Realization of multi-input/multi-output switched linear systems from Markov parameters

Fethi Bencherki\textsuperscript{a,1}, Semiha T"urkay\textsuperscript{b}, H"useyin Akçay\textsuperscript{b,∗}

\textsuperscript{a}Department of Automatic Control, Lund University, Box 117, SE-221 00, Sweden
\textsuperscript{b}Department of Electrical and Electronics Engineering, Eskisehir Technical University, Eskisehir, 26555, Turkey

Abstract

This paper presents a four-stage algorithm for the realization of multi-input/multi-output (MIMO) switched linear systems (SLSs) from Markov parameters. In the first stage, a linear time-varying (LTV) realization that is topologically equivalent to the true SLS on intervals of interest is derived from the Markov parameters assuming that the discrete states have a common MacMillan degree and a mild condition on their dwell times holds. In the second stage, stationary point set of a Hankel matrix with fixed block column and row dimensions and built from the Markov parameters is examined. Splitting of this set into a union of disjoint intervals and intersection of their complements reveals linear time-invariant dynamics prevailing on these intervals. Absolute sum of modal eigenvalues is used as a feature for clustering to extract the discrete states, up to arbitrary similarity transformations. Recovery of the discrete states is complete if a mild unimodality assumption holds and each discrete state visits at least one segment and stays there long enough. In the third stage, the switching sequence is estimated by three schemes. The first scheme is based on correcting the state-space matrices estimated by the LTV realization algorithm from the Markov parameters. It starts from a discrete state estimate, proceeds along the positive and/or the negative real axes, and continues until a switch or a switch pair is detected. This process is repeated until all switches compatible with the dwell time requirements are detected. The second scheme is based on matching the estimated and the true Markov parameters of the SLS system over segments of the switching sequence. The third scheme is also based on matching the Markov parameters, but it is a discrete optimization/hypothesis testing algorithm. The three schemes operate on different dwell time and model structure requirements, but their dwell time requirements are weaker than the one needed to recover the discrete states. In the fourth stage, the discrete states estimated in the second stage are brought to a common basis by applying a novel basis transformation method which is necessary before performing output predictions to prescribed inputs. Robustness of this algorithm to amplitude bounded noise is studied and it is shown that small perturbations may only produce small deviations in the estimates that vanish as noise amplitude diminishes. Time complexity of the four-stage algorithm is also studied. A numerical example illustrates the derived results. A key role in this algorithm is played by a time-dependent switching sequence that partitions the state-space according to time, unlike many other works in the literature in which partitioning is state and/or input dependent.

Keywords: Hybrid system, switched linear system, state-space, realization, Markov parameters.
1. Introduction

Hybrid systems have attracted a vast interest lately due to their universal modeling capabilities of nonlinear dynamical systems. Switching among a finite set of linear submodels facilitated the difficult nonlinear identification problem into an identification of several affine submodels \[1, 2\] and has found its place in many versatile applications in video and texture segmentation \[3, 4, 5\], air traffic systems \[6\], human control behavior \[7\], and evolution of HIV infection \[8\].

Most works in hybrid systems literature abide to input-output representations, namely, piecewise auto-regressive exogenous (PW ARX) and switched auto-regressive exogenous (SARX) modeling techniques \[9, 10, 11, 12, 13, 14, 15\] while especially in MIMO systems, state-space models are most preferred \[16, 17\]. Controllability, observability, fault detection, and observer design concepts are also well understood for state-space models. Equivalence between input-output and state-space models in hybrid systems was introduced in \[18\] by assuming pathwise observability. An SLS is inspired by classical state-space representation based on linear time-invariant (LTI) models and is characterized by its hard switching points between its possible operating subdynamics, unlike a linear parameter-varying (LPV) system whose model parameters most often change smoothly over time \[19\]. Actually, an SLS is nothing but an LPV system with non-smooth and abrupt changes \[20\]. Equivalence between SLSs and the LPVs was formally addressed in \[21, 22\] and a power series based algorithm was proposed for the realization of the latter which is actually the Ho-Kalman algorithm \[23\] adapted to these particular model structures. Markov parameters are useful to estimate state-space matrices describing best the underlying system dynamics when the system’s pulse response is known \[24\].

State-space SLS models are an important subclass of hybrid systems and their estimation is known to be a difficult problem since neither the knowledge of the data partitioning nor the submodels (discrete states) are available \textit{a priori}. The continuous state is also unknown, which further obstructs treating it as a regression problem. Additionally, estimated state-space submodels reside in a different state basis, which hinders their use for output prediction to prescribed input signals \[25\]. A state-space realization algorithm for LPV input-output models was presented in \[26\]. Relationships between state-space models and auto-regressive exogenous (ARX) and auto-regressive moving-average exogenous (ARMAX) input-output models were explored in \[27\] by introducing observer Markov parameters initially in LTI setting and later in LTV setting \[28\]. Complexity of the SLS identification process requires a set of assumptions on dwell times of the discrete states \[29\], observability \[30\], or knowledge of the switching sequence \[31\] which can be vital for the success of a proposed algorithm.

1.1. Related work

Methods that utilize subspace techniques to detect switches, identify active submodels between switches, and merge similar submodels to estimate a switching sequence were reported in \[32, 33\]. In \[34\], input-to-state-to-output measurements were assumed to be available and discrete state parameters were estimated by a sparsity-inducing optimization approach. A novel approach in \[35\] transformed identification problem into an SARX estimation problem by modulated output injection with observer deadbeat gains and discrete state parameters were estimated.
by minimizing a sparsity-promoting norm. An online structured subspace identification method equipped with a switch detection strategy was reported in [36]. An approximate expectation-maximization method for learning maximum-likelihood parameters of a linear switching system was proposed in [37], which might get stuck to a local minimum. This method was improved in [38] by adjusting search direction. Online approach proposed in [39] alternates between Markov parameter estimation and system classification to obtain local models.

The eigensystem realization algorithm (ERA) proposed in [40] is a noisy version of the famous Ho-Kalman algorithm [41]. It uses numerically robust, but expensive singular value decomposition (SVD) and returns a minimal balanced realization of the LTI system. A modified so-called ERA/DC algorithm was presented in [42] where data correlations rather than actual response values were used. The methodology here serves to average-out the effect of noise on the estimated state-space realization under the observation that output covariance matrices are identical to Markov parameter covariance matrices when the system is excited with white noise inputs. Online version of the ERA was reported in [43]. Extented version of the ERA dealing with arbitrary-variations was presented in [44] as a time-varying ERA (LTV-ERA).

1.2. Motivation for the SLS realization

This paper is about the realization of SLSs described by a discrete convolution equation

\[ y(k) = h(k,l) \ast u(l) + e(k) \quad k = 1, \cdots, N \]  

where \( u(l) \) is the input, \( h(k,l), -\infty \leq l \leq k \) are the doubly indexed Markov parameters, \( y(k) \) and \( e(k) \) are respectively the output and noise. The collection \( h(k,l), l \leq k \leq N \) grows as \( N \) increases. The state-space description in Section 2 provides a compact representation of (1) where the state-space matrices are assumed to change at switches.

The goal of system identification is to learn a model for (1) from the input-output data and provide a convergence guarantee, which is often asymptotic in \( N \). Finite sample complexity issues have mostly been ignored in system identification literature. There is a surge of interest from the machine learning community in data-driven control and non-asymptotic analysis. In [45], sample complexity results in learning the Markov parameters from single trajectories of LTI systems were derived. On the same theme, but with an emphasis on the realization issue, further results were obtained in [46]. In contrast to LTI systems, relatively few results are available for the identification and the realization of SLSs in the state-space framework. Sparse optimization approach applied to SLS parameter estimation in [34, 35] has been a promising alternative to least-squares methods.

Realization theory for LPV systems is not complete despite many advances [21, 47, 22, 48]. The realization problem studied in this paper is an extension of a result derived in [35] to MIMO setting and draws on the early LTV realization results [49]. A key role in the realization is played by a time-dependent switching sequence that partitions the state-space according to time, unlike many other works in the literature in which partitioning is state and/or input dependent. In jump Markov linear systems, for example, switching sequence evolves according to a finite state Markov chain [50]. In a recent study [51], time-varying parameters were modeled as an output vector of a finite-dimensional linear system driven by the input-output data (1).

1.3. Organization of the paper

In Section 2, we formulate the SLS realization problem from finite Markov parameter sequences. Section 3 builds on the realization problem for MIMO-LTV systems from input-output
data studied in [49]. It is demonstrated that under mild assumptions on the dwell times of the
discrete states one can extract an LTV realization that is topologically equivalent to the true SLS
on every time interval of sufficient length.

Section 4 starts by examining stationary point set of a Hankel matrix with fixed block and
column dimensions and generated by the Markov parameters of the SLS. This is a special case of
the sum-of-norms regularization method applied to the segmentation problem in SARX models
[52] in which a parameter vector replaces the Hankel matrix. The switches of the SARX model
are estimated in [52] by minimizing a quadratic norm of mismatch error regularized by a sum
of parameter changes. On interval subsets of a stationary point set, an SLS behaves like an LTI
system. It is demonstrated that every sufficiently long segment of a switching sequence contains
an interval from the stationary point set if the discrete states satisfy a unimodality condition. This
result paves the road for the proposal of a discrete state estimation algorithm based on clustering.
Features used for clustering and clustering algorithms are briefly discussed. A feature based on
absolute sum of modal eigenvalues used for clustering extracts all discrete states up to arbitrary
similarity transformations if each discrete state visits at least one segment and dwells long enough
there.

Sections 5 and 6 are devoted to estimation of the switching sequence. We present three
schemes. The first scheme is based on correcting the state-space matrices estimated by the LTV
realization algorithm from the Markov parameters. It starts from a discrete state estimate and
proceeds along the positive and/or the negative real axes and continues until a switch or a switch
pair is detected. This process is repeated until all switches, compatible with the dwell time re-
quirements, are detected. The second scheme is based on matching the estimated and the true
Markov parameters of the SLS system over a segment. This scheme too similarly starts from a
discrete state estimate and proceeds along the positive and/or negative real axes until a switch
or a switch pair is detected. The third scheme is also based on matching the Markov param-
eters, but it is a discrete optimization/hypothesis testing algorithm. The three switch detection
schemes operate on different dwell time and model structure requirements, but their dwell time
requirements are weaker than that one needed to recover the discrete states.

The discrete state estimates in Section 4 are only similar to the true ones. If they will be
used for predicting outputs to prescribed inputs, it is necessary to bring them into a common
basis with only one similarity transformation chosen freely. By exploiting minimality of the
discrete states and putting a mild restriction on the minimum dwell time, we present an elegant
solution to this problem in Section 7. In Section 8 we put all derivations together to form a meta-
algorithm. Sensitivity analysis of this meta-algorithm to amplitude bounded noise is performed,
more specifically it is shown that small perturbations in the Markov parameters may only lead to
small deviations in the estimates that vanish as noise amplitude diminishes. Time complexity of
each stage in the meta-algorithm is studied in detail. A numerical example in Section 9 illustrates
the derived results. Section 10 concludes the paper.

2. Problem formulation

In this paper, we consider a class of MIMO–LTV systems represented by the state-space
equations

\[
\begin{align*}
    x(k+1) &= A(k)x(k) + B(k)u(k), \\
    y(k) &= C(k)x(k) + D(k)u(k)
\end{align*}
\]
where $u(k) \in \mathbb{R}^m$, $y(k) \in \mathbb{R}^p$, and $x(k) \in \mathbb{R}^n$, $k = 1, \cdots, N$ are respectively the input, the output, and the state sequences. The state dimension $n$ is constant and assumed to be known.

Let $\mathbb{N}$ denote the set of positive integers and $\phi$ be a switching sequence, that is, a map from $\mathbb{N}$ onto a finite set $S = \{1, \cdots, \sigma\}$ with a fixed and known $\sigma \in \mathbb{N}$. Let $\mathcal{P}(k) = (A(k), B(k), C(k), D(k))$ for $k = 1, \cdots, N$. Since $\phi(k) \in S$ for all $k$, we define the collection of the discrete states by $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_\sigma\}$. The MIMO–LTV system $\mathcal{H}_\phi$ with the state-space matrices changed by $\phi$ is an SLS. The switching sequence $\phi(k)$ partitions $[1, N]$ to a collection of disjoint intervals $[1, k_1), [k_1, k_{i+1}), \cdots, [k_r, \ N]$ such that

$$\phi(k) = \begin{cases} \phi(1), & 1 \leq k < k_1 \\ \phi(k_i), & k_i \leq k < k_{i+1} \\ \phi(k_r), & k_r \leq k < N. \end{cases}$$  \hfill (4)

Denoting this segmentation by $\chi$, we define the dwell times $\delta_i(\chi)$ and the minimum dwell time $\delta_e(\chi)$ by

$$\delta_i(\chi) = k_{i+1} - k_i \quad \text{and} \quad \delta_e(\chi) = \min_{1 \leq i < r} \delta_i(\chi).$$  \hfill (5)

Thus, $\delta_i(\chi)$ is the waiting time of the discrete state active in the segment $[k_i, k_{i+1})$ and $\delta_e(\chi)$ is the smallest waiting time of all discrete states in the interval $[1, k_r)$. Set $\delta_0(\chi) = k_1 - 1$ and $\delta_r(\chi) = N - k_r$.

By introducing the Markov parameters and the state transition matrix associated with the homogeneous part of $\mathcal{H}_\phi$ defined respectively by

$$h(k, \ell) = \begin{cases} C(k)\Phi(k, \ell + 1)B(\ell), & k > \ell \\ D(k), & k = \ell \\ 0, & k < \ell, \end{cases}$$  \hfill (6)

and

$$\Phi(k, \ell) = \begin{cases} A(k-1) \cdots A(\ell), & k > \ell \\ I_n, & k = \ell \end{cases}$$  \hfill (7)

where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix, from (4), we can calculate the response of the system to any initial state $x(\ell)$, $\ell \geq 1$ and a prescribed input sequence $u(k)$, $k \geq 1$ as follows

$$y(k) = C(k)\Phi(k, \ell)x(\ell) + \sum_{j=\ell}^{k} h(k, j)u(j), \quad \ell \leq k \leq N.$$  \hfill (8)

Since the state-space matrices are changed by the switching sequence, the input-output trajectory of $\mathcal{H}_\phi$ strongly depends on $\phi$. In most works on SLSs, $\phi$ is a typically state or/and input dependent signal whereas in the current work it is an unknown signal. The realization problem we study in this paper is described as follows.

**Problem 2.1.** Given a set of doubly indexed Markov parameters $h(k, \ell)$ of the MIMO–SLS model $\mathcal{H}_\phi$ for $1 \leq k \leq N$ and $1 \leq \ell \leq k$, estimate each $\mathcal{P}_j \in \mathcal{P}$, $j = 1, \cdots, \sigma$ up to a similarity transformation and $\phi(k)$.

In the course of developing a realization algorithm that solves Problem 2.1, we will impose some conditions on the system structure and the switching sequence. We will also discuss the freedom in choosing similarity transformation.
3. LTV realization from Markov parameters

This section prepares the stage for a discrete-state estimation algorithm and three switch detection schemes presented in Sections 4, 5, 6, and 6.3. In this section, we will review the realization problem for MIMO-LTV systems from input-output data. Recall that the description of an LTV system in terms of state-space matrices is not unique: Two given realizations have the same Markov parameters if they are topologically equivalent, i.e., there exists a bounded \( T(k) \in \mathbb{R}^{n \times n} \) with bounded inverse such that for all \( k \in \mathbb{N} \),

\[
\begin{align*}
    \tilde{A}(k) &= T(k + 1)A(k)T^{-1}(k), \\
    \tilde{B}(k) &= T(k + 1)B(k), \\
    \tilde{C}(k) &= C(k)T^{-1}(k), \\
    \tilde{D}(k) &= D(k).
\end{align*}
\]

We call \( T(k) \) a Lyapunov transformation. As far as input-output behavior is concerned, it suffices to estimate the Markov parameters from input-output data.

Let \( q,r \in \mathbb{N} \) be two numbers to be fixed later. For all \( k > r \), we define the Hankel matrices by

\[
    \mathcal{H}_{q,r}(k) = \begin{bmatrix}
    h(k, k-1) & \cdots & h(k, k-r) \\
    \vdots & \ddots & \vdots \\
    \cdots & \cdots & h(k+q-1, k-r)
\end{bmatrix}
\]

which can be factorized from [49] as follows

\[
    \mathcal{H}_{q,r}(k) = \Theta_q(k) \mathcal{R}_r(k - 1)
\]

with \( \Theta_q(k) \) and \( \mathcal{R}_r(k - 1) \) denoting the extended observability and the controllability matrices defined by

\[
    \Theta_q(k) = \begin{bmatrix}
    C(k) \\
    C(k + 1) \Phi(k + 1, k) \\
    \vdots \\
    C(k + q-1) \Phi(k + q-1, k)
\end{bmatrix} \in \mathbb{R}^{q \times n}
\]

and

\[
    \mathcal{R}_r(k - 1) = [B(k - 1) \cdots \Phi(k, k - r + 1)B(k - r)].
\]

The observability and the controllability Gramians for the LTV system (2)–(3) are defined by

\[
    \mathcal{G}_o(k; q) = \Theta_q(k)^T \Theta_q(k) \] (13)

\[
    \mathcal{G}_c(k; r) = \mathcal{R}_r(k - 1) \mathcal{R}_r^T(k - 1) \] (14)

If \( A(k), B(k), C(k) \) are bounded matrices for all \( k, (A(k), B(k), C(k)) \) is said a bounded realization. A bounded realization \( (A(k), B(k), C(k)) \) is uniformly observable iff \( A(k) \) is nonsingular and \( \exists q \in \mathbb{N} \) such that \( \mathcal{G}_o(k; q) \geq \alpha_o I_n \) for some \( \alpha_o > 0 \) and \( \forall k \). This fact is a restatement of Lemma 2 in [49] with changes in the notation. Similarly, a bounded realization \( (A(k), B(k), C(k)) \) is uniformly controllable iff \( A(k) \) is nonsingular and \( \exists r \in \mathbb{N} \) such that \( \mathcal{G}_c(k; r) \geq \alpha_c I_n \) for some \( \alpha_c > 0 \) and \( \forall k \). This fact restates Lemma 1 in [49] with appropriate changes in the notation. The following definition is adapted from [49].

**Definition 3.1.** A bounded realization \( (A(k), B(k), C(k)) \) is uniform if it is uniformly observable and controllable.
For a bounded realization \((A(k), B(k), C(k))\), the definitions of uniformly observable, uniformly controllable, and uniform applies to all \(k \in \mathbb{N}\). However, our observation interval is limited to \([1, N]\). Moreover from \((11)\) and \((14)\), \(k\) must satisfy the inequalities \(k + q - 1 \leq N\) and \(k - r \geq 1\). Hence, the above definitions can only be checked for all \(k \in [r + 1, N - q + 1]\). It is our best interest to pick \(q\) and \(r\) as small as possible while \(\gamma_q(k; q)\) and \(\gamma_r(k; r)\) are positive-definite. For fixed \(k\), \(\gamma_q(k; q)\) and \(\gamma_r(k; r)\) are non-decreasing functions of \(q\) and \(r\), i.e., \(\gamma_q(k; q_1) \geq \gamma_q(k; q_2)\) iff \(q_1 \geq q_2\) and \(\gamma_r(k; r_1) \geq \gamma_r(k; r_2)\) iff \(r_1 \geq r_2\).

Observability is a critical aspect in observer design and identification algorithms for SLSs. Several observability definitions for SLSs (mode observability, strong mode observability, observability, strong observability etc.) have appeared in hybrid systems literature. Checking observability for SLSs is formidable. For example, checking the observability of \((2)-(4)\) for all possible discrete state sequences of length \(N\) can involve \(O(\sigma^N)\) rank tests. To deal with this exponential complexity, in \([53]\) switching times were assumed to be separated by a minimum dwell time that depends on \(n\). Under this assumption, computational complexity was shown to reduce from \(O(\sigma^N)\) to \(O(\sigma^2)\). The minimum dwell time is not very restrictive since rapidly changing modes may cause instability problems \([25]\).

Bounded realization assumption follows from the following.

**Assumption 3.1.** Every discrete state \(\mathcal{P}_j \in \mathcal{P}, 1 \leq j \leq \sigma\) is bounded-input/bounded-output (BIBO) stable and has a Macmillan degree \(n\).

We also impose a mild restriction on the minimum dwell time as follows.

**Assumption 3.2.** The switching sequence \(\varphi\) satisfies \(\delta_s(\chi) \geq n\).

The following result was derived in \([35]\).

**Lemma 3.1.** Suppose Assumptions 3.1, 3.2 hold and \(q, r \geq 2n\). Then, the MIMO-SLS model \((2)-(4)\) is uniform on \([2n + 1, N - 2n + 1]\).}

This lemma enables one to extract an LTV realization which is topologically equivalent to \((2)-(4)\) from the pairs \(\{H_{R_q}(k), H_{R_r}(k + 1)\}, k > r\). In fact, uniform realization property implies that \(H_q\) and \(\bar{R}_r\) are full-rank matrices and the relationship rank \(\text{rank}(H_{R_q}(k)) = n\) holds from an application of the Sylvester inequality \([54]\) to the product in \((10)\). The extended observability and controllability matrices are then determined up to similarity transformations from the SVDs of \(H_{R_q}(k)\) and \(H_{R_r}(k + 1)\). In fact, let \(J_\uparrow\) and \(J_\downarrow\) denote the shift matrices of block-row up and down and \(J_\leftarrow\) and \(J_\rightarrow\) denote the block-column left and right shift matrices, respectively, defined by

\begin{align}
J_\uparrow &= \begin{bmatrix} 0_{(q-1)p \times p} & I_{(q-1)p} \end{bmatrix}, & J_\downarrow &= \begin{bmatrix} I_{(q-1)p} & 0_{(q-1)p \times p} \end{bmatrix}, \\
J_\leftarrow &= \begin{bmatrix} 0_{m \times (r-1)m} & I_{(r-1)m} \end{bmatrix}, & J_\rightarrow &= \begin{bmatrix} I_{(r-1)m} & 0_{m \times (r-1)m} \end{bmatrix}
\end{align}

where \(0_{m \times n}\) denotes the \(m\) by \(n\) matrix of zeros. Then, from the factorization formula \((10)\) with \(k\) and \(k + 1\) plugged in and \((15)-(16)\) we derive the following relations

\begin{align}
J_\uparrow H_q(k) &= (J_\downarrow H_{q+1}(k + 1)J_\uparrow)H_q(k) = H_{q-1}(k + 1)A(k) \quad (17) \\
\bar{R}_r(k)J_\leftarrow &= A(k)(\bar{R}_r(k - 1)J_\leftarrow) = A(k)\bar{R}_{r-1}(k - 1). \quad (18)
\end{align}
Hence, from (17) and/or (18) depending on if \( q > 2n \) and/or \( r > 2n \), we retrieve \( A(k) \) from either/both of the formulas
\[
A(k) = O^{-1}_{q-1}(k+1)J_tO_q(k) = \mathcal{R}_r(k)J_r\mathcal{R}^\dagger_{r-1}(k-1)
\]  
(19)
where \( X^T \) denotes the unique left or right-inverse of a given full-rank matrix \( X \) defined as \((X^T X)^{-1}X^T\) or \(X^T (XX^T)^{-1}\) depending on the context. We fix \( q \) and \( r \) as \( q = 2n+1 \) and \( r = 2n \). Then, \( k > 2n \). For an upper bound on \( k \), the maximum block-row index of \( \mathcal{H}_{q,r}(k+1) \) must satisfy \( k+q \leq N-2n+1 \) or \( k \leq N-4n \). Hence, \( k' \leq k \leq k'' \) where \( k' = 2n+1 \) and \( k'' = N-4n \).

It remains to estimate \( O_q(k) \) and \( O_q(k+1) \). We apply SVD to \( \mathcal{H}_{q,r}(k) \) and \( \mathcal{H}_{q,r}(k+1) \)
\[
\mathcal{H}_{q,r}(k) = U_q(k)\Sigma(k)V^T(k), \\
\mathcal{H}_{q,r}(k+1) = U_q(k+1)\Sigma(k+1)V^T(k+1)
\]  
and let
\[
\hat{O}_q(k) = U_q(k)\Sigma^{1/2}(k), \quad \hat{\mathcal{R}}_r(k-1) = \Sigma^{1/2}(k)V^T(k),
\]  
(21)
\[
\hat{O}_q(k+1) = U_q(k+1)\Sigma^{1/2}(k+1), \quad \hat{\mathcal{R}}_r(k) = \Sigma^{1/2}(k+1)V^T(k+1).
\]  
(22)
Up to similarity transformations \( T(k) \) and \( T(k+1) \), \( \hat{O}_q(k) \) and \( \hat{O}_q(k+1) \) provide estimates of \( O_q(k) \) and \( O_q(k+1) \), i.e., \( O_q(k) = \hat{O}_q(k)T(k) \) and \( O_q(k+1) = \hat{O}_q(k+1)T(k+1) \). Let
\[
\hat{A}(k) = (J_r\hat{O}_q(k+1))\dagger J_r\hat{O}_q(k)
\]  
(23)
so that
\[
\hat{A}(k) = T(k+1)A(k)T^{-1}(k).
\]  
(24)
The estimation of \( B(k) \) and \( C(k) \) is in order. To this end, let
\[
J_C = [I_p \quad O_{p \times (q-1)p}], \quad J_B = \begin{bmatrix} I_m \\ 0_{(r-1)m \times m} \end{bmatrix}.
\]  
(25)
Then, from (11) and (12)
\[
C(k) = J_C\hat{O}_q(k), \quad B(k) = \hat{\mathcal{R}}_r(k)J_B.
\]  
(26)
As the estimates of \( C(k) \) and \( B(k) \), we set
\[
\hat{C}(k) = J_C\hat{O}_q(k), \quad \hat{B}(k) = \hat{\mathcal{R}}_r(k)J_B.
\]  
(27)
Then, from \( \hat{\mathcal{R}}_r(k) = T(k+1)\hat{\mathcal{R}}_r(k) \)
\[
\hat{C}(k) = C(k)T^{-1}(k), \quad \hat{B}(k) = T(k+1)B(k).
\]  
(28)
Set \( \hat{D}(k) = h(k,k) \). Then, from (24) and (28) it follows that \( \hat{\mathcal{D}}(k) = (\hat{A}(k),\hat{B}(k),\hat{C}(k),\hat{D}(k)) \) is topologically equivalent to \( \mathcal{D}(k) \) on the interval \([k',k'']\].

If \( T(k) = T(k+1) \), \( \hat{A}(k) \) is similar to \( A(k) \) denoted by the notation \( \hat{A}(k) \sim A(k) \). It is the case when \( H_{q,r}(k) \) equals \( H_{q,r}(k+1) \). For an LTI system, i.e., \( \sigma = 1 \) in Assumption 3.1, \( \hat{A}(k) \sim A(k) \). The derivations in this section are outlined in Appendix as a subspace-based realization algorithm. The realization returned by Algorithm 1 is topologically equivalent to (2)–(4). The
quadruples returned are not similar to $\mathcal{P}(k)$, yet match the Markov parameters. Matching the discrete states will be achieved by modifying the state-space matrices as described in Section 5.

The LTV realization problem was studied in [28] from a rigid body dynamics perspective. The scheme presented above is based on single input-output data batches. If an ensemble of input-output data are available, state-space models can be identified without constraints on time-variations of the Markov parameters [53] [66]. The derivations above are summarized in the following result.

**Theorem 3.1.** Consider Algorithm 1 with the noiseless Markov parameters of [2]–[4] satisfying Assumptions 3.1–3.2. Then, Algorithm 1 returns a realization topologically equivalent to (3)–(4) on $[k' \ k'']$. 

Notice that $\mathcal{H}_{q,r}(k+1)$ is formed from the $4n$ Markov parameters and a submatrix of $\mathcal{H}_{q,r}(k)$. In fact, partition $\mathcal{H}_{q,r}(k)$ as follows

$$
\mathcal{H}_{q,r}(k) = \begin{bmatrix}
   h(k,k-1) & \cdots & h(k,k-r) \\
   \mathcal{H}_{q,r}'(k) & \vdots \\
   h(k+q-1,k-r)
\end{bmatrix}
$$

and observe that

$$
\mathcal{H}_{q,r}(k+1) = \begin{bmatrix}
   h(k+1,k) \\
   \vdots \\
   h(k+q,k) & \cdots & h(k+q,k-r+1)
\end{bmatrix}.
$$

Algorithm 1 uses $qr + (q + r - 1)(k'' - k' + 1)$ Markov parameters if it computes $\hat{\mathcal{P}}(k)$ for all $k \in [k' \ k'']$ and if it computes for one $k$, $qr + q + r - 1$ Markov parameters. Plug $q = 2n + 1$ and $r = 2n$ in: $(4N - 20n + 2)n \approx 4nn$ and $4n^2 + 6n \approx 4n^2$. If $\hat{\mathcal{P}}(k)$ is computed at every $2n + 1$th sample, $(4n^2 + 6n) (k'' - k' + 1)/(2n + 1) \approx 2Nn$ Markov parameters roughly will be used since $\mathcal{H}_{2n+1,n}(k)$ and $\mathcal{H}_{2n+1,n}(k+2n)$ share no common entries. Hence, more than 50% of the Markov parameters need to be stored if calculations are done more frequently than $O(n)$. The algorithms developed in the sequel will need all Markov parameters to be stored. Time complexity of Algorithm 1, however, depends on the frequency of calculations as the following analysis demonstrates.

### 3.1. Time complexity of Algorithm 1

In Algorithm 1, calculation of the SVDs in [20] is computationally the most expensive step without exploiting the block matrix structure and the nesting property in [29]–[30]. According to [57], the best algorithms for SVD computation of an $M \times L$ matrix have time complexity $O(c_1 M^2 L + c_2 L^3)$ where $c_1, c_2 > 0$ are absolute constants which are 4 and 22 for an algorithm called R-SVD. Recall that $q = 2n + 1$ and $r = 2n$, from $M = qp$ and $L = rm$. Each SVD in [20] has then a time complexity $O(poly(n))$ where $poly(n)$ is a third order polynomial in $n$. Pseudo-inverse of an $M \times L$ ($M > L$) matrix calculated by SVD has complexity $O(LM^2)$. Hence, with $M = (q - 1)p$ and $L = n$, $(J_z \hat{\mathcal{O}}_q(k+1))^\dagger$ in [23] has complexity $O(n^3)$. Multiplication of two matrices $X \in \mathbb{R}^{M \times K}$ and $Y \in \mathbb{R}^{K \times L}$ has complexity $O(MKL)$ when calculated directly. Hence, with $M = n$, $K = (q - 1)p$, $L = n$ we get $O(poly(n))$ for the complexity of multiplication $(J_z \hat{\mathcal{O}}_q(k+1))^\dagger$ times $J_z \hat{\mathcal{O}}_q(k)$. Adding up we get $O(poly(n))$ for the complexity of $\hat{\mathcal{A}}(k)$. Since
$J_B, J_C, \hat{J}_C, \hat{J}_q(k)$ are $n \times rm, rm \times m, p \times q_p, q_p \times n$ matrices respectively, applying the complexity result to the matrix multiplications in (27) we derive $O(poly(n))$ for the complexities of $B$ and $C$ where $poly(n)$ is a second order polynomial in $n$. It follows that $\hat{J}(k)$ has complexity $O(poly(n))$ where $poly(n)$ is a third order polynomial. Denote the number of the elements in $\kappa$ by $|\kappa|$. Then, Algorithm 1 has time complexity $O(|\kappa|poly(n))$. Hence, Algorithm 1 subject to Assumption 3.2 has a worst-case time complexity $O(Npoly(n))$ where $poly(n)$ is a second order polynomial since $|\mathcal{H}| = O(Nn^{-1})$ for evenly distributed switches. If $\mathcal{H}$ is a singleton, then time complexity of Algorithm 1 will be $O(poly(n))$ with $poly(n)$ a third order polynomial in $n$. These calculations took into consideration only $n$ and $N$ since $p$ and $m$ are fixed small numbers. There is also a similar concept called space complexity which is usually weaker than time complexity. In the setting of this paper, for example, the SVD has space complexity $O(n^2)$.

3.2. Robustness of Algorithm 1

Suppose that the Markov parameters in (6) are corrupted by ‘unknown-but-bounded’ type noise

$$\tilde{h}(k, \ell) = h(k, \ell) + e(k, \ell), \; \ell \leq k \; \text{and} \; k' \leq k \leq k''$$

(31)

where $\|e(k, \ell)\|_F \leq \varepsilon$ with $\| \cdot \|_F$ denoting the Frobenius norm of a given matrix defined as the square root of the sum of its squared elements. Our goal is to study how this noise affects the estimates $\tilde{\mathcal{H}}(k), k \in [k' \; k'']$ for small $\varepsilon > 0$.

Let $\mathcal{H}_{2n+1,2n}(k)$ be the Hankel matrix obtained by populating $\mathcal{H}_{2n+1,2n}(k)$ in (30) with $\tilde{h}(k, \ell)$. Split $\mathcal{H}_{2n+1,2n}(k)$

$$\mathcal{H}_{2n+1,2n}(k) = \mathcal{H}_{2n+1,2n}(k) + E(k).$$

(32)

Robustness of Algorithm 1 to unknown-but-bounded type noise follows from the following result.

**Theorem 3.2.** Consider (2)–(4) with the noisy Markov parameters in (31). Suppose that Assumptions 3.1–3.2 hold. Let $\tilde{\mathcal{H}}(k) = (A(k), B(k), C(k), D(k))$ and $\tilde{\mathcal{H}}(k) = (\bar{A}(k), \bar{B}(k), \bar{C}(k), \bar{D}(k))$ denote the models estimated by Algorithm 1 in the noisy and noiseless cases. Then, there exists a realization $\tilde{\mathcal{H}}^\varepsilon(k) = (A^\varepsilon(k), B^\varepsilon(k), C^\varepsilon(k), D^\varepsilon(k))$ that is topologically equivalent to $\tilde{\mathcal{H}}(k)$ satisfying $\|A^\varepsilon(k) - A(k)\|_F \leq c\varepsilon$, $\|B^\varepsilon(k) - B(k)\|_F \leq c\varepsilon$, $\|C^\varepsilon(k) - C(k)\|_F \leq c\varepsilon$, $\|D^\varepsilon(k) - D(k)\|_F \leq c\varepsilon$ for all $\varepsilon \leq \varepsilon_\varepsilon$ and some $c, \varepsilon_\varepsilon > 0$. The Lyapunov transformation $S(k)$ mapping $\tilde{\mathcal{H}}(k)$ to $\tilde{\mathcal{H}}^\varepsilon(k)$ is asymptotically orthogonal, that is, $S^T(k)S(k) \rightarrow I_n$ as $\varepsilon \rightarrow 0$.

**Proof.** Our proof is based on perturbation techniques employed in subspace identification, but far more difficult due to the fact Lyapunov transformations replace time-invariant similarity transformations in the current setup. See, Lemma 4 in Section III.A of [58]. Due to noise not only the singular values of $\mathcal{H}_{2n+1,2n}(k)$, but also its singular subspaces will be perturbed which will cause changes in the state-space matrices. The perturbation term in (32) is bounded in the Frobenius norm as follows

$$\|E(k)\|_F \leq \sqrt{2n(2n+1)}\varepsilon < 4n\varepsilon.$$  

In our perturbation analysis, we assume that $\varepsilon$ can be made as small as we are pleased. Write the first SVD in (20)

$$\mathcal{H}_{2n+1,2n}(k) = [U_{2n+1}(k) \; U_{\varepsilon}(k)] [\begin{array}{c|c} \Sigma(k) & 0 \\ \hline 0 & 0 \end{array}] [\begin{array}{c} V_{2p}(k) \\ V_{\varepsilon}(k) \end{array}]$$

(33)
by completing the singular subspaces with $U_ε(k)$ and $V_ε(k)$. Let

$$
\begin{bmatrix}
U_{2n+1}^T(k) \\
U'_v(k)
\end{bmatrix}
E(k) \begin{bmatrix}
V_{2n}(k) \\
V'_v(k)
\end{bmatrix} =
\begin{bmatrix}
X_{11}(k) & X_{12}(k) \\
X_{21}(k) & X_{22}(k)
\end{bmatrix} = X(k).
$$

(34)

From (34), notice that $\|X(k)\|_F = \|E(k)\|_F < 4n\epsilon$. Moreover, $\|X_a(k)\|_2 \leq \|X_a(k)\|_F, 1 \leq s,t \leq 2$ since the Euclidean norm is dominated by the Frobenius norm. Let $\sigma_j(\cdot)$ denote the $j$th largest singular value of $H_{2n+1,2n}(k)$. Then, with $\tilde{A}$ such that

$$
\tilde{A} = \sigma_1(k) - \|X_{11}(k)\|_2 - \|X_{22}(k)\|_2 > \sigma_4(k) - 8n\epsilon.
$$

(35)

Choose $\epsilon_0 > 0$ such that $\sigma_n(k) > 16n\epsilon_0$ for all $k$. This is possible since there are only $\sigma$ discrete states. Then, from (35) we get $\tilde{A} > \frac{1}{2} \sigma_4(k)$ for all $\epsilon \leq \epsilon_0$. We also have

$$
\frac{\|X_{21}(k)\|_F}{\sigma_4(k)} < \frac{4n\epsilon}{\frac{1}{2} \sigma_4(k)} < \frac{1}{2}, \quad \forall \epsilon \leq \epsilon_0.
$$

(36)

Running Algorithm 1 with the corrupted Markov parameters (31) accounts for replacing the first SVD in (20) by

$$
\hat{H}_{2n+1,2n}(k) = \begin{bmatrix}
O_{2n+1}(k) & \hat{U}_v(k)
\end{bmatrix}
\begin{bmatrix}
\mathbf{\hat{S}}_v(k) & 0 \\
0 & \mathbf{\hat{S}}_v(k)
\end{bmatrix}
\begin{bmatrix}
\hat{V}_v^T(k) \\
\hat{V}_v(k)
\end{bmatrix}.
$$

(37)

Simply plug $k + 1$ in (37) for the second SVD in (20). Now, the chain of inequalities (36) implies that there exist matrices $P_v(k)$ and $Q_v(k)$ satisfying

$$
\left\| \begin{bmatrix}
Q_v(k) \\
P_v(k)
\end{bmatrix}\right\|_F \leq \frac{2\epsilon}{\delta}
$$

(38)

such that range $(V_{2n}(k) + V_v(k)Q_v(k))$ and range $(U_{2n+1}(k) + U_v(k)P_v(k))$ are a pair of singular subspaces for the perturbed Hankel matrix $H_{2n+1,2n}(k)$. See, for example Theorem 8.3.5 in (57). Since the range spaces are equal and $U_{2n+1}(k)$ is of full rank, there exists a unique nonsingular matrix $T(k)$ such that

$$
\hat{U}_{2n+1}(k) = (U_{2n+1}(k) + U_v(k)P_v(k))T(k).
$$

(39)

A similar expression holds for $\hat{U}_{2n+1}(k) + 1$ by plugging $k + 1$ in (39). Let $\hat{\sigma}_{2n+1}(k) = \hat{U}_{2n+1}(k)\Sigma_{2n+1}(k)$. Then, with $\hat{A}_v(k) = U_v(k)P_v(k)\Sigma_{2n+1}(k)$ and $S(k) = \Sigma_{2n+1}(k)T(k)\Sigma_{2n+1}(k)$, from $\hat{\sigma}_{2n+1}(k) = U_{2n+1}(k)\Sigma_{2n+1}(k)$ and (39) we get $\hat{\sigma}_{2n+1}(k) = (\hat{\sigma}_{2n+1}(k) + \hat{A}_v(k))S(k)$. Thus,

$$
\begin{align*}
\hat{A}(k) & \triangleq \left(J_1 \hat{\sigma}_{2n+1}(k+1)\right)^\dagger J_1 \hat{\sigma}_{2n+1}(k) \\
& = S^{-1}(k+1) \left(J_1 \left(\hat{\sigma}_{2n+1}(k+1) + \hat{\sigma}_v(k+1)\right)\right)^\dagger J_1 \left(\hat{\sigma}_{2n+1}(k) + \hat{\sigma}_v(k)\right) S(k)
\end{align*}
$$

and therefore

$$
S(k+1)\hat{A}(k)S^{-1}(k) = \left(J_1 \left(\hat{\sigma}_{2n+1}(k+1) + \hat{\sigma}_v(k+1)\right)\right)^\dagger J_1 \left(\hat{\sigma}_{2n+1}(k) + \hat{\sigma}_v(k)\right).
$$

(40)

The right-hand side of (40) is the least-squares solution of the inconsistent equations

$$
(J_1 \hat{\sigma}_{2n+1}(k+1) + J_1 \hat{\sigma}_v(k+1))A^T(k) = J_1 \hat{\sigma}_{2n+1}(k) + J_1 \hat{\sigma}_v(k).
$$

(41)
The perturbations on the left and right-hand sides of (31) are bounded from (38) as follows
\[ \| J_1 \tilde{\Delta}_v(k+1) \|_F \leq \| \tilde{\Delta}_v(k+1) \|_F = \| P_e(k) \Sigma^{1/2}(k) \|_F \leq \sigma_1(k) \| P_e(k) \|_F \leq 2 \frac{\sigma_1(k)}{\delta} \varepsilon \leq 4\mu_{\Sigma^{1/2}(k)} \varepsilon \]
and \[ \| J_1 \tilde{\Delta}_v(k) \|_F \leq 4\mu_{\Sigma^{1/2}(k)} \varepsilon \] where \( \mu_{\Sigma^{1/2}(k)} \) denotes the condition number of \( \Sigma^{1/2}(k) \).

If \( \varepsilon = 0 \), then \( A^\circ(k) \) equals to \( \hat{A}(k) \) in (23) and if \( \varepsilon \) is small, say \( \varepsilon \leq \varepsilon_1, \) \( A^\circ(k) \) satisfies \( \| A^\circ(k) - \hat{A}(k) \|_F \leq c_1 \varepsilon \) for some \( c_1 > 0 \). See, for example Theorem 5.3.1 in [57]. Let \( \varepsilon_2 = \min\{\varepsilon_0, \varepsilon_1\} \). Then, we have shown that
\[ \| S(k+1)\tilde{A}(k)S^{-1}(k) - \hat{A}(k) \|_F \leq c_1 \varepsilon, \quad \forall \varepsilon \leq \varepsilon_2. \] (42)

The rest of the perturbed state-space matrices are calculated similarly starting with \( \tilde{C}(k) \) as follows
\[ \tilde{C}(k) \overset{\Delta}{=} J_C \hat{\theta}_{2n+1}(k) = J_C \left( \hat{\theta}_{2n+1}(k) + \tilde{\Delta}_v(k) \right) S(k). \]
Thus,
\[ \| \tilde{C}(k)S^{-1}(k) - \hat{C}(k) \|_F \leq c_2 \varepsilon, \quad \forall \varepsilon \leq \varepsilon_2. \] (43)

Let \( \hat{D}(k) = \hat{h}(k,k) \),
\[ \| \hat{D}(k) - \hat{D}(k) \|_F = \| e(k,k) \|_F \leq \varepsilon. \] (44)

Let \( \hat{B}(k-1) = \hat{\theta}(k-1)J_B \). From (37), we first derive an expression for \( \hat{V}^T_{2n}(k) \) as follows
\[ \hat{V}^T_{2n}(k) = \tilde{\Sigma}^{-1}(k)\tilde{U}^T_{2n+1}(k)\tilde{\mathcal{H}}_{2n+2n+1}(k). \] (45)

Then, from the first equation in (22)
\[ \hat{B}(k-1) = \Sigma^{1/2}(k)\hat{V}^T_{2n}(k)J_B = \Sigma^{-1/2}(k)\tilde{U}^T_{2n+1}(k)\tilde{\mathcal{H}}_{2n+2n+1}(k)J_B = \tilde{\Sigma}^{-1/2}(k)\tilde{U}^T_{2n+1}(k)\tilde{\mathcal{H}}_{2n+2n+1}(k)J_B + \tilde{\Sigma}^{-1/2}(k)\tilde{U}^T_{2n+1}(k)E(k)J_B \] (46)

where we used (32). Let \( \sigma_j \) denote the \( j \)th largest singular value of \( \tilde{\mathcal{H}}_{2n+2n+1}(k) \). We bound the second term in (46)
\[ \| \tilde{\Sigma}^{-1/2}(k)\tilde{U}^T_{2n+1}(k)E(k)J_B \|_F \leq \| \tilde{\Sigma}^{-1/2}(k)\tilde{U}^T_{2n+1}(k)E(k) \|_F \leq \| \tilde{\Sigma}^{-1/2}(k) \| \| \tilde{U}^T_{2n+1}E(k) \|_F \leq \tilde{\sigma}_n^{-1}(k) \| E(k) \|_F \leq 4n\tilde{\sigma}_n^{-1}(k) \varepsilon \] (47)

where \( \| \cdot \| \) denotes the spectral norm. From Corollary 8.3.2 in [57], note that
\[ \| \sigma_n(k) - \sigma_n(k) \| \leq \| E(k) \| < 4n \varepsilon. \] (48)

Hence, if \( \varepsilon \leq \varepsilon_2 \) from \( \sigma_n(k) > 16n \varepsilon_0 \)
\[ \sigma_n(k) \geq \sigma_n(k) - 4n \varepsilon \geq \frac{3}{4} \sigma_n(k). \] (49)

Thus, from (47) and (39) if \( \varepsilon \leq \varepsilon_2 \)
\[ \| \tilde{\Sigma}^{-1/2}(k)\tilde{U}^T_{2n+1}E(k)J_B \|_F < 6n\tilde{\sigma}_n^{-1}(k) \varepsilon \] (50)
Split the first term in the second equation of (46) as
\[
\tilde{\Sigma}^{-1/2}(k) U^T_{2n+1}(k) \mathcal{H}_{2n+1,2m}(k) J_B = \tilde{\Sigma}^{-1/2}(k) \tilde{T}^T(k) U^T_{2n+1}(k) \mathcal{H}_{2n+1,2m}(k) J_B \\
+ \tilde{\Sigma}^{-1/2}(k) \tilde{T}^T(k) \Sigma^{-1/2}(k) \tilde{A}_T(k) \mathcal{H}_{2n+1,2m}(k+1) J_B \\
= \tilde{\Sigma}^{-1/2}(k) \tilde{T}^T(k) \Sigma^{1/2}(k) \tilde{B}(k-1) + \tilde{\Sigma}^{-1}(k) \tilde{T}^T(k) \tilde{A}_T(k) \mathcal{H}_{2n+1,2m}(k+1) J_B.
\]

If \( \varepsilon \leq \varepsilon_2 \), the second term above may be bounded from (49) and (38) as follows
\[
\| \tilde{\Sigma}^{-1}(k) \tilde{T}^T(k) \tilde{A}_T(k) \mathcal{H}_{2n+1,2m}(k+1) J_B \|_F \leq \frac{\sigma_n^{-1}(k) \mathcal{H}_{2n+1,2m}(k) \| J_B \| \| \tilde{A}_T(k) S(k) \|_F}{\sigma_n(k)} \leq \frac{4}{3} \mu_2^{1/2}(k) \sigma_1(k) \| P_c(k) \|_F \| S(k) \| \\
\leq \frac{4}{3} \mu_2^{1/2}(k) \sigma_1(k) \frac{4 \varepsilon}{\sigma_n(k)} \| S(k) \| < 6 \mu_2^{1/2}(k) \| S(k) \| \varepsilon. \tag{51}
\]

It remains to bound \( \| S(k) \| \). If \( \varepsilon \leq \varepsilon_2 \), from the definition of \( S(k) \) and (48)
\[
\| S(k) \| \leq \sigma_n^{-1}(k) \sigma_1(k) \| \tilde{T}(k) \| \leq \frac{\sigma_1(k) + 4 \varepsilon}{\sigma_n(k)} \| \tilde{T}(k) \| < \frac{\sigma_1(k) + 4 \varepsilon}{\sigma_n(k)} \| \tilde{T}(k) \| < 2 \mu_2^{1/2}(k) \| \tilde{T}(k) \|. \tag{52}
\]

Next, we derive a bound on \( \| \tilde{T}(k) \| \). To this end, we multiply the left and right-hand sides of (53) from left with \( \tilde{U}^T_{2n+1}(k) \) and recall that \( U_{2n+1}(k), U_{2m+1}(k), U_c(k) \) are unitary matrices and \( U_{2n+1}(k) \perp U_c(k) \). Thus,
\[
I_n - \tilde{T}^T(k) \tilde{T}(k) = \tilde{T}^T(k) P_c^T(k) P_c(k) \tilde{T}(k).
\tag{53}
\]

Since the right-hand side of (53) is positive semi-definite, \( \| \tilde{T}(k) \| \leq 1 \). Then, from (51) and (52)
\[
\| \tilde{\Sigma}^{-1}(k) \tilde{T}^T(k) \tilde{A}_T(k) \mathcal{H}_{2n+1,2m}(k+1) J_B \|_F < 12 \mu_2^{1/2}(k) \varepsilon. \tag{54}
\]

In the last step, we perform a splitting again as follows
\[
\tilde{\Sigma}^{-1/2}(k) \tilde{T}^T(k) \Sigma^{1/2}(k) \tilde{B}(k-1) = \tilde{\Sigma}^{-1/2}(k) \tilde{T}^{-1}(k) \Sigma^{1/2}(k) \tilde{B}(k-1) \\
+ \tilde{\Sigma}^{-1/2}(k) \left( \tilde{T}^T(k) \tilde{T}(k) - I_n \right) \tilde{T}^{-1}(k) \Sigma^{1/2}(k) \tilde{B}(k-1) \tag{55}
\]

where we recognize the relation \( \tilde{\Sigma}^{-1/2}(k) \tilde{T}^{-1}(k) \Sigma^{1/2}(k) \tilde{B}(k-1) = S^{-1}(k) \tilde{B}(k-1) \). We will bound now the second term in (55). To this end, multiplying (53) from left by \( \tilde{T}^{-T}(k) \) and from right by \( \tilde{T}^{-1}(k) \) we get
\[
\tilde{T}^{-T}(k) \tilde{T}(k) = I_n + P_c^T(k) P_c(k).
\]

Then, if \( \varepsilon \leq \varepsilon_2 \) from (38)
\[
\| \tilde{T}^{-1}(k) \|^2 \leq \| I_n + P_c^T(k) P_c(k) \| \leq \| I_n \| + \| P_c(k) \|^2 < 1 + 4 \sigma_n^{-1}(k) \varepsilon < 1 + \frac{\varepsilon}{4n \varepsilon_0} \leq \frac{5}{4}. \tag{56}
\]

Next, from (53)
\[
\| \tilde{T}(k) \tilde{T}(k) - I_n \| \leq \| \tilde{T}(k) \| \| \tilde{T}(k) \| < \frac{16 \varepsilon^2}{\sigma_n^2(k)}. \tag{57}
\]
Since the derived bounds are uniform in $k$, let

$$
\| \hat{\Sigma}^{-1/2}(k) (\tilde{T}(k) \tilde{T}(k) - I_n) \tilde{T}^{-1}(k) \Sigma^{1/2}(k) \hat{B}(k-1) \|_F \leq \frac{\sigma_1(k)}{\sigma_n(k)} \| \tilde{T}^T(k) \tilde{T}^{-1}(k) - I_n \|_F \| \tilde{T}^{-1}(k) \|_F \| \hat{B}(k-1) \|_F
$$

(58)

Since $\tilde{\mathcal{H}}(k)$ is topologically equivalent to $\mathcal{H}(k)$, the Lyapunov transformation mapping $\mathcal{H}(k)$ to $\tilde{\mathcal{H}}(k)$ is uniformly bounded on compact time intervals. Since $\hat{B}(k)$ is bounded for all $k$, $\| \hat{B}(k-1) \|_F$ will be bounded above by some finite $\gamma$ on $[k', k'']$. Choose $0 < \varepsilon_3 \leq \varepsilon_2$ satisfying $\varepsilon_3 \gamma < 1$. Then, from (58) if $\varepsilon \leq \varepsilon_3$

$$
\| \hat{\Sigma}^{-1/2}(k) (\tilde{T}(k) \tilde{T}^{-1}(k) - I_n) \tilde{T}^{-1}(k) \Sigma^{1/2}(k) \hat{B}(k-1) \|_F \leq \frac{24}{\sigma_n^2(k)} \varepsilon.
$$

(59)

Let $c'_3 = 6\sigma_n^{-1}(k) + 12\sigma_n^3 + 24\sigma_n^{-2}(k)$. Then, from (46), (50), (54), and (59) if $\varepsilon \leq \varepsilon_3$

$$
\| \hat{B}(k-1) - S^{-1}(k) \hat{B}(k-1) \|_F \leq c'_3 \varepsilon.
$$

(60)

Since the derived bounds are uniform in $k$, we may substitute $k + 1$ in place of $k$. Furthermore, as $\varepsilon \to 0$

$$
S^T(k)S(k) = \hat{\Sigma}^{1/2}(k) \tilde{T}^T(k) \tilde{T}(k) \Sigma^{-1/2}(k) \to I_n.
$$

Thus, $\| S^{-1}(k) \| \to 1$ as $\varepsilon \to 0$ and if necessary reducing $\varepsilon_3$ to $\varepsilon_1 > 0$ and increasing $c'_3$ to $c_3 < \infty$ we derive

$$
\| S(k+1) \hat{B}(k) - \hat{B}(k) \|_F \leq c_3 \varepsilon
$$

(61)

if $\varepsilon \leq \varepsilon_3$ by reorganizing (60) and plugging $k + 1$ in. Let $c = \max\{c_1, c_2, c_3\}$ and note that $S(k)$ is the sought Lyapunov transformation. We took a long path to derive (61). If we had taken a shorter path by considering the perturbed range space of $V_{2n}(k)$, this would introduce another matrix $\tilde{T}'(k)$ without leading to a Lyapunov transformation.

A close examination of the proof shows that $S(k)$ has some uniqueness properties up to $n$ sign changes, but this property will not be needed in the subsequent analyses.

4. Discrete state estimation

Algorithm 1 delivers a time-varying model that is only topologically equivalent to (2)–(3) on a given subset of $[k', k'']$. Further processing of this model is necessary to reveal the discrete states and the switching sequence. If $\varphi$ is an arbitrary signal, there is no hope to recover $\mathcal{P} \varphi$ and $\varphi$ from the LTV model since $T(k)$ may possibly be a completely arbitrary time-varying matrix. When the dwell times of (2)–(3) are sufficiently large, it is possible to devise algorithms to estimate $\mathcal{P}$ and $\varphi$ from the LTV model as will be demonstrated in this section and Sections 5, 6, 6.3.

We start by taking the differences of (29) and (30)

$$
\delta_{\mathcal{H}}(k) = \mathcal{H}_{q,r}(k+1) - \mathcal{H}_{q,r}(k), \quad k \in [k', k'']
$$

(62)

and define $\mathcal{Z}_{\mathcal{H}}$ as the set of the zeros of the first order difference function in (62) as follows

$$
\mathcal{Z}_{\mathcal{H}} = \{ k \in [k', k''] : \delta_{\mathcal{H}}(k) = 0 \}.
$$

(63)
Recall that we fixed $q = 2n + 1$ and $r = 2n$. We would like $\mathbb{Z}_{X'}$ to have a resemblance to $\chi$, differing only by a few points around switches on every segment or at least on long segments. It is also desired that each sufficiently long interval in $\mathbb{Z}_{X'}$ must reside only in one segment. We will show that both aims are feasible if we put some mild restrictions on the dwell times and the model structure.

Let us choose the $\ell_1$-norm of eigenvalues of square matrices defined by

$$\mathcal{M}(X) = \sum_{i=1}^{n} |\lambda_i(X)|, \quad X \in \mathbb{R}^{n \times n}. \quad (64)$$

as the feature used for clustering. We will elaborate on this feature and clustering later, but for the time being note that it is invariant to matrix similarity transformations as can be verified easily. We assume that $\mathcal{M}$ separates the discrete states in $\mathcal{P}$ as described in the following assumption.

**Assumption 4.1.** The discrete states of the MIMO–SLS model (2)–(4) satisfy

$$\mathcal{M}(A(k)) \neq \mathcal{M}(A(l)) \iff \phi(k) \neq \phi(l). \quad (65)$$

We derive the following result when Assumption 4.1 holds and the dwell times satisfy mild conditions.

**Lemma 4.1.** Consider the MIMO–SLS model (2)–(4). Suppose that Assumption 4.1 holds and $\delta_0(\chi) \geq 4n + 2$, $\delta_0(\chi) > 8n$, $\delta_0(\chi) \geq 6n + 2$. Let $\mathbb{Z}_{X'}$ be as in (63). Then, $\{k_i + 1, k_{i+1}\} \notin \mathbb{Z}_{X'}$ and there exists a collection of closed intervals $S_i \subset \mathbb{Z}_{X'}$, $0 \leq i \leq i'$ such that

$$[k' + 2n_1 k_1 - 2] \subseteq S_0 \subset [k' k_1), \quad (66)$$

$$[k_1 + 2n_1 k_{i+1} - 2] \subseteq S_i \subset [k_i k_{i+1}), \quad 0 < i < i', \quad (67)$$

$$[k_1' + 2n_1 k_{i'} - 2n - 1] \subseteq S_{i'} \subset [k_{i'} k_{i''}]. \quad (68)$$

The closed intervals $S_i$, $0 \leq i \leq i'$ are maximal, i.e., any closed interval $\tilde{S}_i \subset \mathbb{Z}_{X'}$ satisfying one chain of the inequalities in (66)–(68) is a subset of $S_i$.

**Proof.** The $(s, t)$ block entry of $\delta_{X'}(k)$ denoted by $\delta_{X'}^{st}(k)$ can be written as

$$\delta_{X'}^{st}(k) = h(k+s, k-t+1) - h(k+s-1, k-t)$$

$$= C(k+s)A(k+s-1) \cdots A(k-t+1)B(k-t+1)$$

$$- C(k+s-1)A(k+s-2) \cdots A(k-t)B(k-t)$$

for $1 \leq s \leq 2n + 1$ and $1 \leq t \leq 2n$. We first consider the case $0 < i < i'$. Then, $\delta_{X'}^{st}(k) = 0$ if $k + 2n + 1 < k_{i+1}$ and $k - 2n \geq k_i$ for some $i \in (0, i')$. Thus, $\delta_{X'}^{st}(k) = 0$ for all $k$ in the interval $[k_1 + 2n_1 k_{i+1} - 2n - 2]$. Let $S_i \subset \mathbb{Z}_{X'}$ be the largest closed interval with $[k_1 + 2n_1 k_{i+1} - 2n - 2] \subseteq S_i$.

Now, assume that $\delta_{X'}^{st}(k_{i+1} - 1) = \delta_{X'}^{st}(k_{i+1}) = 0$. Then,

$$\mathcal{M}_{2n+1, 2n}(k_{i+1} - 1) = \mathcal{M}_{2n+1, 2n}(k_{i+1}) = \mathcal{M}_{2n+1, 2n}(k_{i+1} + 1). \quad (69)$$

The first equality in (69) implies that $T(k_{i+1} - 1) = T(k_{i+1})$ and $\hat{A}(k_{i+1} - 1) \sim A(k_{i+1} - 1)$ as a result. Likewise, from the second equality, we get $T(k_{i+1}) = T(k_{i+1} + 1)$ and $\hat{A}(k_{i+1}) \sim A(k_{i+1})$. 15
But, $\hat{A}(k_{i+1} - 1) = \hat{A}(k_{i+1})$. Thus, $A(k_{i+1} - 1) \sim A(k_{i+1})$ or $A\varphi(k_{i+1} - 1) \sim A\varphi(k_{i+1})$ and therefore $\mathcal{M}(A\varphi(k_{i+1} - 1)) = \mathcal{M}(A\varphi(k_{i+1}))$. From Assumption 4.1, we then have $\varphi(k_{i+1} - 1) = \varphi(k_{i+1})$. Since $\varphi(k_{i+1} - 1) = \varphi(k_{i})$ and $\varphi(k_{i}) \neq \varphi(k_{i+1})$, we reach a contradiction. Thus, both $k_{i+1} - 1$ and $k_{i+1}$ can not be in $\mathbb{Z}_{\mathcal{M}}$ for all $0 < i < i'$. Hence, if $\delta_{\mathcal{M}}(k_{i+1}) = 0$, then $k_{i+1} - 1 \notin \mathbb{Z}_{\mathcal{M}}$ which implies that $S_i \subset (-\infty, k_{i+1} - 2]$. If $\delta_{\mathcal{M}}(k_{i+1}) \neq 0$, then $S_i \subset (-\infty, k_{i+1} - 1]$. Combining these two results, we get $S_i \subset (-\infty, k_{i+1} - 2] \cup (-\infty, k_{i+1} - 1] = (-\infty, k_{i+1})$. The same arguments applied to $\delta_{\mathcal{M}}(k_{i} - 1)$ and $\delta_{\mathcal{M}}(k_{i})$ show that $\{k_{i-1}, k_{i}\} \notin \mathbb{Z}_{\mathcal{M}}$. Thus, if $k_{i} - 1 \notin \mathbb{Z}_{\mathcal{M}}$, then $k_{i} \notin \mathbb{Z}_{\mathcal{M}}$ and therefore, $S_i \subset (k_{i} \infty)$. If $k_{i} \in \mathbb{Z}_{\mathcal{M}}$, on the other hand, $k_{i} - 1 \notin \mathbb{Z}_{\mathcal{M}}$. Therefore, $S_i \subset [k_{i} \infty)$. Combining both cases we get $S_i \subset (k_{i} \infty) \cup [k_{i} \infty) = [k_{i} \infty)$. It follows that $S_i$ is a subset of $(-\infty, k_{i+1}) \cap [k_{i} \infty) = [k_{i} k_{i+1})$. The case $i = 0$ follows from (67) on plugging $k' \Rightarrow k_{i}$ and $k_{1} \Rightarrow k_{i+1}$. The last case $i = i'$ follows from (67) on plugging $k_{i'} \Rightarrow k_{i}$ and noting that $k_{i'}$ is not a switch.

Each of the intervals $S_i$, $0 < i < i'$ in the lemma contain at least

$$N_S = \min \{\delta_0(\chi) - 6n - 1, \delta_0(\chi) - 4n - 1, \delta_0(\chi) - 8n\}$$

(70)

points provided that $N_S > 0$. When $N_S > 0$, each segment of $\chi$ contains a closed interval of at least $N_S$ points that is a subset of $\mathbb{Z}_{\mathcal{M}}$. As a result, these intervals are disconnected. This observation suggests an estimation algorithm to extract the discrete states by clustering.

Let $\mathbb{Z}_{\mathcal{M},\epsilon} = \{k \in [k' k''] : ||\delta_{\mathcal{M}}(k)||_F < \epsilon\}$. We have $\mathbb{Z}_{\mathcal{M}} = \bigcap_{\epsilon > 0} \mathbb{Z}_{\mathcal{M},\epsilon}$. Since $N < \infty$ and the number of the switches is finite, $\mathbb{Z}_{\mathcal{M}} = \mathbb{Z}_{\mathcal{M},\epsilon_2}$ for some $\epsilon_2 > 0$. Suppose $N_S > vn$ which is guaranteed if $\delta_0(\chi) \geq (6 + 4n) + 1$, $\delta_0(\chi) \geq (4 + v)n + 1$, and $\delta_0(\chi) \geq (8 + v)n$. Consider two segments $[k_{i} k_{i+1})$ and $[k_{i+1} k_{i+2})$ for some $0 < i < i'$. By Lemma 4.1 and the remark just made, they contain two closed intervals $S_i = [\alpha_i, \beta_i]$ and $S_{i+1} = [\alpha_{i+1}, \beta_{i+1}]$ in $\mathbb{Z}_{\mathcal{M},\epsilon_2}$. Let $\gamma_i$ be the midpoints of $S_i$, $0 < i < i'$, i.e., $\gamma_i = (\alpha_i + \beta_i)/2$. Since we assumed $N_S \geq vn$, $[\alpha_i, \beta_i]$ for all $0 < i < i'$ and from (66)-(68)

$$\alpha_{i+1} - \beta_i \leq k_{i+1} + 2n - (k_{i+1} + 2n - 2) = 4n + 2$$

(71)

Thus, if $[k' k'']$ is split to the sets $\mathbb{Z}_{\mathcal{M},\epsilon_2}$ and $\mathbb{Z}_{\mathcal{M},\epsilon_2} - [k' k'']$, the first set will contain disjoint intervals of length at least $vn$ and separated from each other by a distance at most $4n + 2$. If $v > 5$, $S_5$ is detectable by visual inspection since any transition band around it is not larger than $4n + 2$. Now, use the feature (64) for clustering with the estimators $\hat{A}(\gamma_i)$, $0 \leq i \leq i'$ satisfying $k' \leq k_i \leq k''$. Since $\hat{A}(\gamma_i) \sim A(\gamma_i)$, $\mathcal{M}(\hat{A}(\gamma_i)) = \mathcal{M}(A(\gamma_i))$. Hence, there are at most $\sigma_\epsilon$ clusters. It remains to show that the number of the clusters is $\sigma$, but this follows from the following assumption.

**Assumption 4.2.** Every discrete state in $\mathcal{P}$ is visited at least once in $[k' k'']$ and $N_S > 5n$. \[ \square \]

In Section 9 we will use the density-based clustering algorithm, DBSCAN, proposed in [57] implemented by the dbcan command in MATLAB. Another popular method is the $k$-means clustering algorithm [60]. It is implemented by the kmeans command in MATLAB. Unlike the $k$-means clustering algorithm, the density-based clustering algorithm does not need the number of clusters to be specified a priori. In principle, any clustering algorithm can be used. See, for example, [61] for further information on clustering algorithms. The time-varying $H_\epsilon$ norm of $\mathcal{P}(k)$ or the sum of the Frobenius norms of $\hat{A}(k), \hat{B}(k), \hat{C}(k), \hat{D}(k)$ were proposed in [33] as alternative features for clustering. When $D_k$, $1 \leq k \leq \sigma$ is a unimodal collection, $\hat{D}(k)$ may be used as a feature for clustering though it may be hard to enforce the unimodal assumption in practice since more than one discrete state could be strictly proper.

16
If \( k \in S_i \) for some \( 0 \leq i \leq i^* \), then \( T(k) = T(k+1) \). Hence, \( \hat{\mathcal{P}}(k) \sim \mathcal{P}(k) = \mathcal{P}(\gamma) \) for all \( k \in S_i \) and we set \( \hat{\mathcal{P}}(k) = \mathcal{P}(\gamma) \) there. Therefore, we also determine the switching sequence on \( S_i \) as \( \varphi(k) = \varphi(\gamma) \). This leaves \( \varphi(k) \) undetermined only on \( [k_i, k_{i+1}) - S_i \). From Lemma 4.1 this set has at most \( 4n + 1 \) elements if \( 0 < i < i^* \) and even less if \( i = 0 \) or \( i = i^* \). This is the advantage of working with \( \mathbb{Z}_m^N_{\mathcal{R}_e} \). As an example, consider the extreme case of one switch. Once two discrete states are identified, identifying a sequence \( \varphi(\alpha_2 - 1), \varphi(\alpha_2 - 2), \ldots \) we find \( k_1 \) by no more than \( 2n \) trials, no matter how large \( N! \). Another advantage working with \( \mathbb{Z}_m^N_{\mathcal{R}_e} \) is that \( S_i \), \( 0 \leq i \leq i^* \) are determined without bothering the similarity or the Lyapunov transformations.

The second part of Assumption 4.2 is not really necessary. As long as the first part is satisfied by \( \sigma \) sufficiently long segments of \( \chi \), the conclusion drawn still holds. Picking a large lower bound for \( N_3 \) helps visualize \( S_i \) intervals by overlooking transition bands between them. The number of the clusters may exceed \( \sigma \) when the Markov parameters are corrupted by noise. A re-clustering will be necessary to reduce this number by eliminating clusters over short segments. We summarize the results derived in this section as an estimation algorithm in Appendix. This algorithm estimates the discrete states in two stages. The first stage is divided into Steps 1-5 while the second into Steps 6-7. This division reduces computational burden of the clustering algorithm since there are at most \( O(\delta_1^{-1}N) \) clusters. In [16], it was suggested to directly cluster the Markov parameters in \( \mathcal{R}_{2n+1,2n}(k) \) after putting them into a vector form for each \( k \). This is also a feasible approach; but, increases computational burden of the clustering algorithm.

The results derived in this section for the noiseless Markov parameters case are summarized in the following.

**Theorem 4.1.** Consider Algorithm 2 with the noiseless Markov parameters of the MIMO-SLS model [2–4]. Suppose that \( \chi \) and \( \mathcal{P} \) satisfy Assumptions 3.1–3.2 and 4.1–4.2. Then, Algorithm 2 recovers every discrete state in \( \mathcal{P} \) up to similarity transformations.

### 4.1 Time complexity of Algorithm 2

In Steps 1–3 of Algorithm 2, \( \mathbb{Z}_m^N_{\mathcal{R}_e} \) is found by calculating \( \| \delta_{\mathcal{R}_e}(k) \|_{\mathcal{F}}, \forall k \in [k', k''] \) and a given \( e_{\mathcal{R}} > 0 \). Since \( \| \delta_{\mathcal{R}_e}(k) \|_2^2 = \text{trace}(\delta_{\mathcal{R}_e}^T(k)\delta_{\mathcal{R}_e}(k)) \), time complexity of the matrix multiplication inside the trace operator is \( O(n^2) \). Thus, calculation of \( \mathbb{Z}_m^N_{\mathcal{R}_e} \) has time complexity \( O(Nn^2) \) without considering the nesting property (30). In Steps 4–6, Algorithm 1 is run over a subset \( \kappa \subset [k', k''] \). Recall that \( O(1) \leq |\kappa| \leq O(\tilde{N}n^{-2}) \) where \( O(1) \) is for a few long intervals or switches in \( \chi \) and \( O(\tilde{N}n^{-1}) \) for evenly distributed switches. Hence, from Section 3.1 we infer that Steps 4–6 have a worst-case time complexity \( O(N\text{poly}(n)) \). Time complexity of clustering varies from one algorithm to another. For example, DBSCAN algorithm has complexity \( O(|\kappa| \log |\kappa|) \) or \( O(|\kappa|^2) \) depending on the hyper-parameter values selected and whether the data are degenerate or not. Space complexity, i.e., memory requirement for recomputations in the DBSCAN, changes from \( O(|\kappa|) \) to \( O(|\kappa|^2) \). The k-means algorithm has complexity \( O(\tau|\kappa|\sigma) \) where \( \tau \) is the number of iterations needed to complete clustering process. It is observed that \( \tau \approx O(|\kappa|) \). Hence, the k-means algorithm has time complexity \( O(|\kappa|^2\sigma) \). Attempts have been made to improve the iteration complexity and reduce the time-complexity of the k-means algorithm to \( O(|\kappa|\sigma) \). Time or space complexity depends also on whether a supervised or unsupervised choice is made for clustering. An important factor is robustness to noise.

### 4.2 Robustness of Algorithm 2

Recall that \( \mathbb{Z}_m = \mathbb{Z}_m^N_{\mathcal{R}_e} \) for some \( e_{\mathcal{R}} \) when the Markov parameters are noise-free. Consider the bounded-but-unknown noise model in (31). Let \( \tilde{\delta}_{\mathcal{R}_e}(k) = R_{2n+1,2n}(k+1) - R_{2n+1,2n}(k) \). If
Then, no need to change Steps 1–4 in Algorithm 1. For a fixed $P$ in Assumptions 4.1–4.2 as topological equivalence is transitive, Step 7 discloses all discrete-states under Assumptions 4.1–4.2 as $\varepsilon \to 0$ up to arbitrary similarity transformations. The following extends Theorem 4.1 to the noisy Markov parameters case.

**Theorem 4.2.** Consider Algorithm 2 with the noisy Markov parameters in (31). Suppose that $\chi$ and $P$ satisfy Assumptions 3.1–3.2, Then, Algorithm 2 recovers every discrete state in $P$ up to arbitrary similarity transformations as $\varepsilon \to 0$.

### 5. Switch detection from a modified LTV model

In this section, we propose an estimation algorithm to find the complements of $S_i$ in the segments they are lying for $0 \leq i \leq i^*$, cf. Lemma 4.1. This algorithm is iterative in time and starts from a given point in $S_i$. It is based on correcting the quadruples delivered by Algorithm 1 and proceeds in the positive and/or negative directions until a switch or a switch pair trapping $S_i$ is detected. The lower bound constraints on the dwell times in Lemma 4.1 apply. This process is repeated for all segments fulfilling the dwell time constraints in Lemma 4.1.

We first examine advancement in the positive direction recalling that the true/estimated extended observability matrices are related by

$$
\hat{O}_{2n+1}(k + \ell) = \hat{O}_{2n+1}(k + \ell)T^{-1}(k + \ell), \quad k \in [k', k''] \quad \ell = 0, 1.
$$

Let $\hat{V}_{2n+1}(k) = \hat{O}_{2n+1}(k)\hat{O}_{2n+1}(k + 1)$ and

$$
\hat{V}_{2n+1}(k) = \hat{O}_{2n+1}^\dagger(k)\hat{O}_{2n+1}(k + 1).
$$

Then, from (72) with $\ell = 0$ and $\ell = 1$ we get what we call the forward correction operator

$$
\hat{V}_{2n+1}(k) = T(k)\hat{V}_{2n+1}(k)T^{-1}(k + 1).
$$

Premultiplying $A(k)$ and $B(k)$ in (24) and (28) with $\hat{V}_{2n+1}(k)$, we get from (74)

$$
A_t(k) = \hat{V}_{2n+1}(k)A(k) = T(k)\hat{V}_{2n+1}(k)A(k)T^{-1}(k),
$$

$$
B_t(k) = \hat{V}_{2n+1}(k)B(k) = T(k)\hat{V}_{2n+1}(k)B(k).
$$

We leave $\hat{D}(k)$ and $\hat{C}(k)$ as they are, i.e., $\hat{D}(k) = D(k)$ and $\hat{C}(k) = C(k)T^{-1}(k)$. Thus, $T(k)$ maps the corrected quadruple $(\hat{V}_{2n+1}(k)A(k), \hat{V}_{2n+1}(k)B(k), C(k), D(k))$ to $\hat{\mathcal{P}}(k) = (A_t(k), B_t(k), \hat{C}(k), \hat{D}(k))$. It is a time-varying similarity transformation satisfying $\hat{\mathcal{P}}(k) \sim \mathcal{P}(k)$ if $\hat{V}_{2n+1}(k) = I_n$.

Now, consider the block-rows of $\hat{O}_{2n+1}(k)$ given by $C(k)$ and

$$
C(k + j)\Phi(k + j, k) = C(k + j)A(k + j - 1) \cdots A(k), \quad 1 \leq j \leq 2n.
$$

We need $C(k + 2n + 1)$ and $C(k + j), 0 \leq j \leq 2n$ to calculate $\hat{V}_{2n+1}(k)$. They are equal to $C(k_i)$ or $A(k_i)$ for all $k_i \in [k_i, k_{i+1} - 2n - 2]$ if $0 < i < i^*$ and $\hat{\delta}(\chi) \geq 2n + 2$ and therefore, $\hat{O}_{2n+1}(k) = \hat{O}_{2n+1}(k + 1)$. Consequently, $\hat{V}_{2n+1}(k) = \hat{O}_{2n+1}(k)\hat{O}_{2n+1}(k + 1) = I_n$. Hence, $\hat{\mathcal{P}}(k) \sim \mathcal{P}(k)$ provided that the switches $k_i$ and $k_{i+1}$ are known.
Consider the case $0 < i < i^*$ and set $C_i = C(k_i), A_i = A(k_i)$, and $\tilde{C} = C(k + 2n + 1)$. Assuming $k \in [k_i \ k_{i+1} - 2n - 1]$, we have

$$\mathcal{O}_{2n+1}(k+1) = \begin{bmatrix} C(k+1) \\ \vdots \\ C(k+2n)A(k+2n-1) \cdots A(k+1) \\ C(k+2n+1)A(k+2n) \cdots A(k+1) \end{bmatrix} = \begin{bmatrix} C_i \\ \vdots \\ C_iA_i^2n-1 \\ \tilde{C}_i^2n \end{bmatrix},$$

(77)

$$\mathcal{O}_{2n+1}(k) = \begin{bmatrix} C(k) \\ \vdots \\ C(k+2n-1)A(k+2n-2) \cdots A(k) \\ C(k+2n)A(k+2n-1) \cdots A(k) \end{bmatrix} = \begin{bmatrix} C_i \\ \vdots \\ C_iA_i^2n-1 \\ C_i\tilde{A}_i^2n \end{bmatrix}.$$  

From (77),

$$\mathcal{O}_{2n+1}(k)\mathcal{O}_{2n+1}(k+1) = \mathcal{O}_{2n+1}^T(k)\mathcal{O}_{2n+1}(k+1) + (A_i^2n)^T C_i^T (\tilde{C} - C_i) A_i^2n$$

$$= \mathcal{O}_0(k; 2n+1) (I_n + \mathcal{O}_o^{-1}(k; 2n+1)(A_i^2n)^T C_i^T (\tilde{C} - C_i) A_i^2n).$$

(78)

Hence,

$$\mathcal{O}_{2n+1}(k) = T(k) \mathcal{O}_{2n+1}^T(k)\mathcal{O}_{2n+1}(k+1)T^{-1}(k+1)$$

$$= T(k) (I_n + \mathcal{O}_o^{-1}(k; 2n+1)(A_i^2n)^T C_i^T (\tilde{C} - C_i) A_i^2n) T^{-1}(k+1).$$

(79)

Since $\mathcal{M}$ is invariant to similarity transformations, we get

$$\mathcal{M}(\mathcal{O}_{2n+1}^T(k)) = \mathcal{M}(I_n + \mathcal{O}_o^{-1}(k; 2n+1)(A_i^2n)^T C_i^T (\tilde{C} - C_i) A_i^2n).$$

(80)

Since $k \leq k_{i+1} - 2n - 2 \Rightarrow \tilde{C} = C_i$, from (80), $\mathcal{M}(\mathcal{O}_{2n+1}^T(k)) = n$. Conversely, suppose $\mathcal{M}(\mathcal{O}_{2n+1}^T(k)) \neq n$ implies that $\tilde{C} = C(k+2n+1) = C(k+1)$. It remains to pick an initial value $\tilde{k}$ for $k$. From Lemma 4.1, the distance of $S_i = [\alpha_i \beta_i]$ to $k_{i+1}$ denoted by $d(S_i, k_{i+1})$ satisfies $1 \leq d(S_i, k_{i+1}) \leq 2n + 2$. If we choose $\tilde{k} = \beta_i$, then $\tilde{k} = k_{i+1} - 1$ is a possible value, yet the assumption $k \leq k_{i+1} - 2n - 1$ is violated. Therefore, we choose $\tilde{k} = \beta_i - 2n$. Starting from $\tilde{k}$, we reach to $\tilde{k}$ in $k - \tilde{k} = k_{i+1} - 1 - \beta_i$ steps. Tight upper and lower bounds on $k - \tilde{k}$ are derived as follows

$$0 \leq k_{i+1} - 1 - (k_{i+1} - 1) \leq k - \tilde{k} = k_{i+1} - 1 - \beta_i \leq k_{i+1} - 1 - (k_{i+1} - 2n - 2) = 2n + 1.$$  

This case extends to $i = 0$ by letting $k_0 = 1$. For $i = i^*$, we simply let $\varphi(k) = \varphi(\beta_i)$ for all $k \in (\beta_i, k^*)$ since $k^*$ is not a switch. Now, we examine the backward corrections case.

Recall that the true extended controllability matrices and their estimates are related by

$$\mathcal{R}_{2n}(k - \ell) = T(k - \ell + 1)\mathcal{R}_{2n}(k - \ell), \quad k \in [k' \ k''], \quad \ell = 0, 1.$$  

(81)

Let $\mathcal{R}_{2n}(k) = \mathcal{R}_{2n}(k - 1)\mathcal{R}_{2n}^T(k)$ and

$$\mathcal{R}_{2n}(k) = \mathcal{R}_{2n}(k - 1)\mathcal{R}_{2n}^T(k).$$

(82)
Then, from \((81)\) with \(\ell = 0\) and \(\ell = 1\) we get what we call the backward correction operator
\[
\mathcal{W}_2(n) = T(k)\mathcal{W}_2(n)T^{-1}(k+1)
\] (83)

Post-multiplying \(\hat{A}(k)\) in \((24)\) and \(\hat{C}(k)\) in \((28)\) with \(\mathcal{W}_2(n)\), we get from \((83)\)
\[
A_b(k) = \hat{A}(k)\mathcal{W}_2(n) = T(k+1)A(k)\mathcal{W}_2(n)T^{-1}(k+1),
\]
\[
C_b(k) = \hat{C}(k)\mathcal{W}_2(n) = C(k)\mathcal{W}_2(n)T^{-1}(k+1).
\] (84)

Since \(\hat{B}(k) = T(k+1)B(k)\) and \(\hat{D}(k) = D(k)\), we do not need to change them. Thus, \(T(k+1)\) maps the modified quadruple \((A(k)\mathcal{W}_2(n), B(k), C(k)\mathcal{W}_2(n), D(k))\) to \(\mathcal{P}_k(k) = (A_b(k), \hat{B}(k), C_b(k), \hat{D}(k))\).

It is a time-varying similarity transformation and if \(\mathcal{W}_2(n) = I_n\), we then have \(\mathcal{P}_k(k) \sim \mathcal{P}(k)\).

The block-columns of \(\mathcal{W}_2(n)\) start with \(B(k-1)\) and for \(2 \leq j \leq 2n\), they are given by
\[
\Phi(k, j-1)B(k-j) = A(k-1) \cdots A(k+j-1)B(k-j).
\] (85)

We need \(B(k-2n), B(k-j), A(k-j), 0 \leq j < 2n\) to calculate \(\mathcal{W}_2(n)\). They are equal to \(B(k_j)\) or \(A(k_j)\) for all \(k \in [k_1 + 2n \ k_{j+1} - 1]\) if \(\Delta_k(\mathcal{A}) \geq 2n + 1\) and \(0 < i < \tau^*\). Then, \(\mathcal{W}_2(n) = \mathcal{W}_2(n-1)\) and \(\mathcal{W}_2(n) = \mathcal{W}_2(n-1)\mathcal{W}_2(n) = I_n\). Hence, \(\mathcal{P}_k(k) \sim \mathcal{P}(k)\) if \(k_j\) and \(k_{j+1}\) are known.

Consider \(0 < i < \tau^*\) case first as before and set \(B_j = B(k_j), A_j = A(k_j), \hat{B} = B(k-2n)\). Assume \(k \in [k_1 + 2n - 1 \ k_{j+1}]\). Then,
\[
\mathcal{W}_2(n) = [B(k) A(k)B(k-1) \cdots A(k-2n+2)B(k-2n+1)] = [B_j A_j B_j \cdots A_j^{2n-1} B_j],
\]
\[
\mathcal{W}_2(n) = [B(k-1) A(k-1)B(k-2) \cdots A(k-2n+1)B(k-2n)] = [B_j A_j B_j \cdots A_j^{2n-1} B_j].
\] (86)

From \((86)\),
\[
\mathcal{W}_2(n) = [B(k) A(k)B(k-1) \cdots A(k-2n+2)B(k-2n+1)] = [B_j A_j B_j \cdots A_j^{2n-1} B_j].
\]

Hence,
\[
\mathcal{W}_2(n) = T(k)\mathcal{W}_2(n)T^{-1}(k+1)
\]
\[
= T(k) [I_n + A_i^{2n-1} (\hat{B} - B_i)B_i^T (A_i^{2n-1})^T] T^{-1}(k+1).
\] (88)

Since \(\mathcal{M}\) is invariant to similarity transformations,
\[
\mathcal{M}(\mathcal{W}_2(n)) = \mathcal{M}(I_n + A_i^{2n-1} (\hat{B} - B_i)B_i^T (A_i^{2n-1})^T) T^{-1}(k+1;2n).
\] (89)

Since \(k \geq k_2 + 2n \Rightarrow B = B_j\), from \((89)\) we have \(\mathcal{M}(\mathcal{W}_2(n)) = n\). Conversely suppose \(\mathcal{M}(\mathcal{W}_2(n^2)) \neq n\) implies that \(k^1 = k_1 + 2n - 1\). It is the smallest \(k\) satisfying \(\mathcal{M}(\mathcal{W}_2(n)) \neq n\). Then, \(B = B(k_1 - 1) = B(k_1 + 1)\). Note that \(k_1 + 1\) is a switch in the reverse direction. Next, we choose an initial value \(k\) for \(k\). From Lemma 4, the distance of \(S_i = [\alpha_k \ B_i]\) to \(k_j\) satisfies \(0 \leq d(S_i, k_j) \leq 2n\). If we choose \(k = \alpha_1\), then \(\alpha_1 = k_1\) is a possible value, yet the assumption \(k \geq k_1 + 2n - 1\) is violated.
Therefore, we choose $\tilde{k} = \alpha_i + 2n - 1$. Starting from $\tilde{k}$, we reach to $k_i^2$ in $\tilde{k} - k_i^2 = \alpha_i - k_i$ steps. Upper and lower bounds on $\tilde{k} - k_i^2$ are derived as follows

$$0 \leq \tilde{k} - k_i^2 = \alpha_i - k_i \leq 2n.$$ 

This case extends to $i = i^*$ without modification. For $i = 0$, we simply let $\varphi(k) = \varphi(\alpha_0)$ for all $k \in [k' \alpha_0]$ since $k'$ is not a switch. The switch detectability conditions are stated as follows.

**Assumption 5.1.** The MIMO–SLS model (2.26)(2.28) satisfies the following conditions

(a) $\mathcal{M} \left[ I_n + G_0^{-1} (k_{i+1} - 2n - 1; 2n + 1) [A^T (k_0)]^{2n} + [C(k_{i+1} - C(k_i)) [A(k_i)]^{2n}} \right] \neq n$ for all $0 \leq i < i^*$ with the convention $A(k_0) = A(k')$ and $C(k_0) = C(k')$

(b) $\mathcal{M} \left[ I_n + [A(k_i)]^{2n-1} (B(k_{i-1}) - B(k_i)) B^T (k_i) [A^T (k_i)]^{2n-1} \emptyset_{\mathcal{E}}^{2n-1} (k_i + 2n; 2n) \right] \neq n, 0 < i < i^*$.

If Assumption 5.1 holds, the two conditions mentioned in the forward and backward iterations are satisfied. A different feature than $\mathcal{M}$ to detect switches is $\|D(k + 1) - D(k)\|$ where $\|\cdot\|$ is a matrix norm if $D(\varphi(k)) = D(\varphi(l))$ iff $\varphi(k) = \varphi(l)$. Let us examine Conditions (a)–(b) in Assumption 5.1 through simple examples.

**Example 5.1.** Let $X$ be defined by

$$X = \begin{bmatrix} 0 & \lambda \\ 0 & 0 \end{bmatrix}, \quad \lambda \in \mathbb{R}.$$ 

Then, $\mathcal{M}(I_2 + X) = 2$ for all $\lambda \in \mathbb{R}$, yet $I_2 + X \neq I_2$ except $\lambda = 0$.

If the second term in the argument of $\mathcal{M} (\cdot)$ is a strictly upper or lower triangular matrix, then $k_{i+1}$ or $k_i$ cannot be detected by the proposed algorithm. We present two more examples.

**Example 5.2.** $A(k_i)$ is idempotent, i.e., $[A(k_i)]^{2n} = 0$ and/or $[A(k_i)]^{2n-1} = 0$, yet $A(k_i) \neq 0$. Either or both of the conditions in Assumption 5.1 fail since $\mathcal{M}(\hat{V}_{2n+1}(k_{i+1} - 2n - 1)) = n$ and/or $\mathcal{M}(I_{2n}(k_i + 2n - 1)) = n$.

**Example 5.3.** If $C(k_i) \perp C(k_{i+1} - C(k_i))$, $\mathcal{M}(\hat{V}_{2n+1}(k_{i+1} - 2n - 1)) = n$ and $k_{i+1}$ cannot be detected by the algorithm to be proposed. If $B(k_i) \perp B(k_{i-1}) - B(k_i)$, $\mathcal{M}(\hat{V}_{2n+1}(k_{i+1} - 2n - 1)) = n$ and $k_{i-1}$ cannot be detected.

Despite these and many other examples which can be constructed, Assumption 5.1 holds for almost all switch pairs and model sets. The above derivations are summarized as Algorithm 3 in Appendix. This algorithm detects a switch or switch pair subject to the dwell time constraints in Lemma 4.1. In particular, if $N_S > 5n$, then all switches in $\chi$ are detected. When a dwell time constraint fails, alternative procedures presented in the following sections may be resorted. The main result derived in this section for the noiseless data case is summarized in the following.

**Theorem 5.1.** Consider Algorithm 3 driven by the noise-free Markov parameters of (2.26)–(2.28) and the outputs of Algorithm 2. Suppose that $\chi$ and $\mathcal{P}$ satisfy Assumptions 4.1 and 5.1. Then, Algorithm 3 recovers $\varphi$ on the segments with dwell times satisfying $\delta_i(\chi) \geq 4n + 2$ for $0 < i < i^*$, $\delta_0(\chi) \geq 6n + 2$, and $\delta_{i^*}(\chi) > 8n$.

If $N_S > 5n$, each segment of $\chi$ satisfies a dwell time condition in Theorem 5.1 and Algorithm 3 detects all switches in $\chi$. Recall that Algorithm 2 recovers the discrete states in the
uniformly in $k$ the perturbed matrices $\tilde{A}(k)$ as in Section 3.1 and we find them $O(\text{poly}(n))$ for a third order polynomial in $n$. Finding eigenvalues of an $n$ by $n$ matrix has complexity $O(n^3)$. Since the number of steps to detect a switch is bounded above by $2n + 1$, detection of one switch has a worst-case complexity $O(\text{poly}(n))$ for a fourth order polynomial in $n$. The worst-case time complexity of Algorithm 3 is $O(N\text{poly}(n))$ for a third order polynomial in $n$ since there are at most $O(Nn^{-1})$ switches when Assumption 3.2 holds.

5.2. Robustness of Algorithm 3

From Theorem 3.1 we have $\mathcal{M}(\gamma) = \tilde{A}(k)\mathcal{D}(\gamma)$ uniformly in $k$ as $\epsilon \to 0$. This means $\tilde{A}_{i+1}(k) \to \tilde{A}(k)$ and $\mathcal{M}_i(k) \to \tilde{A}(k)$ uniformly in $k$ as $\epsilon \to 0$. Since eigenvalues of a matrix are continuous functions of matrix entries, $\mathcal{M}(\tilde{A}_{i+1}(k)) \to \mathcal{M}(\tilde{A}(k))$ and $\mathcal{M}(\tilde{A}_{i}(k)) \to \mathcal{M}(\tilde{A}(k))$ uniformly in $k$ as $\epsilon \to 0$. Recall that $\mathcal{M}(\cdot)$ is invariant to similarity transformations. Plugging the perturbed matrices $\tilde{A}_{i+1}(k)$ and $\tilde{A}_i(k)$ on the left hand-sides, we see that the threshold criteria in Algorithm 3 never fail as $\epsilon \to 0$ under Assumptions 3.1, 4.1, 4.2, and 5.1. The following extends Theorem 5.1 to the noisy Markov parameters case.

**Theorem 5.2.** Consider Algorithm 3 driven by the noisy Markov parameters in (31) and the outputs of Algorithm 2. Suppose that $\chi$ and $\mathcal{M}$ satisfy Assumptions 3.1, 3.2, 4.1, and 5.1. Then, Algorithm 3 recovers $\phi$ on the segments with dwell times satisfying $\delta_i(\chi) \geq 4n + 2$ for $0 < i < 1$, $\delta_i(\chi) \geq 6n + 2$, and $\delta_i(\chi) > 8n$ as $\epsilon \to 0$.

6. Switch detection from Markov parameters

Let us reconsider the intervals $S_i = [\alpha_i, \beta_i]$ with the midpoints $\gamma_i = (\alpha_i + \beta_i)/2, 0 < i < 1$ in Lemma 4.1. We showed that $\mathcal{M}(\gamma_i) = (\tilde{A}(\gamma_i)\mathcal{B}(\gamma_i), \tilde{C}(\gamma_i), \mathcal{D}(\gamma_i))$ is similar to the true model $\mathcal{M}(\gamma_i) = (A(\gamma_i), B(\gamma_i), C(\gamma_i), D(\gamma_i))$. This means that $A(\gamma_i) = T_{i-1}^{-1}\tilde{A}(\gamma_i)T_i, B(\gamma_i) = T_{i-1}^{-1}\mathcal{B}(\gamma_i), C(\gamma_i) = \tilde{C}(\gamma_i)T_i, \mathcal{D}(\gamma_i) = D(\gamma_i)$ for a non-singular matrix $T_i$. Moreover, $0 \leq \alpha_i - k_i \leq 2n$.

Let $\ell > \beta_i$ and suppose we have shown that $\phi(\gamma_k) = \phi(\gamma_i)$ for all $\ell - 2n \leq k \leq \ell - 1$. Plugging $\ell = \beta_i + 1$, from $\ell - 2n \geq \alpha_i$ we see that $\beta_i - \alpha_i \geq 2n - 1$ holds. Then, from (6) for $0 < j < 2n$,

$$h(\ell, \ell - j) = \begin{cases} C(\ell)A(\gamma_i)j^{-1}B(\gamma_i), & j \neq 0 \\ D(\ell), & j = 0. \end{cases}$$

(90)

We want to cluster $\phi(\ell)$ to either $\phi(\gamma_i)$ or $\phi(\gamma_{i+1})$. Plug the similarity equations for $\mathcal{M}(\gamma_i)$ in (90) to get

$$h(\ell, \ell - j) = \begin{cases} \tilde{C}(\gamma_i)j^{-1}\tilde{B}(\gamma_i), & j \neq 0 \\ \tilde{D}(\ell), & j = 0. \end{cases}$$

(91)
where \( \hat{C} = C(\ell)T_i^{-1} \) and \( \hat{D} = D(\ell) \). Treating \( \hat{C} \) and \( \hat{D} \) as unknowns, the right-hand side of (91) becomes a linear estimator of \( h(\ell, \ell - j) \) denoted by \( \hat{h}(\ell, \ell - j) \). We find \( \hat{C} \) and \( \hat{D} \) by solving the linear least-squares minimization problem

\[
\min_{\hat{C} \in \mathbb{R}^{p \times n}, \hat{D} \in \mathbb{R}^{p \times m}} f(\hat{C}, \hat{D})
\]

where

\[
f(\hat{C}, \hat{D}) = \sum_{j=0}^{2n} \| \hat{h}(\ell, \ell - j) - h(\ell, \ell - j) \|^2_F.
\]

The minimization problem (92) is separable in \( \hat{C} \) and \( \hat{D} \). Thus, \( \hat{D} = D(\ell) \). Suppose a switch detectability condition \( D(\ell) \neq D(k) \iff \varphi(\ell) \neq \varphi(k) \) is in force. Then, \( \hat{D} = D(\ell) \) and \( \hat{C} \) and \( \varphi(\ell) \) are determined unambiguously. When this switch detectability condition does not hold, we determine \( \hat{C} \) by solving the normal equations \( H_{1,2n}(\ell) = \hat{C}\hat{R}_{2n}(\ell - 1) \) where \( \hat{R}_{2n}(\ell - 1) \) is formed from the controllability pair \( (\hat{A}(\gamma), \hat{B}(\gamma)) \). Hence, the unique minimizer is found as

\[
\hat{C} = H_{1,2n}(\ell)\hat{R}_{2n}^T(\ell - 1) = C(\ell)\hat{R}_{2n}(\ell - 1)\hat{R}_{2n}^T(\ell - 1) = C(\ell)T_i^{-1}.
\]

Not surprisingly \( \hat{h}(\ell, \ell - j) = h(\ell, \ell - j) \) for all \( 0 \leq j \leq 2n \). On substituting \( \hat{C} \) from (94) into (91) and comparing it with (90), we see this. To carry on, we impose a switch detectability condition.

**Assumption 6.1.** The MIMO–SLS model (2)–(4) satisfies

\[
[C(k) \ D(k)] \neq [C(l) \ D(l)] \iff \varphi(k) \neq \varphi(l), \quad (95)
\]

\[
\begin{bmatrix} B(k) \\ D(k) \end{bmatrix} \neq \begin{bmatrix} B(l) \\ D(l) \end{bmatrix} \iff \varphi(k) \neq \varphi(l). \quad (96)
\]

If \( \varphi(\ell) = \varphi(\gamma), \ C(\ell) = C(\gamma), \ D(\ell) = D(\gamma) \) and from (94) \( \hat{C} = \hat{C}(\gamma), \hat{D} = \hat{D}(\gamma) \). Conversely, suppose \( \hat{C} = C(\gamma) \) and \( \hat{D} = D(\gamma) \), yet \( \varphi(\ell) \neq \varphi(\gamma) \), that is, \( \varphi(\ell) = \varphi(\gamma + 1) \). Then, \( C(\ell) = C(\gamma + 1) \) and \( D(\ell) = D(\gamma + 1) \). Hence, from (94) \( \hat{C} = C(\gamma + 1)T_i^{-1} = \hat{C}(\gamma + 1)T_{i+1}T_i^{-1} \) where \( T_{i+1} \) is the similarity transformation in \( \hat{P}(\gamma + 1) \sim \hat{P}(\gamma + 1) \). Moreover, \( \hat{D} = D(\ell) = D(\gamma + 1) \). Since by assumption \( \hat{D} = D(\gamma), \) from \( \hat{D}(\ell) = D(\gamma) \) we then get \( D(\gamma) = D(\gamma + 1) \). Furthermore, we derive \( C(\gamma) = C(\gamma + 1) \) from the chain of equalities \( C(\gamma) = \hat{C}(\gamma)T_i = \hat{C}(\gamma + 1)T_{i+1} = C(\gamma + 1) \). Therefore, from (95) in Assumption 6.1 \( \varphi(\gamma) = \varphi(\gamma + 1) \). We reach a contradiction. With \( \hat{D} = \hat{D}(\ell) \) and \( \hat{C} = H_{1,2n}(\ell)\hat{R}_{2n}^T(\ell - 1) \), we conclude \( \hat{C} \hat{D} = [\hat{C}(\gamma) \hat{D}(\gamma)] \iff \varphi(\ell) = \varphi(\gamma) \) when Assumption 6.1 holds. Until \( k_{i+1} \) is detected, \( \ell \) is increased.

We now proceed along the negative axis. Let \( \ell < \alpha \) and suppose for all \( \ell + 1 \leq k \leq \ell + 2n \), we showed that \( \varphi(k) = \varphi(\gamma) \), which requires \( \beta_1 - \alpha \geq 2n - 1 \). This inequality follows from \( \ell + n \leq \beta_1 \) on putting \( \ell = \alpha - 1 \). We want to cluster \( \varphi(\ell) \) to either \( \varphi(\gamma) \) or \( \varphi(\gamma - 1) \). From (96) for \( 0 \leq j \leq 2n \),

\[
h(\ell + j, \ell) = \begin{cases} C(\gamma)A(\gamma)^{j-1}B(\ell), & j \neq 0 \\ D(\ell), & j = 0. \end{cases} \quad (97)
\]

Plugging the similarity equations for \( \hat{P}(\gamma) \) in, we get

\[
h(\ell + j, \ell) = \begin{cases} \hat{C}(\gamma)[\hat{A}(\gamma)]^{j-1}\hat{B}, & j \neq 0 \\ \hat{D}, & j = 0 \end{cases} \quad (98)
\]
where \( \hat{B} = T_i B(\ell) \) and \( \hat{D} = D(\ell) \). Treating \( \hat{B} \) and \( \hat{D} \) as unknowns, the right-hand side of (98) becomes a linear estimator of \( h(\ell + j, \ell) \) denoted by \( \hat{h}(\ell + j, \ell) \). We find \( \hat{B} \) and \( \hat{D} \) by solving the linear least-squares minimization problem

\[
\min_{\hat{B} \in \mathbb{R}^{m \times n}, \hat{D} \in \mathbb{R}^{p \times n}} g(\hat{B}, \hat{D})
\]

where

\[
g(\hat{B}, \hat{D}) = \sum_{j=0}^{2n} \| \hat{h}(\ell + j, \ell) - h(\ell + j, \ell) \|^2 P.
\]

The normal equations for \( \hat{B} \) are \( \mathcal{H}_{2n,1}(\ell + 1) = \hat{\mathcal{O}}_{2n}(\ell + 1) \hat{B} \) with \( \hat{\mathcal{O}}_{2n}(\ell + 1) \) formed from the pair \((\hat{C}(\gamma), \hat{A}(\gamma))\). Furthermore, \( \hat{D} = D(\ell) \). Using the similarity relations, we derive

\[
\hat{B} = \hat{\mathcal{O}}_{2n}^{\dagger}(\ell + 1) \mathcal{H}_{2n,1}(\ell + 1) = \hat{\mathcal{O}}_{2n}^{\dagger}(\ell + 1) \mathcal{O}_{2n}(\ell + 1) B(\ell) = T_i B(\ell).
\]

Note for all \( 0 \leq j \leq 2n \) that \( \hat{h}(\ell + j, \ell) = h(\ell + j, \ell) \).

If \( \varphi(\ell) = \varphi(\gamma) \), \( B(\ell) = B(\gamma) \), \( D(\ell) = D(\gamma) \). Then, from (101) we derive \( \hat{B} = T_i B(\ell) = T_i B(\gamma) = \hat{B}(\gamma) \), and \( \hat{D} = D(\ell) = D(\gamma) = \hat{D}(\gamma) \). Conversely suppose \( \hat{B} = B(\gamma) \) and \( \hat{D} = D(\gamma) \), yet \( \varphi(\ell) \neq \varphi(\gamma) \), that is, \( \varphi(\ell) = \varphi(\gamma) \). Then, \( B(\ell) = B(\gamma) \) and \( D(\ell) = D(\gamma) \). Hence, from (101) we derive \( \hat{B} = T_i B(\gamma) = T_i T_{i-1}^{-1} B(\gamma) \) where \( T_{i-1} \) is the similarity transformation in \( \hat{\mathcal{P}}(\gamma_{i-1}) \sim \mathcal{P}(\gamma_{i-1}) \). Thus, \( B(\gamma) = T_{i-1}^{-1} B(\gamma) = T_{i-1}^{-1} B(\gamma) \). Furthermore, \( \hat{D} = D(\gamma) = D(\gamma) \) and \( \hat{D} = D(\gamma) = D(\gamma) \). Hence, \( \varphi(\gamma) = \phi(\gamma) \) from (96) in Assumption 5.1. We arrive a contradiction. We conclude that with \( \hat{D} = D(\ell) \) and \( \hat{B} = \hat{\mathcal{O}}_{2n}^{\dagger}(\ell + 1) \mathcal{H}_{2n,1}(\ell + 1) = \hat{\mathcal{O}}_{2n}^{\dagger}(\ell + 1) \mathcal{O}_{2n}(\ell + 1) \) holds. Until \( k_i - 1 \) is reached, \( \ell \) is decreased. Since \( \varphi(k_i - 1) \neq \varphi(k_i) = \varphi(k_i + 1) \), it is necessary to pass \( k_i \) left to detect a switch at \( k_i \).

Starting from \( k_i + 1 \) the scheme presented above detects \( k_{i+1} - k_i - 1 \) steps, i.e., between 0 and \( 2n + 1 \) steps, and starting from \( k_i - 1 \) it reaches to \( k_{i+1} - 1 \) in \( k_{i+1} - k_i \) steps, i.e., between 0 and \( 2n \) steps. The numbers of the required steps are the same both in this scheme and Algorithm 3. The differences between the schemes appear to be the requirements on \( \beta_i - \alpha_i \) and the switch detectability conditions. While \( \delta_i(\chi) \geq 4n + 2 \) is stipulated on Algorithm 3, this scheme, called Algorithm 3’, replaces it with \( \beta_i - \alpha_i \geq 2n - 1 \). If \( \delta_i(\chi) \geq (6 + v)n + 1, \delta_i(\chi) \geq (4 + v)n + 1, \) and \( \delta_i(\chi) \geq (8 + v)n \), each segment of \( \chi \) contains an interval \( S_i \) for some \( 0 \leq i \leq i^* \) with at least \( v \nu \) points. Set \( v \geq 2 \). Then, the requirement \( \beta_i - \alpha_i \geq 2n - 1 \) is satisfied. Note that Assumption 4.2 already requires \( v > 5 \). The steps of Algorithm 3’ are outlined in Appendix. The results derived above are captured in the following.

**Theorem 6.1.** Consider Algorithms 3’ with the noisless Markov parameters. Suppose that \( \chi \) and \( \mathcal{P} \) satisfy Assumptions 3.1, 2.2, 4.1, 4.3, and 4.4. Then, Algorithm 3’ recovers \( \varphi \) on segments with dwell times \( \delta_0(\chi) \geq 8n + 1, \delta_i(\chi) \geq 6n + 1 \) for \( 0 < i < i^* \), and \( \delta_{i^*}(\chi) \geq 10n \).

**6.1. Time complexity of Algorithm 3’**

The most time-consuming operations in Algorithm 3’ are the calculations of \( \hat{B} \) and \( \hat{C} \). Similarly to Section 5.1 we find their time complexities \( O(n^3) \). The number of the steps to hit a switch is at most \( O(n) \). Hence, the detection of one switch or all will have worst-case time complexities \( O(n^4) \) or \( O(Nn^3) \), respectively. The latter may be observed when the switches are uniformly distributed in \( \chi \).
6.2. Robustness of Algorithm 3′

First, note that the Lyapunov transformation \( S(k) \) in Theorem 3.2 is absorbed in the similarity transformations \( T_i \) in the formulas for \( \hat{C} \) and \( \hat{B} \). Therefore, \( \hat{P}^\varepsilon(k) \) may replace the perturbed realization \( \hat{P}(k) \) in the calculations. From Theorem 3.2, \( \hat{P}^\varepsilon(k) \to \hat{P}(k) \) as \( \varepsilon \to 0 \). As a result the threshold criteria in Algorithm 3′ never fail as \( \varepsilon \to 0 \) under Assumptions 3.1, 4.1, and 6.1. The following result extends Theorem 6.1 to the noisy Markov parameters case.

**Theorem 6.2.** Consider Algorithms 3′ with the noisy Markov parameters in (31). Suppose that \( \chi \) and \( \hat{\mathcal{P}} \) satisfy Assumptions 3.1, 4.1, and 6.1. Then, Algorithm 3′ recovers \( \varphi \) on segments with dwell times \( \delta_i(\chi) \geq 8n + 1, \delta_i(\chi) \geq 6n + 1 \) for \( 0 < i < i′ \), and \( \delta_i(\chi) \geq 10n \) as \( \varepsilon \to 0 \).

6.3. Estimation of \( \varphi(k) \) on very short segments

Algorithms 2 and 3 or 3′ combined estimates \( \varphi \) on segments with length at least \( 4n + 2 \), but leaves it undetermined on shorter segments. Let \( [k_i, k_{i+1}) \) be such a segment and suppose that \( \hat{\mathcal{P}}(\gamma_{i-1}) \) and \( \varphi(k) \) were estimated on \( [k_i, k_{i+1}) \), but \( \hat{\mathcal{P}}(\gamma) \) and \( k_{i+1} \) need to be estimated. Here, we consider a situation in which the discrete state set is completely identified up to a similarity transformation, but we don’t know which discrete state is active in \( [k_i, k_{i+1}) \) because none of the dwell time constraints in either Theorem 5.1 or Theorem 6.1 apply to \( \delta_i(\chi) \). Set \( k = k_i + n, q = n + 1, \) and \( r = n \) in (9) if \( k_i + 2n \leq k_{i+1} - 1 \), i.e., \( \delta_i(\chi) > 2n \). The resulting Hankel matrix can be expressed in terms of the state-space matrices of \( \hat{\mathcal{P}}(\gamma) \) and \( \hat{\mathcal{P}}(\gamma) \) as follows

\[
\mathcal{H}_{n+1,n}(k_i + n) = \begin{bmatrix}
C(\gamma)B(\gamma) & C(\gamma)[A(\gamma)]^{-1}B(\gamma) \\
C(\gamma)[A(\gamma)]B(\gamma) & \cdots & C(\gamma)[A(\gamma)]^{n-1}B(\gamma)
\end{bmatrix}
\]

\[=
\hat{\mathcal{P}}(\hat{\mathcal{P}}(\gamma)).
\]

(102)

Note the factorization \( \mathcal{H}_{n+1,n}(k_i + n) = \mathcal{O}_{n+1}\mathcal{O}_n \) which determines \( \hat{\mathcal{P}}(\gamma) \) uniquely up to a similarity transformation. This observation suggests that \( \hat{\mathcal{P}}(\gamma) \) can be identified easily as

\[
\hat{\mathcal{P}}(\gamma) = \arg\min_{\mathcal{P}} \|\mathcal{H}_{n+1,n}(k_i + n) - \hat{\mathcal{P}}(\hat{\mathcal{P}}(\gamma))\|_F.
\]

(104)

Then, we declare \( \varphi(k) = \varphi(\gamma) \) on the interval \([k_i, k_i + 2n)\). Since the length of this interval is \( 2n + 1 \), by Algorithm 3′ we can extend \( \hat{\mathcal{P}}(\gamma) \) to \((k_i + 2n, k_{i+1})\) until we hit \( k_{i+1} \). The entire process starts again at \( k_{i+1} \) and continues until left endpoint of a large segment is encountered. For the segment \([k_i', k_{i'}] \), we only need to estimate \( \hat{\mathcal{P}}(\gamma') \) since \( k_{i'} \) is not a switch. Substituting \( k'' = k_{i+1} - 1 \) in the above calculations, we derive the recovery condition \( k'' - k_r \geq 2n \). From \( k'' = N - 4n \) and \( \delta_r(\chi) = N - k_r \), we get \( \delta_r(\chi) \geq 6n \). For the segment \([k', k_{i+1}] \), if \( k_i \) is not known this procedure is applied to estimate both \( \hat{\mathcal{P}}(\gamma) \) and \( k_{i+1} \) by substituting \( k_0 = k' \). The recovery condition is then, \( k_1 - k' > 2n \). From \( k' = 2n + 1 \) and \( \delta_r(\chi) = k_1 - 1 \), we get \( \delta_r(\chi) > 4n \).

In the second case, we assume that \( \hat{\mathcal{P}}(\gamma_{i+1}) \) and \( \varphi(k) \) have been estimated on \([k_{i+1}, k_{i+2}] \) and we want to determine \( \hat{\mathcal{P}}(\gamma), k_r \), and \( \varphi(k) \) on \([k_i, k_{i+1}) \). This is the previous case if one notes...
that $\hat{\mathcal{H}}(\hat{\mathcal{P}}(\gamma))$ does not change when $\mathcal{H}_{n+1,n}(k_i+n)$ is replaced with $\mathcal{H}_{n+1,n}(k_i+n-1)$. We then determine $\hat{\mathcal{P}}(\gamma)$ from

$$\hat{\mathcal{P}}(\gamma) = \arg \min_{P \in \mathcal{P}} \| H_{n+1,n}(k_i+n-1) - \hat{\mathcal{H}}(\hat{\mathcal{P}}(\gamma)) \|_F. \quad (105)$$

The condition on $\delta(\chi)$ is found $\delta(\chi) > 2n$ as before from the inequalities $k = k_i+n-1$ and $k-n \geq k_i$. For the segment $[k_1' \ k_1]$, if $k_1$ is not known this procedure is applied to estimate both $\mathcal{P}(\gamma_1)$ and $\mathcal{P}(\gamma_2)$ by substituting $k_0 = k_1'$. The recovery condition is then, $k_1-k_0 > 2n$. From $k' = 2n+1$ and $\tilde{\delta}(\chi) = k_1-1$, we get $\tilde{\delta}(\chi) > 4n$. For the segment $[k_2 \ k_2']$, if $k_2$ is not known this procedure is applied to estimate both $\mathcal{P}(\gamma_k)$ and $\mathcal{P}(\gamma_{k+1})$ by substituting $k_{r-1} = k' + 1$. We summarize the above calculations in Algorithm 3” detailed in Appendix. The result derived above is captured in

**Theorem 6.3.** Consider Algorithms 3” with the noise-free Markov parameters. Suppose that $\chi$ and $\mathcal{P}$ satisfy Assumptions 3.1, 3.2, and 3.4. Then, Algorithm 3” recovers $\phi$ on segments with dwell times satisfying $\tilde{\delta}(\chi) > 4n$, $\delta(\chi) \geq 2n + 1$, for $0 < i < t^*$, and $\delta(\chi) \geq 6n$.

**6.4. Time complexity of Algorithm 3”**

Computationally two expensive processes are the calculations of $\hat{\mathcal{P}}(\gamma_j)$ in (104) and (105) for $j = 1, \ldots, \sigma$, but they are done only once. Moreover, it suffices to consider only the calculation of $\hat{\mathcal{C}}(\gamma_j)$$[\hat{\mathcal{A}}(\gamma_j)]^{2n-1} \hat{\mathcal{B}}(\gamma_j)$. Recall that eigen decomposition has time complexity $O(n^3)$. Then, $[\hat{\mathcal{A}}(\gamma_j)]^\ell$ for $\ell = 1, \ldots, 2n-1$ are efficiently calculated from this decomposition. Products of three matrices forming Markov parameters have time complexity $O(n^3)$. Thus, $\hat{\mathcal{P}}(\gamma)$ has time complexity $O(n^3)$ for each $j$. There are $\sigma$ discrete states, hence the overall time complexity of (104) or (105) becomes $O(\sigma n^3)$. Now, (104) or (105) are followed by Algorithm 3’ which has time complexity $O(n^3)$. It follows that Algorithm 3” has complexity $O(d_1 \sigma n^3 + d_2 n^4)$ for some $d_1, d_2 > 0$ to detect one switch. If the switches are evenly distributed in $\chi$, this complexity grows to $O(d_1 \sigma N n^2 + d_2 N n^3)$.

**6.5. Robustness of Algorithm 3”**

Since Markov parameters of topologically equivalent systems are identical, from Theorem 5.2 the perturbed Hankel matrix in (103) satisfies $\hat{\mathcal{H}}(\hat{\mathcal{P}}(\gamma)) \rightarrow \mathcal{H}(\hat{\mathcal{P}}(\gamma))$ as $\varepsilon \rightarrow 0$. Then, Algorithm 3” uniquely identify the active discrete state. The rest follows from the robustness of Algorithm 3’ to amplitude bounded noise. We summarize this result in the following.

**Theorem 6.4.** Consider Algorithms 3” with the noisy Markov parameters in (31). Suppose $\chi$ and $\mathcal{P}$ satisfy Assumptions 3.1, 3.2, and 4.1. Then, Algorithm 3” recovers $\phi$ on segments with dwell times satisfying $\tilde{\delta}(\chi) > 4n$, $\delta(\chi) \geq 2n + 1$, for $0 < i < t^*$, and $\delta(\chi) \geq 6n$ as $\varepsilon \rightarrow 0$.

In passing, among the recovery conditions the least stringent is the one in Theorem 6.3. The next one is stated in Theorem 5.3. When Assumption 4.2 holds, any one of Algorithms 3–3” may be combined with Algorithm 2 to form a meta-algorithm since all switch detection conditions will automatically be satisfied by Algorithm 2. Therefore, a meta algorithm may include them all. In Section 9.1.2 this will be demonstrated in a numerical example.
The choice of a basis for the discrete state estimates requires attention if the estimated SLS model will be used for predicting outputs to prescribed input sequences. To understand how this issue arises, re-write the Markov parameters in (6)

\[ h(k, l) = C(k)A(k - 1) \cdots A(l + 1)B(l), \quad k > l + 1, \quad (106) \]

\[ h(k, k - 1) = C(k)B(k - 1), \quad \text{and} \quad h(k, k) = D(k). \]

Recall that the discrete state estimates in \( \tilde{\mathcal{P}} \) are similar to the true ones in \( \mathcal{P} \), that is, for every \( \tilde{\mathcal{P}} \in \tilde{\mathcal{P}} \), there exists a \( \mathcal{P} \in \mathcal{P} \) and a \( T_j \in \mathbb{R}^{n \times n} \) satisfying \( A_j = T_j^{-1} \hat{A}_j T_j, \quad C_j = \hat{C}_j T_j, \quad B_j = T_j^{-1} \hat{B}_j, \quad \text{and} \quad D_j = \hat{D}_j \), where \( \tilde{\mathcal{P}} = (A_j, B_j, C_j, D_j) \) and \( \mathcal{P} = (\hat{A}_j, \hat{B}_j, \hat{C}_j, \hat{D}_j) \). In (106), plug \( k = k_1 + \xi, \quad 0 \leq \xi \leq n \) and \( l = k_1 - \eta, \quad 1 \leq \eta \leq n \) in. Suppose \( \mathcal{P}_{j_1} \) and \( \mathcal{P}_{j_2} \) are the active discrete states on \( [k_{i-1}, k_i) \) and \( [k_i, k_{i+1}) \), respectively. Assuming \( \delta_\iota(\chi) \geq 2n \) and using the similarity relations for \( \mathcal{P}_{j_1} \) and \( \mathcal{P}_{j_2} \), we derive

\[ h(k_i + \xi, k_i - \eta) = \tilde{C}_{j_2} \tilde{\Delta}_{j_2} \tilde{A}_{j_2}^{-1} \tilde{T}_{j_1}^{-1} \tilde{A}_{j_1}^{\eta-1} \tilde{B}_{j_1}. \quad (107) \]

This equation covers all possibilities for \( T_{j_2} T_{j_1}^{-1} \) since it is at the leftmost position when \( \xi = 0 \) and at the rightmost position when \( \eta = 1 \). Note that \( T_{j_2} T_{j_1}^{-1} \) appears only once in the string of state-space matrices since \( \xi + \eta \leq 2n \).

Fix \( \eta \geq 1 \), evaluate (107) for \( \xi \) between 1 to \( n \), and concatenate the resulting equations to derive

\[ Z_{1, \eta} \overset{\Delta}{=} \begin{bmatrix} h(k_i + 1, k_i - \eta) \\ \vdots \\ h(k_i + n, k_i - \eta) \end{bmatrix} = \hat{\Theta}_{n}(j_2) X_\eta \quad (108) \]

where \( \hat{\Theta}_{n}(j_2) \) is the extended observability matrix constructed from the pair \( \hat{C}_{j_2} \) and \( \hat{A}_{j_2} \) and

\[ X_\eta = \hat{A}_{j_2} \tilde{T}_{j_2} T_{j_1}^{-1} \hat{A}_{j_1}^{\eta-1} \hat{B}_{j_1}. \quad (109) \]

Then, from (108)

\[ X_\eta = \hat{\Theta}_{n}^{\dagger}(j_2) Z_{1, \eta}, \quad \eta = 1, \ldots, n. \quad (110) \]

Evaluate (109) for \( \eta = 1, \ldots, n \) and concatenate them to get

\[ [X_1 \cdots X_n] = \hat{A}_{j_2} T_{j_2} T_{j_1}^{-1} \hat{\Theta}_{n}^{\dagger}(j_1) \quad (111) \]

where \( \hat{\Theta}_{n}^{\dagger}(j_1) \) is the extended controllability matrix constructed from the pair \( \hat{A}_{j_1} \) and \( \hat{B}_{j_1} \). Then,

\[ \hat{A}_{j_2} T_{j_2} T_{j_1}^{-1} = [X_1 \cdots X_n] \hat{\Theta}_{n}^{\dagger}(j_1) \overset{\Delta}{=} Y_{i,i-1}. \quad (112) \]

Assuming \( A_{j_2} \) has no eigenvalues at the origin, we get \( T_{j_2} T_{j_1}^{-1} = \hat{A}_{j_2}^{-1} Y_{i,i-1} \). We derive \( T_{j_1} T_{j_2}^{-1} = Y_{i,i-1}^{-1} \hat{A}_{j_2} \) on inverting this equation. Suppose we want to estimate \( T_{j_1} T_{j_1}^{-1} \). Write it as a product

\[ T_{j_1} T_{j_2}^{-1} = (T_{j_2} T_{j_1}^{-1})(T_{j_2} T_{j_1}^{-1}) \]

which is readily calculated if we know the products. At our disposition, we have Assumption 4.2 which tells us that every discrete state can be reached from another
by crossing a finite number of the switches. Since there are only \( \sigma \) discrete states, we start from \( \tilde{P}_0 \), estimate \( T_\sigma T_{\sigma-1}^{-1} \), move to \( T_{\sigma-1} T_{\sigma-2}^{-1} \), and so on. This procedure is repeated \( \sigma - 1 \) times. Some steps may not be implemented directly. In this case, a path is chosen and the inversion and the product formulas are applied.

We define a basis transform by letting \( \tilde{A}_j = \Pi_j^{-1} \tilde{A}_j \Pi_j, \tilde{B}_j = \Pi_j^{-1} \tilde{B}_j, \tilde{C}_j = \tilde{C}_j \Pi_j, \tilde{D}_j = \tilde{D}_j \) where \( \Pi_j = T_j T_j^{-1} \) and setting \( \tilde{\mathcal{J}}_j = (\tilde{A}_j, \tilde{B}_j, \tilde{C}_j, \tilde{D}_j) \). Rewrite (107) with the state-space matrices of \( \mathcal{J}_j \) and \( \tilde{\mathcal{J}}_j \):

\[
 h(k_t + \xi, k_t - \eta) = \tilde{C}_j \tilde{A}_j^T \Pi_j T_j T_j^{-1} \Pi_j \tilde{A}_j^T \tilde{B}_j.
\]

But,

\[
\tilde{C}_j = \hat{C}_j, \Pi_j = \hat{C}_j T_j T_j^{-1} T_j T_j^{-1} = C_j T_j^{-1},
\]

\[
\tilde{A}_j = \Pi_j^{-1} \hat{A}_j \Pi_j = T_j T_j^{-1} T_j T_j^{-1} T_j T_j^{-1} = T_j A_j T_j^{-1},
\]

\[
\tilde{B}_j = \Pi_j^{-1} \hat{B}_j = T_j T_j^{-1} T_j B_j = T_j B_j.
\]

Replacing back leads to \( h(k_t + \xi, k_t - \eta) = C_j A_j^T A_j^{\eta-1} B_j \). Hence, the Markov parameters of (2)–(4) are matched as desired. Since \( \Pi_1 = I_n \), there is no need to operate on \( \mathcal{J}_1 \). Clearly, this matching is applicable to any pair of the discrete states and the Markov parameters with large lags exhibiting several switches. The transformed discrete states and the switching signal may be used to calculate the response of the system to arbitrary inputs. The steps of this transformation is formalized as Algorithm 4 in Appendix. Time complexity of \( \Pi_1 \) is \( O(poly(n)) \) for each \( j \). The worst-case time complexity of Algorithm 4 is then \( O(Npoly(n)) \) and it happens when the switches are uniformly distributed in \( \chi \) and \( j \) assumes values from a set with cardinality \( O(Nn^{-1}) \). Before summarizing the result derived in this section, we state the sole requirement as follows.

**Assumption 7.1.** The discrete states of the MIMO–SLS model (2)–(4) have no poles at zero.

**Theorem 7.1.** Suppose Assumption 7.1 holds and \( \delta_n(\chi) \geq 2n \). Then, Algorithm 4 solves the basis transformation problem for the MIMO–SLS model (2)–(4).

8. From Markov parameter estimates to MIMO-SLS models

In this section, we combine the results derived in Sections 2–7 to form a meta-algorithm to identify the MIMO-SLS model (2)–(4) from its noisy Markov parameters. The steps of this meta-algorithm are illustrated in Figure 1 as a flowchart. We collect the results derived in Sections 3–7 in the following main result.

**Theorem 8.1.** Consider (2)–(4) with the noisy Markov parameters in (31). Assume that \( \chi \) and \( \mathcal{P} \) satisfy Assumptions 3.1–3.2, 4.1, 5.1, 6.1 and 7.1. If Assumption 4.2 holds, then the meta-algorithm recovers \( \mathcal{P} \) and \( \chi \) as \( \varepsilon \to 0 \). Suppose that the first part of Assumption 4.2 holds and \( \varphi \) satisfies a mixing condition. If each segment in \( \chi \) satisfies a dwell requirement in one of Theorems 5.1, 6.1 and 6.3, then the meta-algorithm recovers \( \mathcal{P} \) and \( \chi \) as \( \varepsilon \to 0 \).
In all algorithms we developed so far, observability and controllability concepts have played a foundational role. We have assumed that the discrete states have a common Macmillan degree. When this assumption is dropped, identifiability, controllability, and observability for the MIMO-SLS models become much harder to analyze. For example, identifiability of an SLS does not imply identifiability of its submodels [62]. Pivotal role played by observability in the
estimation/realization of the SLSs has been noted early in hybrid system literature \[30\]. The observability part of Lemma 3.1 appeared as Theorem 2 in \[25\]. Extension of the Ho-Kalman realization theory for the LPV state-space models was reported in \[48\].

This paper did not address the issue of estimating Markov parameters from input-output data. The reasons are twofold. First, the realization problem for SLSs is interesting in its own \[21\], \[22\], \[63\], \[64\]. See also the references therein. An equally important subject in SLSs is model order reduction. We refer to recent works \[65\], \[66\], \[67\]. Second, we decoupled identification and realization problems and treated the latter in great detail in this paper though both can be treated in a unified manner. This approach has several merits. Identification of an SLS is normally conducted in two stages: estimation of the discrete states and detection of the switches, with the order being interchangeable. The realization problem solved in this paper recovers the switching sequence in addition to the discrete states. The switching sequence is an arbitrary time-varying signal subject to mild restrictions in contrast to existing works in hybrid systems literature. By lifting the issue of estimating the switching sequence to the realization stage, attention is directed to estimating the Markov parameters in parsimonious models.

One approach for estimating Markov parameters is to use multiple trajectories when it is possible \[28\], \[55\], \[56\]. But, this approach has limited applications to system identification. Markov parameter estimation problem is not tractable for generic LTV systems due to the curse of dimensionality, unless further system assumptions such as slowly varying or smooth, mode switching with long waiting times, etc. are made. In \[14\], switched ARX models were identified by a kernel based estimation method. This method encompasses the parameter estimation and the switch detection stages by solving an optimization problem. Recent work \[35\] used deadbeat observers to transform state-space identification problem into an SARX identification problem. This transformation generates parsimonious models by packing infinite strings of Markov parameters into finite sequences. In principle, any of these methods can be used to estimate Markov parameters. Work is under progress for direct estimation of the Markov parameters from input-output data.

9. Numerical example

Consider the following MIMO–SLS state-space model adapted from \[34\]

\[
A_1 = \begin{bmatrix} 0.15 & 0.40 & -0.65 \\ -0.75 & 0.1 & -0.35 \\ 0.20 & 0.70 & 0.20 \end{bmatrix}, \quad B_1 = \begin{bmatrix} -0.20 & 0.45 \\ -0.06 & 0 \\ 0.22 & 0 \end{bmatrix},
\]

\[
C_1 = \begin{bmatrix} 0 & 0.40 & 0.45 \\ 0 & -0.60 & 0.90 \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0 & -0.35 \\ -1.70 & -0.25 \end{bmatrix},
\]

\[
A_2 = \begin{bmatrix} 0.27 & 0.24 & -0.55 \\ 0.24 & 0.65 & 0.30 \\ -0.55 & 0.30 & 0.27 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -0.55 & 0 \\ -1.40 & 1 \\ 0.05 & -0.72 \end{bmatrix},
\]

\[
C_2 = \begin{bmatrix} 0.70 & 1 & -0.27 \\ -0.35 & 0 & -1.10 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 2.15 & 0.25 \\ 0 & -0.36 \end{bmatrix},
\]

30
\[
A_3 = \begin{bmatrix}
0.45 & 0.02 & 0.42 \\
-0.17 & 0.53 & 0.20 \\
0.38 & 0.26 & 0
\end{bmatrix},
B_3 = \begin{bmatrix}
0 & 0.15 \\
0.27 & -0.46 \\
0.07 & 0.54
\end{bmatrix},
C_3 = \begin{bmatrix}
0 & 0.60 & 0.28 \\
0 & 0.86 & 0.45
\end{bmatrix},
D_3 = \begin{bmatrix}
0 & -0.90 \\
0 & 0.85
\end{bmatrix},
\]

where the problem of identifying a state-space SLS model for the case when the continuous state is known was treated. In the sequel, we address the problem of discrete state estimation given an exact sequence of Markov parameters.

9.1. SLS estimation in a noiseless setup

We first deal with the estimation of the discrete-states.

9.1.1. Discrete state estimation

A switching sequence that conforms with Assumption 4.2 in all but few segments was generated via sampling from a random uniform distribution. It is shown in Figure 2 where the red dots represent the switches. The discrete state estimation was accomplished via Algorithm 2. As inputs to the algorithm, we selected \( \varepsilon_Z = 10^{-4} \), \( \nu = 6 \), and estimated the state-space quadruples \( \hat{P}_k \) over the time range \( k' \leq k \leq k'' \) from Algorithm 1. The \texttt{dbscan} command in MATLAB was implemented by selecting \( \text{epsilon}=10^{-5} \) as a threshold value for the neighborhood search radius and \( \text{minpts}=1 \) for the minimum number of neighbors. As an entry to \texttt{dbscan}, we supplied \( M(\hat{\Theta}_i) \) as the clustering feature where \( \gamma_i \) is the midpoint of the targeted intervals which satisfy \( \beta_i - \alpha_i \geq \nu n \). Let \( I = \{i : \beta_i - \alpha_i \geq \nu n\} \) denote the set of indices satisfying this constraint. Figure 3a shows the clustering result over \( M(\hat{\Theta}_i) \), \( i \in I \). As anticipated, \( \sigma \) is correctly estimated, i.e., \( \hat{\sigma} = 3 \). Using clusters, we now determine the set of submodel estimates \( \hat{P}_j \), \( j \in S \). Notice that not all segments satisfy \( \beta_i - \alpha_i \geq \nu n \), yet recovery is perfect.

To assess the fitting capability of the estimates, the submodel estimates’ eigenvalues were compared to the true ones as shown in Figure 3b. Observe the exact matches to the true eigenvalues.

Figure 2: The switching sequence in the example.
9.1.2. Switching sequence estimation

Now we address the issue of switching sequence estimation assuming that the submodel set was estimated in the previous stage. This could be achieved via Algorithm 3’ solely if our SLS has a minimum dwell time of at least $6n + 1$, i.e., $\delta_i(\chi) \geq 6n + 1$. On the other hand, Algorithm 3 can be used if $\delta_i(\chi) \geq 4n + 2$. Lastly, if there exist very short segments with a minimum dwell time of at least $\delta_i(\chi) \geq 2n + 1$, we use Algorithm 3’’. We call a segment $[k_i, k_{i+1})$ satisfying $\delta_i(\chi) \geq 6n + 1$ ’medium-to-long’, a segment $[k_i, k_{i+1})$ with $6n + 1 > \delta_i(\chi) \geq 4n + 2$ ’short’, and lastly a segment $[k_i, k_{i+1})$ satisfying $4n + 2 > \delta_i(\chi) \geq 2n + 1$ ’very short’. Algorithms (3’, 3’’, 3’’) estimate the medium-to-long, the short, and the very short segments, respectively. To show their working mechanisms, we generated a switching sequence containing segments of the three types. Different type segments were marked with distinctive colors for better discernibility: medium-to-long ’blue’, short ’brown’, and very short ’black’. The switching sequence generated is shown in Figure 4a. In Figure 4b, the segments in $\chi$ are classified according to their types.

Figure 4: The switching sequence generated in Section 9.1.2.
We start by estimating the submodels, and this is accomplished via Algorithm 2 with $\nu = 6$. Next, we sequentially run the switch detection algorithms. We first try Algorithm 3' over the medium-to-long segments, but it fails to estimate the switching sequence at some points in the short and the very short segments. This explains why a large set of points are not yet attributed to any of the submodels as shown in Figure 5a. These points were put in Cluster 0 to distinguish them from the points in the segments which have already been attributed a correct submodel index. The red dots are the switches as before. Next, we try Algorithm 3 to estimate the switching sequence over the short segments, but it fails at some points in the very short segments. The switching sequence estimate is shown in Figure 5b. Finally, we run Algorithm 3'' to estimate the rest of the switching sequence, i.e., over the very short segments. The final switching sequence estimate and its zoomed image are shown in Figure 5c and Figure 7. It is an exact match to the true one. As a final remark, note that Algorithm 3' operates on the intervals satisfying $\beta_i - \alpha_i \geq \nu n$. This condition was used in Algorithm 2 to estimate the discrete states.

9.1.3. Basis transform for the discrete states

This subsubsection is concerned with performing the necessary discrete state transformations to render the SLS model usable for predicting output given an input signal. To perform that, one needs the estimated sub-models from Algorithm 2 and the estimated switching sequence from Algorithms (3, 3', 3''). Having that, we employ Algorithm 4 to obtain $\Pi_j = T_j T_1^{-1}$ for $j = 2, 3$. They were calculated as follows

\[
\Pi_2 = \begin{bmatrix}
-0.4549 & 0.0311 & 0.1299 \\
-0.0140 & 0.5774 & -0.2281 \\
-0.4472 & -0.3224 & -0.8538
\end{bmatrix},
\]

\[
\Pi_3 = \begin{bmatrix}
-0.4595 & 0.1516 & 0.9003 \\
-0.0115 & -0.1930 & 0.2244 \\
-0.2481 & -0.1360 & -0.0860
\end{bmatrix}.
\]

After that, we apply $\Pi_j$ to $\hat{\phi}$ to put all the submodels in a common state basis. In particular, we apply $\Pi_2$ to $\hat{\phi}_2$ to render it in a common state basis as of that corresponding $\hat{\phi}_1$. In a similar manner, $\Pi_3$ is applied to $\hat{\phi}_3$ which transforms its basis to that of $\hat{\phi}_1$. To demonstrate the effectiveness of this approach, we generated an input signal as a sum of harmonics, and we simulated its output through the SLS model. We did that both for the original SLS and the estimated one, and we inspected the match between the two. In generating the output signal we assumed that there were no switches occurring before $k'$ and after $k''$, hence we extended $\hat{\phi}(k)$ to 1 and $N$. Both the true and the modeled SLSs were started at rest, i.e., the initial state was taken to be $x_0 = [0 \ 0 \ 0]^T$. Figure 7 displays the true output signals and the estimation errors side by side.

To assess the fidelity of these estimates, we introduce the variance-accounted-for criterion defined by

\[
VAF = \left(1 - \frac{\text{var}(\hat{\gamma}(k))}{\text{var}(\gamma(k))}\right) \times 100\%.
\]

We calculated (113) for both outputs. For Output 1, we got VAF=99.99% and for Output 2, VAF=99.99% indicating perfect match between the two. This perfect match resulted from the agreement between the pairs true switching sequence-Markov parameter set and their estimates since the I/O map explicitly depends on them.
9.2. SLS recovery from noisy Markov parameters measurements

We now study SLS recovery in a noisy setup. The same SLS model as in the noiseless case was adopted here. Table I lists the smallest nonzero singular values in the noiseless Hankel matrices of the submodels, which according to [45] determine how robustly a realization algorithm
Figure 6: The true ‘o’ and the estimated ‘-’ switching sequences zoomed from Figure 5c.

Figure 7: The outputs and the estimation errors.

(a) True Output 1 (left) and the estimation error (right).

(b) True Output 2 (left) and the estimation error (right).

Figure 7: The outputs and the estimation errors.
could retrieve a particular discrete state. It is clear from the table that Submodels 1 and 2 can be learned better than Submodel 3 when the exact Markov parameters pertaining to these submodels are perturbed with the same noise level.

Table 1: The smallest nonzero singular values in the Hankel matrices of the individual submodels.

| Submodel | $\mathcal{P}_1$ | $\mathcal{P}_2$ | $\mathcal{P}_3$ |
|----------|----------------|----------------|----------------|
| $\sigma_{\min}$ | 0.4063 | 0.3560 | 0.0180 |

We generated a switching sequence by sampling from a uniform distribution. Instead of using the exact Markov parameters, we contaminated them via an additive white noise. From the noisy doubly indexed Markov parameters, the SLS identification was conducted similarly to the previous case. Algorithm 2 recovered the submodels. These submodels were exploited to retrieve the switching sequence by running Algorithms (3, 3’, 3’’) depending on the type of segment considered. Algorithm 4 transformed the discrete states estimated in a precedent stage to a common state basis so that the output prediction could be conducted harmlessly. This could be as well inspected by checking that the true Markov parameters and the estimated ones are identical through the entire time interval. The individual steps are summarized in the flowchart of Figure 1. Figure 8 plots the feature used for clustering defined in (64) together with the switching sequence for one noise realization with signal-to-noise ratio (SNR) 40dB. Note the gross error in the last segment in agreement with Table I, that is, the realization algorithm performs poorly when it tries to learn Submodel 3. The estimated SLS eigenvalues shown in Figure 9 highly mismatch the eigenvalues of Submodel 3.

To study robustness of the meta-algorithm to noise, we calculated the mismatch error in estimating the true Hankel matrix in (9) defined pointwise in time by

$$
\varepsilon_{H}(k) = \| \mathcal{H}_{2n+1,2n}(k) - \hat{\mathcal{H}}_{2n+1,2n}(k) \|_F, \quad \forall k \in \left[ k', k'' \right]
$$

where $\hat{\mathcal{H}}_{2n+1,2n}(k)$ is the Hankel matrix estimate. Since the SLS realization problem involves both $\varphi$ and $\mathcal{P}$, we selected a criterion reflecting both through Markov parameters. Alternatively, we could have conducted output predictions to prescribed inputs. Figure 10 displays the mismatch error (114) as a function of time for a range of SNRs. The mismatch error is flat over the segments with the exception of the switches where the changes in the discrete states are abrupt. As SNR is decreased, the mismatch error increases rapidly.

9.3. Realization from input-output data

In this subsection we demonstrate how our meta-algorithm could be incorporated in a complete system identification package that estimates an SLS starting from the input-output measurements rather than the Markov parameters. To achieve this goal, we append our meta-algorithm to an identification algorithm proposed in [35] to form a two-stage algorithm. The algorithm in [35] is a standalone algorithm which can be used to fully estimate an SLS in the state-space form from the input-output data. It can also be used to estimate the Markov parameters by solving a sparse optimization problem. We emphasize that this algorithm successfully retrieves the Markov parameters from a single trajectory, unlike the earlier works [55, 56, 28] where multiple trajectories are assumed to be available.

Solution of sparse optimization problem essentially grants us a finite set of observer Markov parameters which encodes entire knowledge of the system dynamics. From observer parameters,
one can retrieve infinite strings of the Markov parameters. Without going into technical details,
it suffices to say that conditions on the input-output data, the dwell times, persistence of excitation on the inputs, and some identifiability criteria permit successful recovery of the Markov parameters. The meta-algorithm of this paper complements this scheme. Let us consider the SLS model

\begin{figure}[h]
\centering
\begin{subfigure}{0.4\textwidth}
\includegraphics[width=\textwidth]{a.png}
\caption{SNR=45dB.}
\end{subfigure}
\begin{subfigure}{0.4\textwidth}
\includegraphics[width=\textwidth]{b.png}
\caption{SNR=40dB.}
\end{subfigure}
\begin{subfigure}{0.4\textwidth}
\includegraphics[width=\textwidth]{c.png}
\caption{SNR=35dB.}
\end{subfigure}
\caption{The mismatch error at different SNRs.}
\end{figure}
We generated a switching sequence of length $N = 1,000$ by sampling from a uniform distribution. The SLS model was excited with a multi-sine input starting from the initial condition $x(0) = (0 0)$. Running Algorithm 6 in [35] by using the CVX package [68] with the observer order $\tau = 2n$, we get the observer Markov parameter estimates $h(k, k - \ell)$ for $2n < k \leq N - 2n$ and $0 \leq \ell < 2n$. Let us call Algorithm 1 in [35] Algorithm 1’ to avoid confusion with Algorithm 1 in this paper. Algorithm 1’ driven by the observer Markov parameters returns $h(k, k - \ell)$ for $0 \leq \ell \leq k$ and $2n < k \leq N - 2n$. Algorithm 1 relies only on the first $4n + 1$ Markov parameters.

Thus, running Algorithm 1’ we get as many Markov parameter estimates as we want. Having estimated the Markov parameters, we run the meta-algorithm. We will perform a Monte-Carlo simulation study for 100 noise realizations.

The eigenvalues estimated by this two-stage scheme are plotted in Figure 11. The average case performance will be assessed by first computing the relative errors $\|M_j - \hat{M}_j\|_F / \|M_j\|_F$, $j \in S$ where $\hat{M}_j = (\hat{A}_j, \hat{B}_j, \hat{C}_j, \hat{D}_j)$ and

$$M_j = \begin{bmatrix} \hat{A}_j & \hat{B}_j \\ \hat{C}_j & \hat{D}_j \end{bmatrix},$$

summing over $S$, and averaging over 100 realizations. The result is denoted by $\delta$. In Table 2, $\delta$ is displayed for 20, 30, and 40dB SNRs. In the same table we compared the two stage scheme reported here to the algorithm in [34], which assumes the availability of state measurements and proceeds by solving a sparse SARX regression optimization problem. However, the availability of the state measurements is a stringent assumption in practice.
Table 2: $\delta_\varphi$ calculated over the 100 noise realizations.

| Schemes | $\delta_\varphi$ | SNR (dB) |
|---------|------------------|----------|
| Proposed | 0.0169 0.0297 0.0839 | 40dB 30dB 20dB |
| [34]    | 0.0104 0.0404 0.1581 |

A measure of fit for the switching sequence estimates is the percentage of correctly classified points. It is called FIT$_\varphi$ and calculated by the formula

$$FIT_\varphi = \left( 1 - \frac{1}{N} \sum_{k=1}^{N} \text{sign} |\hat{\varphi}(k) - \varphi(k)| \right) \times 100\%.$$  \hspace{1cm} (115)

Figure 12 plots the histogram of FIT$_\varphi$ over 100 noise realizations at SNR=30dB for both our approach and the scheme in [34]. From Table 2 and Figure 12 we see that the two-stage scheme outperforms the algorithm in [34] and have an edge in the sense that it operates under mild assumptions.

9.4. Realization of randomly generated SLSs

In this subsection, we examine the performance of the meta-algorithm in estimating randomly generated SLSs. The discrete states were sampled from a normal distribution. The switching sequences were generated by sampling from a uniform distribution similarly to the previous subsections. The SLSs and the switching sequences satisfy Assumptions 3.1–3.2, 4.1, 5.1, 6.1, and 7.1. Randomly generated SLSs were corrupted by noise to achieve a certain SNR prior to carrying out identification. We then generated the Markov parameters to drive the meta-algorithm. In this study, we considered SISO-SLSs with $n = 2$ and $\sigma = 3$. We picked $N = 650$
for Monte-Carlo simulations and assessed the performance of our algorithm using first $\delta_\varphi$ and $\text{FIT}_\varphi$. The results in Table 3 were obtained by first estimating an SLS and $\varphi$ for each run and averaging the sum of the relative errors and $\text{FIT}_\varphi$ over 50 runs. For each run, we picked a random SLS and a random noise realization. We repeat calculations for a range of SNRs to fill the table. As another measure of performance, for each run next we calculated the RMS value of (114) over $k$ and averaged it over 50 runs. The results are plotted in Figure 13 as a function of SNR levels.

| Errors | $\delta_\varphi$ | $\text{FIT}_\varphi$ (%) |
|--------|----------------|--------------------------|
| SNR (dB) | 50dB | 40dB | 30dB | 20dB | 50dB | 40dB | 30dB | 20dB |
| Average Values | 0.0110 | 0.0423 | 0.1818 | 0.7651 | 100 | 99.98 | 98.20 | 94.54 |

Figure 13: The average RMS of the error in (114) calculated over 50 runs at different SNRs.

10. Conclusions

This paper laid forward a four-stage algorithm for the realization of MIMO-SLSs from Markov parameters. Its first stage recovers an LTV realization that is topologically equivalent to the SLS under mild system and dwell time assumptions. Since topological equivalence does not guarantee similarity to the discrete states on any segment, it was necessary to apply forward/backward corrections point-wise in time to the LTV realization, to reveal the segments. This process established a time-varying similarity to the discrete states. Next, the stationary point set on which the LTV realization has an LTI pulse response were sought. This search was effected by a clustering algorithm using a feature space that is invariant to similarity transformations. Combination of these schemes formed Stage 2 of the realization algorithm. In this stage, the discrete state estimates similar to the true ones were extracted from the LTV realization under some dwell time and identifiability conditions. In the third stage, the switching sequence
was estimated by three schemes, complementing each other. The first scheme operates on the forward/backward corrections and estimates the switching sequence over short segments. The second scheme is based on matching the estimated and the true Markov parameters for LTV systems and it can be applied to medium-to-long segments. The third scheme is also based on matching the Markov parameters, but for LTI systems only via Hankel matrix factorization. It can be applied to segments of even shorter length which cannot be handled by the first two schemes. The schemes are applied in the order second-to-first-to-third. The issue of discrete states residing in different state bases were resolved by applying a basis correction in Stage 4; this equips the retrieved SLS with the potential to conduct output predictions.

The success of the various schemes in the realization algorithm put us face to face with the compulsion to impose several restrictions on the discrete state set and their dwell times; nonetheless, all that can be said is that they are mild in nature. The involvement of the observability and controllability matrices in the schemes presented so far was quite noticeable, as noted in several other related works in the literature on SLS identification and realization. Future work will focus on developing an identification algorithm to estimate the Markov parameters from input-output measurements of a single trajectory.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank professor Laurent Bako for providing the codes for his algorithm in [34].

References

[1] R. Rui, T. Ardeshiri, and A. Bazanella, “Identification of piecewise affine state-space models via expectation maximization,” in: Proceedings of the IEEE Conference on Computer Aided Control System Design, Buenos Aires, Argentina, pages 1066–1071, September 2016.
[2] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, “Identification of hybrid systems a tutorial,” European Journal of Control, vol. 13, no. 2-3, pp. 242–260, 2007.
[3] R. Vidal, “Recursive identification of switched ARX systems,” Automatica, vol. 44, no. 9, pp. 2274–2287, 2008.
[4] N. Ozay, M. Szaier, C. M. Lagoa, and O. I. Camps, “A sparsification approach to set membership identification of switched affine systems,” IEEE Transactions on Automatic Control, vol. 57, no. 3, pp. 634–648, 2011.
[5] N. Ozay, C. Lagoa, and M. Szaier, “Set membership identification of switched linear systems with known number of subsystems,” Automatica, vol. 51, pp. 180–191, 2015.
[6] X. Jin and B. Huang, “Robust identification of piecewise switching autoregressive exogenous process,” AICHe Journal, vol. 56, no. 7, pp. 1829–1844, 2010.
[7] R. Murray-Smith, “Modelling human control behaviour with context-dependent Markov-switching multiple models,” IFAC Proceedings Volumes, vol. 31, no. 26, pages 461–466, 1998.
[8] A. Mestre, Hybrid subspace identification: an application to HIV infection, PhD thesis, Universidade Técnica de Lisboa-Instituto Superior Técnico, 2010.
[9] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, “A bounded-error approach to piecewise affine system identification,” IEEE Transactions on Automatic Control, vol. 50, pp. 1567–1580, 2005.
[10] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, “A clustering technique for the identification of piecewise affine systems,” Automatica, vol. 39, no. 2, pp. 205–217, 2003.
[11] A. L. Juloski, S. Weiland, and W. Heemels, “A Bayesian approach to identification of hybrid systems,” IEEE Transactions on Automatic Control, vol. 50, pp. 1520–1533, 2005.
[12] Z. Lassoued and K. Abderrahim, “An experimental validation of a novel clustering approach to PWARX identification,” Engineering Applications of Artificial Intelligence, vol. 28, pp. 201–209, 2014.
[13] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, “An algebraic geometric approach to the identification of a class of linear hybrid systems,” in: Proceedings of the 42nd IEEE International Conference on Decision and Control, Maui, HI, pages 167–172, December 2003.
the Loewner framework,” SIAM Journal on Scientific Computing, vol. 40, pp. B572–B610, 2018.

[68] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming,” Version 2.1, 2014.
Appendix: Pseudo codes for the algorithms

Algorithm 1. LTV realization from Markov parameters

Set \( q = 2n + 1 \) and \( r = 2n \).

**Inputs:** Markov parameters \( h(k + i, k - j) \) for \( 0 \leq i < q, 1 \leq j \leq r, k \in [k', k''] \)

while \( k \in [k', k''] \)

1: Compute \( \mathcal{H}_{q,r}(k) \) and \( \mathcal{H}_{q,r}(k + 1) \) from (29)–(30)
2: Compute the SVDs in (20)
3: Compute the extended observability/controllability matrices estimates in (21-22)
4: Estimate \( \hat{A}(k), \hat{C}(k), \hat{B}(k) \) from (23) and (27), and \( \hat{D}(k) = h(k, k) \)
5: \( k \leftarrow k + 1 \)
end while

**Outputs:** \( \hat{P}(k) = (\hat{A}(k), \hat{B}(k), \hat{C}(k), \hat{D}(k)) \), \( \forall k \in \mathcal{K} \).

Algorithm 2. Discrete state estimation

**Inputs:** Markov parameters \( h(k + i, k - j) \) for \( 0 \leq i < 2n + 1, 1 \leq j \leq 2n, k \in [k', k''] \), a small \( \varepsilon_Z > 0 \), and \( \nu > 5 \).

1: Compute \( \delta_H(k) \), for all \( k \in [k', k''] \) from (62)
2: Initialize \( Z_{H,\varepsilon_Z} = \phi \)
3: while \( k \in [k', k''] \)
   if \( \| \delta_H(k) \|_F \leq \varepsilon_Z \)
      \( Z_{H,\varepsilon_Z} = Z_{H,\varepsilon_Z} \cup \{ k \} \)
   end if
   \( k \leftarrow k + 1 \)
end while

4: Identify maximal disjoint intervals \( [\alpha_i, \beta_i] \subset Z_{H,\varepsilon_Z}, \beta_i - \alpha_i \geq \nu n \)
5: Retrieve \( \hat{A}(\gamma), \gamma = (\alpha_i + \beta_i)/2 \) from the output Algorithm 1 .
6: Compute \( \mathcal{H}(\hat{A}(\gamma)) \)
7: Estimate \( \hat{P} \) by running the clustering algorithm in [59] over \( \gamma \) and re-clustering if necessary

**Outputs:** \( \hat{P}_j, j \in \mathcal{S} \) and the intervals in Step 4 containing them.
Algorithm 3. Switch detection based on corrections

Inputs: $S_i = [\alpha_i, \beta_i], 0 \leq i \leq i^*; h(k+i,k-j), 0 \leq i < 2n+1, 1 \leq j \leq 2n, k \in [k', k'']$

Run the following loop:

for $0 \leq i \leq i^*$

if $(i == 0$ and $\delta_i(\chi) \geq 6n + 2)$ or $(i \neq 0, i \neq i^*$, and $\delta_i(\chi) \geq 4n + 2)$

Initialize $k = \beta_i - 2n$

while $|\cdot(\chi_{2n+1}(k)) - n| < \epsilon$

$k = k + 1$

end while

$k_{i+1} = k + 2n + 1$

end if

if $(i == i^*$ and $\delta_i(\chi) > 8n$ or $(i \neq 0, i \neq i^*$, and $\delta_i(\chi) \geq 4n + 2)$

Initialize $k = \alpha_i + 2n - 1$

while $|\cdot(\chi_{2n}(k)) - n| < \epsilon$

$k = k - 1$

$k_i = k - 2n + 1$

end while

end if

end for

Outputs: $\phi(k)$ on segments satisfying $\delta_i(\chi) \geq 4n + 2$

Algorithm 3'. Switch detection from Markov parameters

Inputs: $\mathcal{H}(\gamma), \gamma = (\alpha + \beta)/2, S_i = [\alpha_i, \beta_i], 0 \leq i \leq i^*$; $h(k+i,k-j), 0 \leq i \leq 2n, 1 \leq j \leq 2n, k \in [k', k''], \epsilon > 0$

Run the following loop:

for $0 \leq i \leq i^*$

if $(i == 0$ and $\delta_i(\chi) \geq 8n + 1)$ or $(i \neq 0, i \neq i^*$, and $\delta_i(\chi) \geq 6n + 1)$

Initialize $\ell = \beta_i - 1, \mathcal{C} = \mathcal{C}(\gamma), \mathcal{D} = \mathcal{D}(\gamma)$

while $||\mathcal{C} - \mathcal{C}(\gamma)||_F + ||\mathcal{D} - \mathcal{D}(\gamma)||_F < \epsilon$

$\ell = \ell + 1$

$\mathcal{C} = \mathcal{H}^{2n}(\ell)\mathcal{H}_{2n}(\ell - 1), \mathcal{D} = \mathcal{D}(\ell)$

end while

Set $k_{i+1} = \ell$

end if

if $(i == i^*$ and $\delta_i(\chi) \geq 10n$ or $(i \neq 0, i \neq i^*$, and $\delta_i(\chi) \geq 6n + 1)$

Initialize $\ell = \alpha_i + 1, \mathcal{B} = \mathcal{B}(\gamma), \mathcal{D} = \mathcal{D}(\gamma)$

while $||\mathcal{B} - \mathcal{B}(\gamma)||_F + ||\mathcal{D} - \mathcal{D}(\gamma)||_F < \epsilon$

$\ell = \ell - 1$

$\mathcal{B} = \mathcal{B}_{2n}(\ell + 1), \mathcal{H}_{2n,1}(\ell + 1), \mathcal{D} = \mathcal{D}(\ell)$

end while

Set $k_{i+1} = \ell + 1$

end if

end for

Outputs: $\phi(k)$ on segments satisfying $\delta_i(\chi) \geq 6n + 1$
Algorithm 3'. Switch detection on short intervals

Inputs: \( \hat{\bar{\mathcal{P}}} \), one switch from each segment, Markov parameters, and \( \varepsilon > 0 \)

Run the following loop:

for \( 1 \leq i < i' \)

while \( \delta_i(\chi) > 2n \) and \( k_i \) is known 'forward estimation'

1: Compute \( \hat{\bar{\mathcal{P}}}(\gamma_i) \) from (104) and let \( \phi(k) = \phi(\gamma_i) \) on \( [k_i, k_i + 2n] \)
2: Estimate \( k_{i+1} \) and extend \( \hat{\bar{\mathcal{P}}}(\gamma_i) \) to \( (k_i + 2n, k_{i+1}) \)
   by forward updating in Algorithm 3

end while

while \( \delta_{i-1}(\chi) > 2n \) and \( k_{i-1} \) is known 'backward estimation'

1: Compute \( \hat{\bar{\mathcal{P}}}(\gamma_{i-1}) \) from (105) and let \( \phi(k) = \phi(\gamma_{i-1}) \) on \( [k_{i-1}, k_i - 2n] \)
3: Estimate \( k_{i-1} \) and extend \( \hat{\bar{\mathcal{P}}}(\gamma_{i-1}) \) to \( [k_{i-1}, k_i - 2n] \)
   by backward updating in Algorithm 3

end while

end for

if \( \delta_{i-1}(\chi) \geq 6n \)

'\( k_i \) is known' \( \Rightarrow \) 'forward estimation from \( k_i \)'
'\( k_i \) is not known' \( \Rightarrow \) 'backward estimation from \( k_i + 1 \)'

end if

if \( \delta_{0}(\chi) > 4n \)

'\( k_1 \) is known' \( \Rightarrow \) 'backward estimation from \( k_1 \)'
'\( k_1 \) is not known' \( \Rightarrow \) 'forward estimation from \( k' \)'

end if

Outputs: \( \phi(k) \) on segments satisfying \( \delta_i(\chi) \geq 2n + 1 \)

Algorithm 4. Basis transformation for the discrete states

Inputs: Markov parameters, \( \hat{\bar{\mathcal{P}}} \) (ordered), and \( \phi \)

1: Estimate \( \Pi_j = T_jT_j^{-1} \) for \( j = 2, \cdots, \sigma \) from (108)–(110) and (111)–(112) with \( j \) and 1 plugged in places of \( j_2 \) and \( j_1 \) by choosing suitable paths and inversions if needed
2: Transform \( \hat{\bar{\mathcal{P}}} \) to \( \bar{\mathcal{P}} \) using \( \Pi_j \) as the similarity transformations for \( j = 2, \cdots, \sigma \).

Outputs: \( \hat{\bar{\mathcal{P}}} \) and \( \bar{\mathcal{P}}, j = 2, \cdots, \sigma \).