Model Reduction by Moment Matching for Linear Switched Systems

Mert Baştug\textsuperscript{1,2}, Mihály Petreczky\textsuperscript{2}, Rafael Wisniewski\textsuperscript{1} and John Leth\textsuperscript{1}

Abstract—A moment-matching method for the model reduction of linear switched systems (LSSs) is developed. The method is based upon a partial realization theory of LSSs and it is similar to the Krylov subspace methods used for moment matching for linear systems. The results are illustrated by a numerical example.

I. INTRODUCTION

A linear switched system (abbreviated by LSS) is a system which switches among a finite number of linear subsystems. That is, the state and output trajectory of an LSS is a concatenation of the state and output trajectories of the linear subsystems. At each time instance, the active linear subsystem is determined by a switching signal, which is considered as an additional external input to the system, i.e., any switching sequence is admissible. Linear switched systems represent a class of hybrid systems and they have been studied extensively, see [12], [25] for an overview.

Model reduction is the problem of approximating a dynamical system with another one of smaller complexity. For LSSs, by complexity, we will mean the number of continuous states, and by model reduction we will mean the approximation of the original LSS by another one, with a smaller number of continuous states.

Contribution of the paper In this paper, we present a model reduction algorithm based on the partial realization theory for LSSs [21]. The main idea is to replace the original LSS by an LSS of smaller order, such that a finite number of Markov parameters of the two LSSs coincide. By Markov parameters of the LSS we mean the Markov parameters of its input-output map, as defined in [21]. When applied to the linear case, the definition of [21] yields the usual definition of Markov parameters. The Markov parameters can be interpreted as high-order partial derivatives of the input-output maps with respect to the switching times. Hence, if those Markov parameters of the two LSSs which correspond to lower order partial derivatives coincide, then, by the same logic as used in Taylor series approximation, one could say that the two LSSs (more precisely, their input-output maps) are close. Hence, the proposed algorithm extends the well-known moment matching approach for linear systems [1]. By analogy with the linear case, we will refer to the proposed model reduction approach as moment matching for LSSs. The contribution of the paper is thus twofold: (1) the paper formulates model reduction by moment matching for LSSs, and (2) it presents an algorithm for computing the reduced order system. The algorithm presented in this paper is similar to Krylov subspace methods for approximation of linear systems.

Motivation The order of the controller and the computational complexity of controller synthesis usually increase as the number of continuous states of the plant model increases. Hence, the smaller the plant model is, the easier it is to synthesize the control law and to implement it. This becomes especially relevant for hybrid systems, as many of the existing control synthesis methods are computationally demanding and result in large scale controllers. For example, many of the existing control synthesis methods rely on computing a finite-state abstraction of the plant model [26]. Often, the computational complexity of these methods and the size of the controller are exponential in the number of continuous states of the plant. This gets even worse for control problems with partial observation [18]. This means that even plant models of moderate size can become intractable and even a small reduction in the number of states can make a difference. For this reason, we expect that model reduction of switched systems will be useful for control of switched systems.

Related work In the linear case, model reduction is a mature research area, see [1] and the references therein. The subject of model reduction for hybrid and switched systems was addressed in several papers [4], [30], [15], [5], [9], [28], [29], [7], [10], [11], [16], [24]. Except [9], the cited papers propose techniques which involve solving certain LMIs, and for this reason, they tend to be applicable only to switched systems for which the continuous subsystems are stable. In contrast, the approach of this paper works for systems which are unstable. However, this comes at a price, since we are not able to propose analytic error bounds, like the ones for balanced truncation [23]. From a practical point of view, the lack of an analytic error bound need not be a very serious disadvantage, since it is often acceptable to evaluate the accuracy of the approximation after the reduced model has been computed. For example, one can compute the $L_2$ distance between the original and reduced order model [23].

Note that the proposed algorithm does have a system theoretic interpretation, as it operates on the Markov parameters. Note also that the Markov parameters of switched systems characterize their input-output behavior uniquely [17]. Moreover, the Euclidean distance between Markov parameters can be used to define a natural distance for LSSs with state-space representations [22], [20]. For this reason, we believe that it might be possible to give a theoretical justification...
for the proposed algorithm. However, this remains a topic
of future research. Note that even in the linear case, it is
very challenging to give analytic error bounds for algorithms
based on moment matching. [1].

The model reduction algorithm proposed in this paper is
similar in spirit to moment matching for linear systems [1],
[8] and bilinear systems [13], [2], [6]; however, the details
and the system class considered are entirely different. The
model reduction algorithm for LPV systems described in [27]
is related, as it also relies on a realization algorithm and
Markov parameters. In turn, the realization algorithms and
Markov parameters of LPV systems and LSSs are closely
related, [19]. However, the algorithm of [27] applies to a
different system class, and it is not yet clear if it yields a
partial realization.

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 LSSs. In Section IV, 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 V in detail. Finally,
in Section VI the algorithm is illustrated on a numerical
example.

II. PRELIMINARIES: NOTATION AND TERMINOLOGY

Denote by \( \mathbb{N} \) the set of natural numbers including 0. Denote by \( \mathbb{R}^+ \) the set \([0, +\infty)\) of nonnegative real numbers. In the sequel, let \( PC(\mathbb{R}^+, S) \), with \( S \) a topological subspace
of an Euclidean space \( \mathbb{R}^n \), denote the set of piecwise-
continuous and left-continuous maps. That is, \( f \in PC(\mathbb{R}^+, S) \)
if it has finitely many points of discontinuity on any compact
subinterval of \( \mathbb{R}^+ \), and at any point of discontinuity both
the left-hand and right-hand side limits exists, and \( f \) is
continuous from the left. In addition, denote by \( AC(\mathbb{R}^+, \mathbb{R}^n) \)
the set of absolutely continuous maps, and \( L_{loc}(\mathbb{R}^+, \mathbb{R}^n) \) the
set of Lebesgue measurable maps which are integrable on
any compact interval.

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 [17], [23].

Definition I (LSS): A continuous time linear switched
system (LSS) is a control system of the form
\[
\begin{align*}
\frac{dx(t)}{dt} &= A_\sigma(t)x(t) + B_\sigma(t)u(t), \quad x(0) = x_0 \quad (1a) \\
y(t) &= C_\sigma(t)x(t) \quad (1b)
\end{align*}
\]
where \( Q = \{1, \ldots, D\} \), \( D > 0 \), called the set of discrete
modes, \( \sigma \in PC(\mathbb{R}^+, Q) \) is called the switching signal, \( u \in \mathbb{R}^m \) is called the input, \( x \in AC(\mathbb{R}^+, \mathbb{R}^n) \) is called
the state, and \( y \in PC(\mathbb{R}^+, \mathbb{R}) \) is called the output. Moreover,
\( A_q \in \mathbb{R}^{n \times n} \), \( B_q \in \mathbb{R}^{n \times m} \), \( C_q \in \mathbb{R}^{p \times n} \) are the matrices of the
linear system in mode \( q \in Q \), and \( x_0 \) is the initial state. The
notation
\[
\Sigma = (p, m, n, Q, \{ (A_q, B_q, C_q) | q \in Q \}, x_0) \quad (2)
\]
is used as a short-hand representation for LSSs of the form
(1). The number \( n \) is called the dimension (order) of \( \Sigma \) and
will sometimes be denoted by \( \dim \Sigma \).

Throughout the paper, \( \Sigma \) denotes an LSS of the form (1). Next, we present the basic system theoretic concepts for
LSSs.

Definition 2: The input-to-state map \( X_{\Sigma, x} \) and input-
to-output map \( Y_{\Sigma, x} \) of \( \Sigma \) are the maps
\[
\begin{align*}
X_{\Sigma, x} : L_{loc}(\mathbb{R}^+, \mathbb{R}^m) \times PC(\mathbb{R}^+, Q) \rightarrow AC(\mathbb{R}^+, \mathbb{R}^n); (u, \sigma) \mapsto X_{\Sigma, x}(u, \sigma) \\
Y_{\Sigma, x} : L_{loc}(\mathbb{R}^+, \mathbb{R}^m) \times PC(\mathbb{R}^+, Q) \rightarrow PC(\mathbb{R}^+, \mathbb{R}^p); (u, \sigma) \mapsto Y_{\Sigma, x}(u, \sigma)
\end{align*}
\]
defined by letting \( t \mapsto X_{\Sigma, x}(u, \sigma)(t) \) be the solution to the
Cauchy problem (1a) with \( t_0 = 0 \) and \( x_0 = x \), and letting
\( Y_{\Sigma, x}(u, \sigma)(t) = C_\sigma(t)X_{\Sigma, x}(u, \sigma)(t) \) as in (1b).

The input-output behavior of an LSS realization can be
formalized as a map
\[
f : L_{loc}(\mathbb{R}^+, \mathbb{R}^m) \times PC(\mathbb{R}^+, Q) \rightarrow PC(\mathbb{R}^+, \mathbb{R}^p). \quad (3)
\]
The value \( f(u, \sigma) \) represents the output of the underlying
(black-box) system. This system may or may not admit a
description by an LSS. Next, we define when an LSS
describes (realizes) a map of the form (3).

The LSS \( \Sigma \) of the form (1) is a realization of an input-
output map \( f \) of the form (3), if \( f \) is the input-output map of
\( \Sigma \) which corresponds to some initial state \( x_0 \), i.e., \( f = Y_{\Sigma, x_0} \).
The map \( Y_{\Sigma, x_0} \) will be referred to as the input-output map of
\( \Sigma \), and it will be denoted by \( Y_{\Sigma} \). The following discussion is
only for realizable input-output maps.

We say that the LSSs \( \Sigma_1 \) and \( \Sigma_2 \) are equivalent if \( Y_{\Sigma_1} = Y_{\Sigma_2} \).
The LSS \( \Sigma_m \) is said to be a minimal realization of \( f \), if \( \Sigma_m \)
is a realization of \( f \), and for any LSS \( \Sigma \) such that \( \Sigma \) is a realization of \( f \), \( \dim \Sigma_m \leq \dim \Sigma \). In [17], it is stated that
an LSS realization \( \Sigma \) is minimal if and only if it is span-
reachable and observable. See [17] for detailed definitions of
span-reachability and observability for LSSs. Moreover, if \( \Sigma \) is a realization of \( f \), then there exists an algorithm for
computing from \( \Sigma \) a minimal realization \( \Sigma_m \) of \( f \), see [17],
[23]. Hence, in the sequel, unless stated otherwise we will
assume that the LSSs are minimal realizations of their input-
output maps.

IV. MOMENT MATCHING FOR LINEAR SWITCHED
SYSTEMS: PROBLEM FORMULATION

In this section, we state formally the problem of moment
matching for LSSs.

Notation I: Consider a finite non-empty set \( Q \) with \( D \)
elements, 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 \( Q \). Each non-empty word
\( w \) is of the form \( w = q_1q_2 \cdots q_k \) for some \( q_1, q_2, \ldots, q_k \in Q \).
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 \( \epsilon \). 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 concatenation of word $w\in Q^*$ with $v\in Q$ is denoted by $vw$: if $v=v_1v_2\cdots v_k$, and $w=w_1w_2\cdots w_m$, $k>0$, $m>0$, $v_1,v_2,\ldots,v_k,w_1,w_2,\ldots,w_m\in Q$, then $vw=v_1v_2\cdots v_kw_1w_2\cdots w_m$. If $v=\varepsilon$, then $vw=w$; if $w=\varepsilon$, then $vw=v$. For simplicity, the finite set $Q$ will be identified with its index set, that is $Q=\{1,2,\cdots,D\}$.

Next, consider an input-output map $f$ of the form (3). Notice that the restriction to a finite interval $[0,t]$ of any $\sigma\in PC(R^+,Q)$ can be interpreted as finite sequence of elements of $Q\times R_+$ of the form

$$\mu(\sigma) = (q_1,t_1)(q_2,t_2)\cdots(q_k,t_k)$$

(4) where $q_1,\ldots,q_k\in Q$ and $t_1,\ldots,t_k\in\R^+_\setminus\{0\}, t_1+\cdots+t_k = t$, such that for all $s\in[0,t]$ 

$$\sigma(s) = \begin{cases} 
q_1 & \text{if } s\in[0,t_1] \\
q_2 & \text{if } s\in(t_1,t_1+t_2] \\
 \vdots & \text{if } s\in(t_1+\cdots+t_{i-1},t_1+\cdots+t_{i-1}+t_i] \\
q_i & \text{if } s\in(t_1+\cdots+t_{i-1},t_1+\cdots+t_{i-1}+t_i+\cdots+t_{i+k}] \\
q_k & \text{if } s\in(t_1+\cdots+t_{i-1},t_1+\cdots+t_{i-1}+t_i+\cdots+t_{i+k}+\cdots+t_k] 
\end{cases}$$

From [17], it follows that a necessary condition for $f$ to be realizable by an LSS is that $f$ has a generalized kernel representation. For a detailed definition of a generalized kernel representation of $f$, we refer the reader to [17, Definition 19]. For our purposes, it is sufficient to recall that if $f$ has a generalized kernel representation, then there exists a unique family of analytic functions $\{K_{q_1\cdots q_k}:R^+\to R^p\}$ and $\{G_{q_1\cdots q_k}:R^+\to R^{p\times m}\}$, $q_1,\ldots,q_k\in Q, k\geq 1$, such that for all $(u,\sigma)\in L_{f0}(R^+,R^m)\times PC(R^+,Q)$, $t>0$ 

$$f(u,\sigma)(t) = K_{q_1q_2\cdots q_k}(t_1,t_2,\cdots,t_k) + \sum_{i=1}^k \int_{0}^{t_i} G_{q_iq_{i+1}\cdots q_k}(t_i-s,t_{i+1}\cdots t_k)u(s) + \sum_{j=1}^{i-1} t_j) \, ds,$$

with $\mu(\sigma) = (q_1,t_1)(q_2,t_2)\cdots(q_k,t_k)$ and the functions $\{K_{q_1\cdots q_k},G_{q_1\cdots q_k} \mid q_1,\ldots,q_k\in Q,k\geq 0\}$ satisfy a number of technical conditions, see [17, Definition 19] for details. From [17], it follows that there is a one-to-one correspondence between $f$ and the family of maps $\{K_{q_1\cdots q_k},G_{q_1\cdots q_k} \mid q_1,\ldots,q_k\in Q,k\geq 0\}$. If $f$ has a realization by an LSS 1, then the functions $K_{q_1q_2\cdots q_k}(t_1,t_2,\cdots,t_k)$ and $G_{q_1q_2\cdots q_k}(t_1,t_2,\cdots,t_k)$ satisfy 

$$K_{q_1q_2\cdots q_k}(t_1,t_2,\cdots,t_k) = C_{q_k}e^{A_{q_k}t_k}e^{A_{q_{k-1}}t_{k-1}}\cdots e^{A_{q_1}t_1}\mathbf{x}_0$$

$$G_{q_1q_2\cdots q_k}(t_1,t_2,\cdots,t_k) = C_{q_k}e^{A_{q_k}t_k}e^{A_{q_{k-1}}t_{k-1}}\cdots e^{A_{q_1}t_1}B_{q_k}.$$ 

We can now define the Markov parameters of $f$ as follows. 

\textbf{Definition 3 (Markov parameters):} Let $f$ have a GCR. The Markov parameters of $f$ are the values of the map 

$$M^f:Q^*\to R^{p\times (md+1)}$$

defined by, 

$$M^f(v) = \begin{bmatrix}
S_0(v_1), & S_1(v_1) & \cdots & S_D(v_1) \\
S_0(v_2), & S_1(v_2) & \cdots & S_D(v_2) \\
\vdots & \vdots & \ddots & \vdots \\
S_0(v_D), & S_1(v_D) & \cdots & S_D(v_D)
\end{bmatrix},$$

where the vectors $S_0(vq)\in R^p$ and the matrices $S(q_0vq)\in R^{p\times m}$ are defined as follows. For all $q_0,q\in Q$,

$$S_0(q) = K^f_q(0)$$

and for all $q_0,q\in Q$,

$$S(q_0vq) = \frac{\partial}{\partial t_1} \cdots \frac{\partial}{\partial t_k} K^f_{q_1q_2\cdots q_k}(t_1,\cdots,t_k,0) \Bigr|_{t_1=\cdots=t_k=0}.$$
that of the minimal LSS realization \( f \). In order to formalize the intuition described above, we introduce the following definitions.

**Definition 4 (Partial realization):** The LSS (1) is called \( N \)-partial realization of \( f \), if

\[
\forall v \in Q^*, |v| \leq N : M^f(v) = \tilde{C}A^f_B
\]

with \( \tilde{C} = [C_1^T \cdots C_D^T]^T \) and \( \tilde{B} = [x_0 \ B_1 \ B_2 \cdots B_D] \). If \( \Sigma \) is of the form (1) and \( Y_2 \) is the input-output map of \( \Sigma \), then the concept of \( N \)-partial realization can be interpreted as follows: \( \Sigma \) is an \( N \)-partial realization of \( f \), if those Markov parameters of \( f \) and \( Y_2 \) which are indexed by words of length at most \( N \) coincide. The problem of model reduction by moment matching can now be formalized as follows.

**Problem 1 (Moment matching):** Let \( \Sigma \) be an LSS (1) and let \( f = Y_2 \) be its input-output map. Fix \( N \in \mathbb{N} \). Find an LSS \( \tilde{\Sigma} \) such that \( \dim \tilde{\Sigma} < \dim \Sigma \) and \( \tilde{\Sigma} \) is an \( N \)-partial realization of \( f = Y_2 \).

Note that there is a trade-off between the choice of \( N \) and the dimension \( \Sigma \). This follows from the result of [21, Theorem 4].

**Theorem 1:** Assume that \( \Sigma \) is a minimal realization of \( f \) and \( 2\dim \Sigma - 1 \leq N \). Then for any LSS \( \Sigma \) which is an \( N \)-partial realization of \( f \), \( \dim \Sigma \leq \dim \Sigma \).

That is, if we choose \( N \) too high, namely if we choose any \( N \) such that \( N \geq 2n - 1 \), where \( n \) is the dimension of a minimal LSSs realization of \( f \), then there will be no hope of finding an LSS which is an \( N \)-partial realization of the original input-output map, and whose dimension is lower than \( n \).

In order to solve the moment matching problem, one could consider applying the partial realization algorithm [21]. However, that approach yields a model reduction algorithm whose memory-usage and run-time complexity is exponential.

**V. THE MODEL REDUCTION ALGORITHM**

In this section, the aim is to present an efficient model reduction algorithm which transforms an LSS \( \Sigma \) into an LSS \( \tilde{\Sigma} \) such that \( \dim \tilde{\Sigma} \leq \dim \Sigma \) and \( \tilde{\Sigma} \) is an \( N \)-partial realization of the input-output map of \( \Sigma \). The presented algorithm has polynomial computational complexity and does not involve the explicit computation of the Hankel matrix as in [21].

In the sequel, the image (column space) of a real matrix \( M \) is denoted by \( \text{im}(M) \) and \( \text{rank}(M) \) is the dimension of \( \text{im}(M) \). We will start with presenting the following definitions.

**Definition 5 ((Partial) Unobservability subspace):** For an LSS \( \Sigma \), and \( N \in \mathbb{N} \) define the \( N \)-step unobservability subspace as

\[
\mathcal{O}_N(\Sigma) = \bigcap_{v \in Q^*, |v| \leq N} \ker(C_{A^f_v}).
\]

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

It is not difficult to see that \( \mathcal{O}_0 = \bigcap_{q \in Q} \ker(C_q) \) and for any \( N > 0 \), \( \mathcal{O}_N = \mathcal{O}_0 \cap \bigcap_{q \in Q} \mathcal{O}_{N-1} \). From [25], [17], it follows that \( \Sigma \) is observable if and only if \( \mathcal{O}_N(\Sigma) = \{0\} \) for all \( N \geq \dim \Sigma - 1 \).

**Definition 6 ((Partial) Reachability space):** For an LSS \( \Sigma \), define the \( N \)-step reachability space as follows:

\[
\mathcal{R}_N(\Sigma) = \text{Span}\{\text{im}(A^f_v) \mid v \in Q^*, |v| \leq N\},
\] (7)

where \( \tilde{B} = [x_0, B_1, B_2, \cdots, B_D] \). If \( \Sigma \) is clear from the context, we will denote \( \mathcal{R}_N(\Sigma) \) by \( \mathcal{R}_N \).

It is easy to see that \( \mathcal{R}_0 = \text{im}(\tilde{B}) \) and \( \mathcal{R}_N = \text{im}(\tilde{B}) + \sum_{q \in Q} A_q \mathcal{R}_{N-1} \), for \( N > 0 \). It follows from [17], [25] that \( \Sigma \) is span-reachable if and only if \( \dim \mathcal{R}_N \leq n \) for all \( N \geq n - 1 \).

**Theorem 2 (One sided moment matching (columns)):** Let \( \Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) \mid q \in Q\}, x_0) \) be an LSS realization of the input-output map \( f, P \in \mathbb{R}^{n \times r} \) be a full column rank matrix such that

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

If \( \Sigma = (p, m, r, Q, \{(\tilde{A}_q, \tilde{B}_q, \tilde{C}_q) \mid q \in Q\}, \tilde{x}_0) \) is an LSS such that for each \( q \in Q \), the matrices \( \tilde{A}_q, \tilde{B}_q, \tilde{C}_q \) and the vector \( \tilde{x}_0 \) are defined as

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

where \( P^{-1} \) is a left inverse of \( P \), then \( \tilde{\Sigma} \) is an \( N \)-partial realization of \( f \).

**Proof:** See [3].

Using a dual argument, the following result can be proven.

**Theorem 3 (One sided moment matching (observability)):** Let \( \Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) \mid q \in Q\}, x_0) \) be an LSS realization of the input-output map \( f, W \in \mathbb{R}^{r \times n} \) be a full row rank matrix such that

\[
\mathcal{O}_N(\Sigma) = \ker(W).
\]

Let \( W^{-1} \) be any right inverse of \( W \) and let \( \Sigma = (p, m, r, Q, \{(A_q, B_q, C_q) \mid q \in Q\}, \tilde{x}_0) \) be an LSS such that for each \( q \in Q \), the matrices \( A_q, B_q, C_q \) and the vector \( \tilde{x}_0 \) are defined as

\[
\tilde{A}_q = W A_q W^{-1}, \quad \tilde{B}_q = W B_q, \quad \tilde{C}_q = C_q W^{-1}, \quad \tilde{x}_0 = W x_0.
\]

Then \( \tilde{\Sigma} \) is an \( N \)-partial realization of \( f \).

Finally, by combining the proofs of Theorem 2 and Theorem 3, the following can be shown.

**Theorem 4 (Two sided moment matching):** Let

\[
\Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) \mid q \in Q\}, x_0)
\]

be an LSS realization of the input-output map \( f, V \in \mathbb{R}^{r \times r} \) and \( W \in \mathbb{R}^{r \times n} \) be respectively full column rank and full row matrices such that

\[
\mathcal{R}_N(\Sigma) = \text{im}(V), \quad \mathcal{O}_N(\Sigma) = \ker(W) \text{ and } \ker(WV).
\]

If \( \Sigma = (p, m, r, Q, \{(\tilde{A}_q, \tilde{B}_q, \tilde{C}_q) \mid q \in Q\}, \tilde{x}_0) \) is an LSS such that for each \( q \in Q \), the matrices \( A_q, B_q, C_q \) and the vector \( x_0 \) are defined as

\[
\tilde{A}_q = W A_q W(VW)^{-1}, \quad \tilde{B}_q = W B_q, \quad \tilde{C}_q = C_q V(WV)^{-1}, \quad \tilde{x}_0 = W x_0,
\]

then \( \tilde{\Sigma} \) is a \( 2N \)-partial realization of \( f \).

Now, we will present an efficient algorithm of model reduction by moment matching, which computes either an \( N \) or \( 2N \)-partial realization \( \tilde{\Sigma} \) for an \( f \) which is realized by an LSS \( \Sigma \). First, we present algorithms for computing the subspaces \( \mathcal{R}_N \) and \( \mathcal{O}_N \). To this end, we will use the following notation: if \( M \) is any real matrix, then denote by \( \text{orth}(M) \) the matrix \( U \) such that \( U \) is full column rank, \( \text{rank}(U) = \text{rank}(M) \), \( \text{im}(U) = \text{im}(M) \) and \( UU^T = I \). Note
that \( U \) can easily be computed from \( M \) numerically, see
for example the Matlab command \texttt{orth}. The algorithm for
computing \( \mathcal{O}_N \) is presented in Algorithm 1 below. By duality,

\begin{algorithm}
\textbf{Algorithm 1} Calculate a matrix representation of \( \mathcal{O}_N \),
\begin{itemize}
  \item \textbf{Inputs:} \( \{(A_q, B_q) \mid q \in Q, \xi_0 \} \) and \( N \)
  \item \textbf{Outputs:} \( P \in \mathbb{R}^{r \times n} \) such that \( \text{rank}(P) = r \), \( \text{im}(P) = \mathcal{O}_N \).
\end{itemize}
\begin{algorithmic}
  \State \texttt{P} := \texttt{U}_0, \texttt{U}_0 := \texttt{ort}([x_0, B_1, \ldots, B_D]).
  \For {\texttt{k} \in 1 \ldots N}
      \State \texttt{P} := \texttt{ort}([\texttt{U}_0, A_1 P, A_2 P, \ldots, A_D P]).
  \EndFor
  \State \textbf{return} \texttt{P}.
\end{algorithmic}
\end{algorithm}

we can use Algorithm 1 to compute \( \mathcal{O}_N \), the details are
presented in Algorithm 2.

\begin{algorithm}
\textbf{Algorithm 2} Calculate a matrix representation of \( \mathcal{O}_N \)
\begin{itemize}
  \item \textbf{Inputs:} \( \{A_q, C_q\} \mid q \in Q \) and \( N \)
  \item \textbf{Output:} \( W \in \mathbb{R}^{r \times n} \) such that \( \text{rank}(W) = r \) and \( \ker(W) = \mathcal{O}_N \).
\end{itemize}
\begin{algorithmic}
  \State Apply Algorithm 1 with inputs \( \{(A_q^T C_q^T) \mid q \in Q, 0\} \) to obtain
  \State a matrix \( P \).
  \State \textbf{return} \( W = P^T \).
\end{algorithmic}
\end{algorithm}

Notice that the computational complexity of Algorithm 1 and
Algorithm 2 is polynomial in \( N \) and \( n \), even though
the spaces of \( \mathcal{O}_N \) (resp. \( \mathcal{O}_N \)) are generated by images (resp.
kerneals) of exponentially many matrices.

Using Algorithm 1 and 2, we can formulate a model
reduction algorithm, see Algorithm 3.

\begin{algorithm}
\textbf{Algorithm 3} Moment matching for LSSs
\begin{itemize}
  \item \textbf{Inputs:} \( \Sigma = (p, m, n, Q, \{(A_q, B_q, C_q) \mid q \in Q, \xi_0\}) \) and \( N \in \mathbb{N} \).
  \item \textbf{Output:} \( \tilde{\Sigma} = (p, m, r, Q, \{(\tilde{A}_q, \tilde{B}_q, \tilde{C}_q) \mid q \in Q, \tilde{\xi}_0\}) \).
\end{itemize}
\begin{algorithmic}
  \State Using Algorithm 1-2 compute matrices \( P \) and \( W \) such that
  \State \( P \) is full column rank, \( W \) is full row rank and \( \text{im}(P) = \mathcal{O}_N \), \( \ker(W) = \mathcal{O}_N \).
  \If {\text{rank}(P) = \text{rank}(W) = \text{rank}(WP)}
      \State \texttt{r} := \texttt{rank}(P)
      \State \( \tilde{A}_q = WA_qP(WP)^{-1}, \tilde{C}_q = C_qP(WP)^{-1}, \tilde{B}_q = WB_q, \tilde{\xi}_0 = W\xi_0 \).
  \EndIf
  \If {\text{rank}(P) \geq \text{rank}(W)}
      \State \texttt{r} := \texttt{rank}(P) \quad \text{P}^{-1} \quad \text{be a left inverse of } P \quad \text{and set}
      \State \( \tilde{A}_q = P^{-1}A_qP, \tilde{C}_q = C_qP, \tilde{B}_q = P^{-1}B_q, \tilde{\xi}_0 = P^{-1}\xi_0 \).
  \EndIf
  \If {\text{rank}(P) < \text{rank}(W)}
      \State \texttt{r} := \texttt{rank}(W) \quad \text{and let } W^{-1} \quad \text{be a right inverse of } W \quad \text{Set}
      \State \( \tilde{A}_q = WA_qW^{-1}, \tilde{C}_q = C_qW^{-1}, \tilde{B}_q = WB_q, \tilde{\xi}_0 = W\xi_0 \).
  \EndIf
  \State \textbf{return} \( \tilde{\Sigma} = (p, m, r, Q, \{(\tilde{A}_q, \tilde{B}_q, \tilde{C}_q) \mid q \in Q, \tilde{\xi}_0\}) \).
\end{algorithmic}
\end{algorithm}

Theorem 2 – 4 imply the following corollary on correctness of Algorithm 3.

\textbf{Corollary 1 (Correctness of Algorithm 3):} Using the notations of Algorithm 3, the following holds: If \( \text{rank}(P) = \text{rank}(W) = \text{rank}(WP) \), then Algorithm 3 returns an \( 2N \) partial realization of \( f = Y_2 \). Otherwise, Algorithm 3 returns an \( N \) partial realization of \( f = Y_2 \).

\textbf{Remark 1 (Implementation):} The implementation of Algorithm 3 in MATLAB is available from \url{https://kom.aau.dk/~merthb/}. A random switching signal with minimum dwell time (time between two subsequent changes in the switching signal) of 0.1 and a random input signal \( u(t) \) with uniform distribution is used for simulation. The simulation time interval is \( t = [0, 3] \). For \( N = 1 \), an approximation LSS of order 9 whose Markov parameters indexed by the words of length at most 1 are matched with the original LSS \( \Sigma \) is acquired, i.e., the original LSS \( \Sigma \) is approximated by a 1-partial realization \( \tilde{\Sigma}_1 \). Note that the precise number of matched Markov parameters is equal to the number of words in the set \( Q^* \) of length at most \( N = 1 \), which is in this case 3. The output \( y(t) \) of the original system \( \Sigma \) and the output \( \tilde{y}(t) \) of the reduced order system \( \tilde{\Sigma}_1 \) are simulated with the parameters given for 500 random switching sequences and input trajectories. For each simulation, the responses of the original and reduced order LSSs are compared with the best fit rate (BFR) (see \cite{14, 27}) which is defined as

\[
\text{BFR} = 100\% \max \left( 1 - \frac{||y(-) - \tilde{y}(-)||_2}{||y(-) - y_m||_2}, 0 \right)
\]

where \( y_m \) is the mean of \( y \) and \( ||\cdot||_2 \) is the \( \ell_2 \) norm. Even though the BFR is defined for output sequences rather than functions with the domain \( \mathbb{R}^+ \), we could still apply it, since \( y \) and \( \tilde{y} \) were obtained by computing a numerical solution of the LSS, and as a result they are both defined on a discretization of the time axis, i.e., \( y \) and \( \tilde{y} \) are arrays containing the output values in the sampled time instances. For this example, the mean of the BFRs for 500 simulations is acquired as 79.0518%; whereas, the best acquired BFR is 90.8013% and the worst is 62.7846%. The outputs \( y(t) \) and \( \tilde{y}(t) \) of the most successful simulation are shown in Fig. 1.
VII. CONCLUSIONS

A moment matching procedure for model reduction of LSSs has been given. It has been proven that with this procedure, as long as a certain criterion is satisfied, it is possible to acquire at least one reduced order approximation to the original LSS whose first some number of Markov parameters are matched with the original one's. The procedure is based on constructing matrices whose image or kernel is the partial reachability or unobservability subspaces of an LSS respectively. Since we do not explicitly compute the Hankel matrices, the computational complexity does not increase exponentially with the number of moments to be matched, which is particularly important for large scale systems.

REFERENCES