(ii) $\forall q, q_0 \in Q, v \in Q^* : q_0vq \in \mathcal{L} \implies C_qA_v B_0 = C_q\tilde{A}_\tilde{v} \tilde{B}_0$.

That is, in order to find a DTLLS $\tilde{\Sigma}$ which is an $L$-equivalent realization of $\Sigma$, it is sufficient to find a DTLLS $\tilde{\Sigma}$ 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 $\Sigma$ match the corresponding Markov parameters of the input-output map of $\tilde{\Sigma}$. 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 $\Sigma$ 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. This this end we use the following notation

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

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

\[
\mathcal{R}_L(\Sigma) = \text{span}\left\{ (A_{q_0}x_0) \mid v \in Q^*, v \in ((L)_1), \right\} \\
\cup \left\{ \text{im}(A_{q\cdot}B_{q_0}) \mid v \in Q^*, q_0v \in \mathcal{L} \right\}.
\] (11)

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

Recall that according to Notation 3,

\[
((L)_1)_s = \{ s \in Q^* \mid \exists v \in Q^*, \tilde{q} \in Q : sv\tilde{q} \in L \}, \\
(s((L)_1))_s = \{ s \in Q^* \mid \exists v_1, v_2 \in Q^*, \tilde{q} \in Q : v_1sv_2v\tilde{q} \in L \}.
\] (12)

Intuitively, the $L$-reachability space $\mathcal{R}_L(\Sigma)$ of a DTLLS $\Sigma$ is the space consisting of all the states $x \in X$ which are reachable from $x_0$ with some continuous input and some switching sequence from $L$. It follows from [15], [19] that $\Sigma$ is span-reachable if and only if $\dim \mathcal{R}_Q \geq n$.

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

\[
\mathcal{O}_L(\Sigma) = \bigcap_{v \in Q^*, q \in \mathcal{L}} \text{ker}(C_qA_v).
\] (13)

If $\Sigma$ is clear from the context, we will denote $\mathcal{O}_L(\Sigma)$ by $\mathcal{O}_L$.

Recall that according to Notation 3

\[
\mathcal{O}_L(\Sigma) = \{ s \in Q^* \mid \exists v \in Q^* : vs \in L \}.
\] (14)

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

From [19], it follows that $\Sigma$ is observable if and only if $\mathcal{O}_Q = \{0\}$. 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 (13) is not $vq \in s((L)_1))$. See Remarks 2 and 3 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 DTLLS and $L \subseteq Q^*$. Let $\dim \mathcal{R}_L(\Sigma) = r$ and $P \in \mathbb{R}^{r \times r}$ be a full column rank matrix such that

\[
\mathcal{R}_L(\Sigma) = \text{im}(P).
\]

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

\[
\tilde{A}_q = P^{-1}A_qP, \tilde{B}_q = P^{-1}B_q, \tilde{C}_q = C_qP, \tilde{x}_0 = P^{-1}x_0,
\]

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

That is, Lemma 3 says that if we find a matrix representation of the $L$-reachability space, then we can compute a reduced order DTLLS which is an $L$-realization of $\Sigma$.

Before presenting the proof of Lemma 3 we will prove the following claim.

**Claim 1:** With the conditions of Lemma 3 the following holds:

(i) For all $v \in Q^*$ such that $v \in ((L)_1)_s$,

\[
v = \varepsilon \implies PP^{-1}x_0 = x_0,
\]

$v = q_1 \cdots q_k, k \geq 1 \implies PP^{-1}A_{q_k} \cdots PP^{-1}A_{q_1}PP^{-1}x_0 = A_{q_k} \cdots A_{q_1}x_0$.

(ii) For all $v \in Q^*, q_0v \in Q$ such that $q_0v \in ((s(L)_1))_s$,

\[
v = \varepsilon \implies PP^{-1}B_{q_0} = B_{q_0},
\]

$v = q_1 \cdots q_k, k \geq 1 \implies PP^{-1}A_{q_k} \cdots PP^{-1}A_{q_1}PP^{-1}B_{q_0} = A_{q_k} \cdots A_{q_1}B_{q_0}$.

**Proof:** (Claim 1 (ii)) The proof is by induction on the length of $v$. For $|v| = 0$, let $q_0 \in Q$ and $q_0v \in ((s(L)_1))_s$. The assumption $\mathcal{R}_L(\Sigma) = \text{im}(P)$ in Lemma 3 implies $\text{im}(B_{q_0}) \subseteq \text{im}(P)$.

Hence, there exists an $\Lambda \in \mathbb{R}^{r \times m}$ such that $PA = B_{q_0}$, and therefore $PP^{-1}B_{q_0} = PP^{-1}PA = PA = B_{q_0}$.

For $|v| = k \geq 1$, let $v = q_1 \cdots q_k, q_0v \in ((s(L)_1))_s$, observe that if $q_0v \in ((s(L)_1))_s$ then also $q_0\tilde{v} \in ((s(L)_1)_s)$, where $\tilde{v} = q_1 \cdots q_{k-1}$, since the set $((s(L)_1)_s)$ is prefix closed. Assume the claim holds for $|v| = k - 1$, i.e., for $v = q_1 \cdots q_{k-1}$. Then

\[
PP^{-1}A_{q_k} PP^{-1}A_{q_{k-1}} \cdots PP^{-1}A_{q_1} PP^{-1}B_{q_0} = PP^{-1}A_{q_k} A_{q_{k-1}} \cdots A_{q_1} B_{q_0}.
\]
Since again \( \text{im}(A_q \cdots A_q B_{q_0}) \subseteq \text{im}(P) \), it follows that
\[
PP^{-1}A_q^k A_{q_{k-1}} \cdots A_{q_1} B_{q_0} = A_q^k A_{q_{k-1}} \cdots A_{q_1} B_{q_0},
\]
proving this part. The proof of part (i) is similar.

Proof: [Proof of Lemma 3] We will show that part (i) and (ii) of Corollary 1 hold.

(iii) Using Claim 1(ii) and observing \( s(L) \subseteq \langle s(L) \rangle \), it follows that, for all \( v \in Q^*, q_0, q \in Q \) such that \( q_0 v q \in s(L) \)
\[
v = e \implies C_q B_{q_0} = C_q PP^{-1} B_{q_0} = C_q B_{q_0},
\]
\[
v = q_1 \cdots q_k, k \geq 1 \implies \bar{C}_q B_{q_0} = C_q PP^{-1} A_q \cdots PP^{-1} A_{q_1} PP^{-1} B_{q_0} = C_q A_q^k A_{q_{k-1}} \cdots A_{q_1} B_{q_0},
\]
(i) Similar to part (ii).

By similar arguments we also obtain:

**Lemma 4:** Let \(\Sigma = (p, m, n, Q, \{ (A_q, B_q, C_q) \} Q \subseteq Q, \bar{x}_0) \) be a DTLSS and let \( \Sigma \subseteq \Sigma \) be a full row rank matrix such that
\[
\Theta_{L}(\Sigma) = \ker(W).
\]

Let \(\Sigma = (p, m, n, R, \{ (A_q, B_q, C_q) \} Q \subseteq Q, \bar{x}_0) \) 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 \(\Sigma\) is an \( L\)-equivalent to \(\Sigma\).

**Remark 2:** (Lemma 3) Observe that from Lemma 2 the only Markov parameters involved in the output at a final state of the NDFA are of the form \( C_q A_q x_0 \) where \( v \in (L)_1 \) and \( C_q A_q B_{q_0} \), where \( q_0 v \in s(L)_1 \). However, for the induction in the proof of Claim 1 to work out, it is crucial that the words \( v, q_0 v \) which are indexing the elements of the space \( \text{span}\{A_q x_0, \text{im}(A_q B_{q_0})\} \), must belong to prefix closed sets. Since the smallest prefix-closure of a language \( K \) must be \( K \), the prefix-closure of the sets \( (L)_1 \) and \( \langle s(L) \rangle \) are used in the definition of \( L\)-reachability space; namely the sets \( (L)_1 \) and \( \langle s(L) \rangle \), respectively. This fact leads to matching all the outputs (or the Markov parameters involved in the output) of the original and reduced order systems generated in the course of a switching sequence from \( L \), as opposed to matching only the final outputs. The latter would be sufficient for \( L\)-equivalence.

**Remark 3:** (Lemma 2) Observe that from Lemma 2 the only Markov parameters involved in the output at a final state of the NDFA are of the form \( C_q A_q x_0 \) and \( C_q A_q B_{q_0} \) where \( v q \in s(L) \). In addition, for the induction in the proof of the counterpart of Claim 1 to function in the case of Lemma 4, it suffices that the words \( v q \) which are indexing the elements of the space \( \text{span}(C_q A_q) \) belong to a suffix-closed set. Since \( s(L) \) is already suffix-closed, in this case there is no need to expand the set \( s(L) \) for the definition of \( L\)-unobservability space. Hence the reduced order system found by the use of Lemma 3 will be an \( L\)-realization, but it need not be anything more. These will be illustrated in the last section with numerical examples.

VI. MODEL REDUCTION ALGORITHM

In this section, we present an algorithm 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 \( \mathcal{K}(G) \) and \( \mathcal{K}(H) \) as follows:
\[
\mathcal{K}(G) = \text{span}\{ \text{im}(A_q G) \mid v \in K \}, \quad \mathcal{K}(H) = \bigcap_{v \in K} \ker(A_q v).
\]

Then the \( L\)-reachability space of \( \Sigma \) can be written as
\[
\mathcal{R}_L = \mathcal{R}_{((L)_1)}, (x_0) + \sum_{q \in Q} \mathcal{R}_{q L} \mathcal{R}_L, (B_q),
\]
where \( (L)_1 \) is defined as in (12), and
\[
q(K) = \{ s \in Q^* \mid \exists v_1, v_2 \in Q^*, \hat{q} \in Q, v_1 q s v_2 \hat{q} \in L \}.
\]

In (15), \( + \) and \( \sum \) denote sums of subspaces, i.e. if \( \mathcal{W}, \mathcal{V} \) are two linear subspaces of \( \mathbb{R}^n \), then \( \mathcal{W} + \mathcal{V} = \{ a + b \mid a \in \mathcal{W}, b \in \mathcal{V} \} \). Similarly, if \( \{ \mathcal{W}_i \}_{i \in I} \) is a finite family of linear subspaces of \( \mathbb{R}^n \), then \( \bigcap_{i \in I} \mathcal{W}_i = \{ \sum_{i \in I} a_i \mid a_i \in \mathcal{W}_i, i \in I \} \).

The \( L\)-unobservability space can be written as
\[
\Theta_{L} = \bigcap_{q \in Q} \Theta_{q L} (C_q),
\]
where
\[
(K)^q = \{ s \in Q^* \mid \exists v \in Q^*, v s q \in L \}.
\]

Note that if \( L \) is regular, then \( (K)^q \) and \( (K)^q \), \( q \in Q, (L)_1 \), are also regular and NDFA’s accepting \( (K)^q \), \( q \in Q, (L)_1 \), respectively. It then 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 \( \mathcal{K}(G) \) and \( \mathcal{K}(H) \) 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 \( M, \text{orth}(M) \) denotes the matrix \( U \) whose columns form an orthogonal basis of \( \text{im}(M) \).

**Lemma 5 (Correctness of Algorithm 1 – Algorithm 2):** Algorithm 1 computes \( \mathcal{K}(G) \) and Algorithm 2 computes \( \mathcal{K}(H) \).

Proof: We prove only the first statement of the lemma, the second one can be shown using duality. Let \( P_{s,i} = \text{span}\{ A_q G \mid v \in Q^*, |v| \leq i, x_0 \rightarrow s \} \). It then follows that after the execution of Step 2 \( \text{im}(P_s) = P_{s,0} \) for all \( s \in S \). Moreover, by induction it follows that
\[
P_{s,i+1} = P_{s,i} + \sum_{q \in Q} A_q P_{s,i} \rightarrow s,q
\]
for all \( i = 0, 1, \ldots, s \). Hence, by induction it follows that at the \( i \)th iteration of the loop in Step 3 \( \text{im}(P_s) = P_{s,i} \).
Algorithm 1 Calculate a matrix representation of $\mathcal{R}_k(G)$.
\textbf{Inputs:} $(\{A_q\}_{q \in Q}, G)$ and $\alpha = (S, \{\rightarrow_q\}_{q \in Q}, F, s_0)$ such that $L(\alpha) = K$, $F = \{s_{f_1}, \ldots, s_{f_k}\}$, $k \geq 1$ and $\alpha$ is co-reachable.
\textbf{Outputs:} $\hat{P} \in \mathbb{R}^{n \times p}$ such that $\hat{P}^T \hat{P} = I_p$, rank($\hat{P}$) = $\hat{r}$, im($\hat{P}$) = $\mathcal{R}_k(G)$.

1: $\forall s \in S \setminus \{s_0\}$ : $P_s := 0$.
2: $P_{s_0} := \text{orth}(G)$.
3: flag = 0.
4: while flag = 0 do
5: $\forall s \in S : P_{s_0}^\text{old} := P_s$.
6: for $s \in S$ do
7: $W_s := P_s$.
8: for $q \in Q, s' \in S : s \rightarrow_q s'$ do
9: $W_s := [W_s, A_q P_s^\text{old}].$
10: end for
11: $P_s := \text{orth}(W_s)$.
12: end for
13: if $\forall s \in S : \text{rank}(P_s) = \text{rank}(P_s^\text{old})$ then
14: flag = 1.
15: end if
16: end while
17: return $\hat{P} = \text{orth}\left([P_{s_{f_1}} \ldots P_{s_{f_k}}]\right)$.

Notice that $P_{s_{i+1}} \subseteq P_{s_{i+1}} \subseteq \mathbb{R}^n$ and hence there exists $k_s$ such that $P_{s_{k_s}} = P_{s_{k_s}}$, $k \geq k_s$, and thus $P_{s_{k_s}} = R_s$,

\[ R_s = \text{span}\{\text{im}(A_i, G) \mid v \in Q^*, s_0 \rightarrow_v s\}. \]

Let $k = \max\{k_i : s \in S\}$. It then follows that $P_{s_{k+1}} = P_{s_{k+1}} = \text{im}(P_s)$ for all $s \in Q$ and hence after $k$ iterations, the loop will terminate. Moreover, in that case, $\text{im}(P_{s_{k+1}}) = R_{s_{k+1}}, i \in \{1, \ldots, k\}$. But notice that for any $v \in Q^*, q \in Q, s_0 \rightarrow_v s_f$, and only if $v \in K$, and $s_0 \rightarrow_q s_{f_j}$ and only if $qv \in K$, $i \in \{1, \ldots, k\}$. Hence, the $R_s = \mathcal{R}_k$ and thus $\text{im}\left([P_{s_{f_1}} \ldots P_{s_{f_k}}]\right) = \mathcal{R}_k$.

Notice that the computational complexities of Algorithm 1 and Algorithm 2 are polynomial in $n$, even though the spaces of $\mathcal{R}_k$ (resp. $\mathcal{R}_k$) 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 then imply the following corollary.

\textbf{Lemma 3} is less than the rank of $W$ from Lemma 4. In the second example, the opposite is the case. For both examples, we used the same NDFA from Figure 1 to define the set of admissible switching sequences: the NDFA is defined as the tuple $\alpha = (S, \{\rightarrow_q\}_{q \in Q}, F, s_0)$ 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 $F = \{s_f\}$.

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

1: $\forall s \in S \setminus F : W_s := 0$.
2: $\forall s \in F : W_s^T := \text{orth}(H^T)$.
3: flag = 0.
4: while flag = 0 do
5: $\forall s \in S : W_s^\text{old} := W_s$.
6: for $s \in S$ do
7: $P_s := W_s$.
8: for $q \in Q, s' \in S : s \rightarrow_q s'$ do
9: $P_s := \left[P_s \ W_s^\text{old} A_q\right]$.
10: end for
11: $W_s := \text{orth}(P_s^T)$.
12: end for
13: if $\forall s \in S : \text{rank}(W_s) = \text{rank}(W_s^\text{old})$ then
14: flag = 1.
15: end if
16: end while
17: return $\hat{W} = W_{s_0}$.

Fig. 1. The NDFA $\alpha$ accepting the switching sequence language for both examples.

Observe that the language $L$ accepted by the NDFA $\alpha$ is the set $L = \{12, 123, 1234, \ldots\}$ 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 parameters of the single input - single output (SISO) DTLSS $\Sigma$ of order $n = 7$ with $Q = \{1, 2, 3\}$ used for the first example are of the following form.

\textbf{VII. NUMERICAL EXAMPLES}

In this section, the model reduction method for DTLSSs with restricted discrete dynamics will be illustrated by 2 numerical examples. The data used for both examples and MATLAB codes for the algorithms stated in the paper are available online from https://kom.aau.dk/~merzb/. In the first example, it turns out that the rank of the $P$ matrix from...
Algorithm 3 Reduction for DTLSSs

Inputs: $\Sigma = (p, m, n, Q, \{ (A_q, B_q, C_q) | q \in Q \}, x_0)$ and $\mathcal{A} = (S, \{-q\} \cup F, s_0)$ such that $L(\mathcal{A}) = L$, $F = \{ s_{f_1}, \ldots, s_{f_k} \}$, $k \geq 1$ and $\mathcal{A}$ is co-reachable.

Output: $\tilde{\Sigma} = (p, m, r, Q, \{ (\tilde{A}_q, \tilde{B}_q, \tilde{C}_q) | q \in Q \}, \tilde{x}_0)$.

1. Compute a co-reachable NDFA $\mathcal{A}'$, from $\mathcal{A}$ such that $L(\mathcal{A}') = (L(\mathcal{A}))_s$, where $(L(\mathcal{A}))_s$ is as in (12).
2. Use Algorithm [1] with inputs $\{ (A_q, B_q, C_q) | q \in Q \}$ and NDFA $\mathcal{A}'$. Store the output $\hat{P}$ as $P_{\hat{x}_0} := \hat{P}$.
3. for $q \in Q$ do
4. Compute a co-reachable NDFA $\mathcal{A}_{q}$ from $\mathcal{A}$ such that $L(\mathcal{A}_{q}) = \nu(K)$, where $\nu(K)$ is as in [15].
5. Use Algorithm [2] with inputs $\{ (A_q, B_q) | q \in Q \}$ and NDFA $\mathcal{A}_{q}$. Store the output $\tilde{P}$ as $P_{\tilde{x}_0} := \tilde{P}$.
6. end for
7. $P = \text{orth}(P_{\tilde{x}_0} P_{\hat{x}_1} \cdots P_{\hat{x}_D})$
8. for $q \in Q$ do
9. Compute a co-reachable NDFA $\mathcal{A}_{\sigma_{q}}$ from $\mathcal{A}$, such that $L(\mathcal{A}_{\sigma_{q}}) = \nu^q(K)$, where $\nu^q(K)$ is as in [15].
10. Use Algorithm [2] with inputs $\{ (A_q, B_q,C_q) | q \in Q \}$ and NDFA $\mathcal{A}_{\sigma_{q}}$. Store the output $\tilde{W}$ as $W_{\tilde{x}_0} := \tilde{W}$.
11. end for
12. $W^T = \text{orth}(W_{\tilde{x}_0}^T W_{\hat{x}_1}^T \cdots W_{\hat{x}_D})$
13. if rank($P$) $< \text{rank}(W)$ then
14. Let $r = \text{rank}(P)$, $P^{-1}$ be a left inverse of $P$ and set
   $\tilde{A}_q = P^{-1}A_q P$, $\tilde{C}_q = C_q P$, $\tilde{B}_q = P^{-1}B_q$, $\tilde{x}_0 = P^{-1}x_0$.
15. end if
16. if rank($P$) $\geq \text{rank}(W)$ then
17. Let $r = \text{rank}(W)$ and let $W^{-1}$ be a right inverse of $W$. Set
   $\tilde{A}_q = W A_q W^{-1}$, $\tilde{C}_q = C_q W^{-1}$, $\tilde{B}_q = W B_q$, $\tilde{x}_0 = W x_0$.
18. end if
19. return $\tilde{\Sigma} = (p, m, r, Q, \{ (\tilde{A}_q, \tilde{B}_q, \tilde{C}_q) | q \in Q \}, \tilde{x}_0)$.

where $\text{randn}$ and $\text{zeros}$ are the MATLAB functions which generates arrays containing random real numbers with standard normal distribution and zeros respectively. Applying Algorithm [3] to this DTLSS whose admissible switching sequences are generated by the NDFA shown in Figure [1] yields a reduced order system $\tilde{\Sigma}$ of order $r = 4$, whose output values are the same as the original system $\Sigma$ along the allowed switching sequences. This corresponds to an $L$-realization in the sense of Definition [5] since the language $L$ of the NDFA $\mathcal{A}$ is defined as the set of all words generated by $\mathcal{A}$ starting from its initial state and ending in a final state. In this example, it turns out the algorithm makes use of Lemma [2] and constructs the $P$ matrix. The $P \in \mathbb{R}^{n \times r}$ matrix acquired is $P = \begin{bmatrix} 1 & 0 \\ 0 & \end{bmatrix}$. Note that as stated in Remark [2] the resulting DTLSS is more than just an $L$-realization of $\Sigma$, its output coincides with the output of $\Sigma$ for all instances along the allowed switching sequences, rather than just the instances corresponding to the final states of the NDFA (the switching sequence generated by $\mathcal{A}$ used for simulating the examples is given in (19)). This fact is visible from Figure [2] where it can be seen that output of both systems corresponding to all instances along the switching sequence $\sigma$ of length $|\sigma| = 11$ defined by

$$\sigma = 12312312312$$

Coincide (the input sequence of length 11 used in the simulation is also generated by the function $\text{randn}$). Finally, observe that the DTLSS $\Sigma$ is minimal (note that the definition of minimality for linear switched systems are made by considering all possible switching sequences in $Q^+$ [16]), however for the switching sequences restricted by the NDFA $\mathcal{A}$, it turns out 3 states are disposable. In fact, this is the main idea of the paper.

![Fig. 2. Example 1: The responses of the original DTLSS $\Sigma$ of order 7 and the reduced order DTLSS $\tilde{\Sigma}$ of order 4 acquired by Algorithm [3] for the switching sequence in (19)](image)
Applying Algorithm 4 to this DTLSS, yields a reduced order system $\hat{\Sigma}$ of order $r = 3$, whose output values are the same as the original system $\Sigma$ 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 $\Sigma$ and $\hat{\Sigma}$ are the same for all the switching sequences generated by the governing NDFA, i.e., for all $\sigma \in L(\epsilon)$).

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 $W = \begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \\ A_3 & B_3 \end{bmatrix}$.

In this example, note that the resulting DTLSS is merely an $L$-realization of $\Sigma$ and nothing more as stated in Remark 3, i.e., its output coincides with the output of $\Sigma$ for only the instances corresponding to the final states of the NDFA. This fact is visible from Figure 3 where it can be seen that output corresponding to the final state $s_f$ of the NDFA coincides for $\Sigma$ and $\hat{\Sigma}$ (Observe that for all switching sequences generated by $\epsilon$ ending with the label 2, the output values of $\Sigma$ and $\hat{\Sigma}$ are the same). Again, the input sequence of length 11 used in the simulation is generated by the function $\text{randn}$. Finally, note that the DTLSS $\Sigma$ is again minimal whereas for the switching sequences restricted by the NDFA $\epsilon$, it turns out 4 states are disposable in this case.

![Responses of the original and reduced order systems](image)

**Fig. 3.** Example 2: The responses of the original DTLSS $\Sigma$ of order 7 and the reduced order DTLSS $\hat{\Sigma}$ of order 3 acquired by Algorithm 4 for the switching sequence in Figure 19. Note that only the second and fifth element of the output sequence is equal for $\Sigma$ and $\hat{\Sigma}$ in first five elements, others look the same as a result of the scaling.

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


