Algebraic Identification of MIMO SARX Models

Laurent Bako, René Vidal

Algebraic Identification of MIMO SARX Models

Laurent Bako[1,2] and René Vidal[2]

[1] Ecole des Mines de Douai, Département Informatique et Automatique, 59508, Douai, France
[2] Center for Imaging Science, Johns Hopkins University, Baltimore, MD 21218, USA

Abstract. We consider the problem of identifying the parameters of a multiple-input multiple-output switched ARX model with unknown number of submodels of unknown and possibly different orders. This is a very challenging problem because of the strong coupling between the unknown discrete state and the unknown model parameters. We address this challenge by algebraically eliminating the discrete state from the switched system equations. This algebraic procedure leads to a set of hybrid decoupling polynomials on the input-output data, whose coefficients can be identified using linear techniques. The parameters of each subsystem can then be identified from the derivatives of these polynomials. This exact analytical solution, however, comes with an important price in complexity: The number of coefficients to be identified grows exponentially with the number of outputs and the number of submodels. We address this issue with an alternative scheme in which the input-output data is first projected onto a low-dimensional linear subspace. The projected data is then fit with a single hybrid decoupling polynomial, from which the classification of the data according to the generating submodels can be obtained. The parameters of each submodel are then identified from the input-output data associated with each submodel.

1 Introduction

Hybrid systems are mathematical models of physical processes which exhibit both continuous and discrete behaviors. Such systems can be thought of as a collection of dynamical submodels with interacting behavior resulting from switching among all the submodels. The switches can be exogenous, deterministic, state-driven, event-driven, time-driven or totally random. Given input-output data generated by such a system, the identification problem consists of determining the parameters of each dynamical submodel as well as those of the switching mechanism (if any).

Prior work. Most of the existing hybrid system identification methods have been developed for the class of piecewise auto-regressive exogenous (PWARX) systems [1–6], for which the regressor space is partitioned into polyhedral regions with one ARX submodel associated with each polyhedron. For a comprehensive review of hybrid system identification techniques, we refer the readers to the survey paper [7]. The optimization based method [1] solves the identification problem as a linear or quadratic mixed integer programming problem. The clustering based procedures [2–4] use clustering to separate the data into different groups, linear regression to find the boundaries of the polyhedral regions, and linear identification to determine a submodel for each region. Other methods alternate between assigning the data to submodels and estimating simultaneously their parameters by performing a weights learning technique on a fuzzy parameterized model [8], solving a Minimum Partition into Feasible Subsystems (MinPFS) problem [6] or resorting to Bayesian inference [5]. The algebraic approach [9, 10] is applicable to the class of Switched ARX (SARX) models, where the switching mechanism can be arbitrary. This approach uses a single decoupling polynomial that vanishes on all the data regardless of their generating submodel. Once this polynomial is computed, the problem reduces to that of recovering the system parameters from the derivatives of the polynomial evaluated at a subset of the regressors.

Unfortunately, most of the aforementioned identification methods can only deal with single-input single-output (SISO) systems. While a few identification methods for multiple-input multiple-output (MIMO) switched linear [11–13] and piecewise affine [14–16] systems in state-space form do exist, they generally require the restrictive assumption of a minimum dwell time in each discrete state. In addition, they often iterate between data clustering and model estimation, which is quite sensitive to initialization.

Paper contributions. We present an algebraic solution to the problem of identifying MIMO SARX models. The orders of the submodels are unknown and possibly different and the number of submodels is not available. Our method is based on a technique called Generalized Principal Component Analysis (GPCA) [17], which can cluster data into multiple subspaces by polynomial fitting and differentiation. In contrast to the identification of SISO SARX models [10], where only one vanishing polynomial is used to embed the data lying in a mixture of hyperplanes, the identification of MIMO SARX models involves a potentially unknown number $n_h \geq 1$ of independent homogeneous polynomials that vanish on subspaces of co-dimension higher than one. In order to conveniently construct the regressors to which the embedding is applied, we first estimate the orders of the submodels and the number of discrete states from a rank constraint on the input-output data. Then, given the number of submodels, we compute the number of vanishing polynomials n_h and subsequently identify the ARX parameters from the derivatives of these polynomials. However, the number of coefficients to be estimated grows exponentially with the number of outputs and the number of submodels, thereby making the method computationally expensive. We thus propose an alternative method that first partitions the data according to each submodel using a single vanishing polynomial. Given the classification of the data according to each submodel, the parameters of each submodel are then identified using linear techniques.

2 Problem Statement

We consider a MIMO SARX model of the form

$$y(t) = \sum_{i=1}^{n_{\lambda_t}} A_{\lambda_t}^i y(t-i) + \sum_{i=0}^{n_{\lambda_t}} B_{\lambda_t}^i u(t-i) + e(t), \tag{1}$$

where $y(t) \in \mathbb{R}^{n_y}$ is the output vector, $u(t) \in \mathbb{R}^{n_u}$ is the input vector, $\lambda_t \in \{1, \ldots, s\}$ is the discrete state, n_{λ_t} is the order of the j-th submodel for $\lambda_t = j$, s is the number of submodels of the SARX system and $\left\{A_j^i\right\}_{j=1,\cdots,s}^{i=1,\cdots,n_j} \in \mathbb{R}^{n_y \times n_y}$ and $\left\{B_j^i\right\}_{j=1,\cdots,s}^{i=1,\cdots,n_j} \in \mathbb{R}^{n_y \times n_u}$ are the associated parameter matrices. The modeling error or process noise is represented by $e(t) \in \mathbb{R}^{n_y}$. In this representation, there may exist for certain models j an integer $\delta_j < n_j$ such that $B_j^i = 0$ for $i > \delta_j$ but we require that $A_j^{n_j} \neq 0$ for all j.

Given input-output data $\{u(t), y(t)\}_{t=1}^{N}$ generated by an SARX system of the form (1), and upper bounds on the system orders $\bar{n} \geq \max(n_j)$ and on the number of submodels $\bar{s} \geq s$, the identification problem can be formulated as follows: identify the number of submodels s, their orders $\{n_j\}_{j=1}^{s}$ and their parameters $\left\{A_j^i, B_j^i\right\}_{j=1,\cdots,s}^{i=1,\cdots,n_j}$.

3 Algebraic Identification of MIMO Switched ARX Systems

To begin with the identification procedure, let us define the parameter matrices

$$\begin{aligned} \Gamma_j &= \left[B_j^{n_j}\ A_j^{n_j}\ \cdots\ B_j^1\ A_j^1\ B_j^0\ A_j^0\right] \in \mathbb{R}^{n_y \times (n_j+1)(n_u+n_y)}, \\ P_j &= \left[0_{n_y \times q_j}\ \Gamma_j\right] \in \mathbb{R}^{n_y \times K},\ j = 1, \cdots, s, \end{aligned} \tag{2}$$

and the regressor vector

$$x_n(t) = \left[u(t-n)^\top\ y(t-n)^\top\ \cdots\ u(t-1)^\top\ y(t-1)^\top\ u(t)^\top\ -y(t)^\top\right]^\top \in \mathbb{R}^K, \tag{3}$$

with $n = \max_j(n_j)$, $A_j^0 = I_{n_y}$, $q_j = (n - n_j)(n_u + n_y)$ and $K = (n+1)\,(n_u + n_y)$.

For now, assume that the data is not corrupted by noise i.e. $e(t) = 0$ in (1). Then, the equations defining an SARX system of the form (1) may be re-written as

$$\left(P_1 x_n(t) = 0\right) \vee \cdots \vee \left(P_s x_n(t) = 0\right), \tag{4}$$

where $\vee$ refers to the *logical or* operator. To eliminate the discrete state from this set of sn_y equations, similarly to the case of SISO SARX models [9], we take the product of one equation per submodel. The advantage of doing so is that we obtain a set of polynomial constraints $\prod_{j=1}^{s}\left(\theta_{i_j}^\top x_n(t)\right) = 0$, with $\theta_{i_j}^\top = P_j(i,:)$ for $i = 1, \ldots, n_y$ and $j = 1, \ldots, s$, that are satisfied by all the data regardless of their generating submodel. Consequently, the equations in (4) are equivalent to a set of up to n_y^s (not necessarily independent) homogeneous polynomials $p_{i_1,\cdots,i_s}$ on $x_n(t)$ of the form

$$p_{i_1,\cdots,i_s}(z) = \prod_{j=1}^{s}\left(\theta_{i_j}^\top z\right) = \sum h_{i_1,\cdots,i_s}^{n_1,\cdots,n_K} z_1^{n_1} \cdots z_K^{n_K} = h_{i_1,\cdots,i_s}^\top \nu_s(z). \tag{5}$$

Here, $\nu_s : \mathbb{R}^K \to \mathbb{R}^{M_s(K)}$, with $M_s(K) = \binom{K+s-1}{s}$, is the Veronese map which associates to $z \in \mathbb{R}^K$ the vector of all monomials of degree s, $z_1^{s_1} \cdots z_K^{s_K}$, $s_1 + \cdots + s_K = s$, organized in a descending lexicographic order. Therefore, each $p_{i_1,\cdots,i_s}$ is a homogeneous polynomial of degree s with coefficient vector $h_{i_1,\cdots,i_s} \in \mathbb{R}^{M_s(K)}$ and all monomials of degree s in K variables stacked as a vector in $\nu_s(z) \in \mathbb{R}^{M_s(K)}$.

3.1 Known Number of Submodels of Known and Equal Orders

In this subsection, we assume that the number of submodels s is known, and that the orders of all the submodels are also known and equal to n. Note that the regressor vectors $x_n(t)$ generated by the hybrid model (1) lie in the union of the s subspaces $\{\mathrm{null}(P_j)\}_{j=1}^{s}$. A basis for each one of these subspaces can be estimated using the GPCA algorithm [17] as follows. From the entire set $\{u(t), y(t)\}_{t=1}^{N}$ of input-output data available, construct the matrix of embedded regressor vectors

$$L(n,s) = \left[\nu_s\big(x_n(n+1)\big)\ \cdots\ \nu_s\big(x_n(N)\big)\right]^\top \in \mathbb{R}^{(N-n) \times M_s(K)}. \tag{6}$$

Then the coefficient vectors $h_{i_1,\cdots,i_s}$ of the vanishing polynomials must satisfy

$$L(n,s)h_{i_1,\cdots,i_s} = 0. \tag{7}$$

In order to solve for the parameters $h_{i_1,\cdots,i_s}$ from (7), one needs to compute the null space of the embedded data matrix $L(n,s)$. Note that $h_{i_1,\cdots,i_s}$ is the symmetric part of the tensor product of an indexed set of rows $\{\theta_{i_j}\}_{j=1}^{s}$ taken from $\{P_j\}_{j=1}^{s}$, i.e. $h_{i_1,\cdots,i_s} = \operatorname{Sym}(\theta_{i_1} \otimes \cdots \otimes \theta_{i_s}) \in \mathbb{R}^{M_s(K)}$, where $\otimes$ denotes the Kronecker product. The linear span of all these coefficient vectors gives a subspace of $\mathbb{R}^{M_s(K)}$ that we will refer to as the space of homogeneous polynomials of degree s vanishing on the data. By computing the null space of $L(n,s)$, we obtain a basis for this subspace. In what follows, we will denote such a basis of dimension n_h as $H = \begin{bmatrix} h_1 & \cdots & h_{n_h} \end{bmatrix}$. Notice that the elements of this basis need not have the structure of a symmetric tensor product.

When the data are perfect and rich enough so that the dimension of the null space of $L(n,s)$ is exactly equal to n_h, the matrix of polynomial coefficients H can be computed as a basis for $\operatorname{null}(L(n,s))$ using the Singular Value Decomposition (SVD) of $L(n,s)$. A basis for $\operatorname{span}(P_{\lambda_t}^\top)$ can then be computed by differentiating the polynomials defined by H at $x_n(t)$. The parameter matrix P_{λ_t} of the submodel generating $x_n(t)$ can then be computed as the basis of $\operatorname{span}(P_{\lambda_t}^\top)$ with an identity matrix at the end, as defined in (2). As we do not need to compute the parameter matrices at each time instant, we can alternatively choose s regressors $z_j \in \operatorname{null}(P_j)$ (see §4.1) and obtain the s parameter matrices $\{P_j\}_{j=1}^{s}$ from the derivatives of the vanishing polynomials at $\{z_j\}_{j=1}^{s}$. Algorithm 1 gives a basic version of the GPCA algorithm [17] for computing the system parameter matrices $\{P_j\}_{j=1}^{s}$ from input-output data in a deterministic framework.

In practice the input-output data may be affected by noise. In this case, even with the assumption that the orders and the number of submodels are known, the matrix $L(n,s)$ is likely to be full rank and so, one may not be able to get the right basis H of polynomials. Therefore, it is desirable to know in advance the dimension n_h of this basis. In this way, H could be approximated by the right singular vectors of $L(n,s)$ that correspond to its n_h smallest singular values. But since the matrices P_j are not known, it is not easy to compute n_h in a general framework. However, under certain assumptions on the intersection between the null spaces of the matrices P_j, we can derive a closed form formula for n_h as outlined in Proposition 1.

Algorithm 1 (Identification of MIMO SARX systems using the GPCA algorithm)

Step 1: Compute a basis H for the null space of $L(n,s)$ by SVD and let the corresponding basis of vanishing polynomials of degree s be $\mathcal{Q}(z) = \begin{bmatrix} p_1(z) & \cdots & p_{n_h}(z) \end{bmatrix} = \nu_s(z)^\top H$.

Step 2: Let
$$\nabla\mathcal{Q}(z) = \begin{bmatrix} \frac{\partial p_1(z)}{\partial z} & \cdots & \frac{\partial p_{n_h}(z)}{\partial z} \end{bmatrix} = \left(\frac{\partial \nu_s(z)}{\partial z}\right)^\top H.$$

Step 3: Obtain by SVD a basis $T_j \in \mathbb{R}^{K \times n_y}$ for $\operatorname{span}(P_j^\top)$ as the range space of $\nabla\mathcal{Q}(z_j)$, $j = 1, \cdots, s$, where $z_j \in \operatorname{null}(P_j)$ but is not in $\operatorname{null}(P_i)$, for all $i \neq j$.

Step 4: Let $T_j^\top = \begin{bmatrix} T_j^1 & T_j^2 \end{bmatrix}$ be a partition of $T_j^\top$ such that $T_j^2 \in \mathbb{R}^{n_y \times n_y}$. T_j^2 is necessarily invertible and we can get $P_j = \left(T_j^2\right)^{-1} T_j^\top \in \mathbb{R}^{n_y \times K}$, $j = 1, \cdots, s$.

Proposition 1 *Let H be the symmetric tensor product of a set of matrices $B_1, \ldots, B_s$ in $\mathbb{R}^{K\times m}$. That is, H is the matrix whose columns are all vectors in $\mathbb{R}^{M_s(K)}$ of the form $\mathrm{Sym}\,(b_{i_1} \otimes \cdots \otimes b_{i_s})$, where $b_{i_1}, \ldots, b_{i_s}$ are, respectively, columns of $B_1, \cdots, B_s$. If $\sum_{i=1}^{s} \mathrm{rank}(B_i) - s < K$ and for all $\{i_1, \cdots, i_q\} \subset \{1, \cdots, s\}$, $q \leq s$,*

$$\mathrm{rank}(\left[B_{i_1}, \cdots, B_{i_q}\right]) = \min\Big(K, \sum_{j=1}^{q} \mathrm{rank}(B_{i_j})\Big), \tag{8}$$

then $\mathrm{rank}(H) = \prod_{j=1}^{s} \mathrm{rank}(B_j)$.

Assumption (8) of Proposition 1 corresponds to an important property of the subspace arrangement $\cup_{j=1}^{s} \mathrm{null}(B_j^\top)$ that is known as *transversality*. This property states that the dimension of the intersection of any subset of subspaces in the arrangement $\cup_{j=1}^{s} \mathrm{null}(B_j^\top)$ is as small as possible [18]. Under this assumption, the number of independent homogeneous polynomials that vanish on $\cup_{j=1}^{s} \mathrm{null}(B_j^\top)$ is equal to $\mathrm{rank}(H)$. If the same property holds for $\cup_{j=1}^{s} \mathrm{null}(P_j)$ and if $(n+1)\,(n_u + n_y) > (s-1)\,n_y$, then it follows from Proposition 1 that n_h is given by $n_h = \prod_{j=1}^{s} \mathrm{rank}(P_j) = n_y^s$ since $\mathrm{rank}(P_j) = n_y$ for all j. Although our formula is less general than the one derived in [19], it is much easier to compute. In the rest of the section, we will assume that the conditions of Proposition 1 hold, unless stated otherwise.

To summarize, given n and s, the parameter matrices P_j follow directly from Algorithm 1. If noise is present in the data, the same algorithm still applies but with the difference that the basis H is approximated by the singular vectors of $L(n,s)$ that are associated with its $n_h = n_y^s$ smallest singular values.

3.2 Unknown Number of Submodels of Unknown and Possibly Different Orders

Consider now the more challenging case where neither the orders nor the number of submodels are known and the orders are possibly different. Consequently, n_h is also unknown. This means that we need to derive all the parameters of the SARX model (1) directly from the data. In order to properly estimate these parameters, we shall first identify the orders and the number of submodels. Once this task is accomplished, Algorithm 1 can be applied to a certain submatrix of $L(n,s)$ that will be defined later.

Before proceeding further, we need to introduce some notations. For r and l, positive integers, we use the same definitions for $x_r(t)$ and $L(r,l)$ as before. Without loss of generality, we denote by $n = n_1 \geq n_2 \geq \cdots \geq n_s$ the orders of the different submodels that constitute the SARX system and let $\rho = \left[n_1 \cdots n_s\right] \in \mathbb{N}^s$ be a vector consisting of all the orders enumerated in a non-increasing order. It follows from (2) and (3) that the equations defining the SARX model (1) may be re-written as

$$(\mathit{\Gamma}_1 x_{n_1}(t) = 0) \vee \cdots \vee (\mathit{\Gamma}_s x_{n_s}(t) = 0)\,, \tag{9}$$

where $x_{n_j}(t) \in \mathbb{R}^{K_j}$, $K_j = (n_j+1)(n_u+n_y)$ and $\mathit{\Gamma}_j \in \mathbb{R}^{n_y \times K_j}$ for $j = 1, \ldots, s$. As before, we may eliminate $\vee$ in (9) by taking the product of one equation per submodel. This leads to a set of polynomial equations on the input-output data of the form

$$\big(\theta_1^\top x_{n_1}(t)\big) \cdots \big(\theta_s^\top x_{n_s}(t)\big) = h^\top \eta_\rho\big(x_n(t)\big), \tag{10}$$

where $\theta_j^\top \in \mathbb{R}^{1\times K_j}$ is a row of $\mathit{\Gamma}_j$, for $j = 1, \ldots, s$, and $\eta_\rho\big(x_n(t)\big)$ is a vector obtained from $\nu_s\,(x_n(t))$ after removing some of the monomials. $\eta_\rho\big(x_n(t)\big)$ does not contain all the monomials, because $n_j \leq n$ for all $j = 1, \ldots, s$, hence $x_{n_j}(t)$ is a sub-vector of $x_n(t)$, and so the product in (10) does not give rise to all the monomials in $\nu_s\left(x_n(t)\right)$.

In order to define the set of monomials that are to be removed, let $z = x_n(t)$ and consider a monomial $z_1^{\alpha_1} \cdots z_K^{\alpha_K}$, $\alpha_1 + \cdots + \alpha_K = s$. From the definition of $x_n(t)$ in (3), it can be seen that the element $z_j^{\alpha_j}$ is contained in a monomial of $\eta_\rho\big(x_n(t)\big)$ if the number of regressors $x_{n_i}(t)$ with length $K_i \geq K_1 - j + 1$ (that is the number of regressors where z_j shows up) is greater or equal to α_j. Therefore, in order for the whole monomial $z_1^{\alpha_1} \cdots z_K^{\alpha_K}$ to be included in $\eta_\rho\big(x_n(t)\big)$, we must have that $k_j \geq \alpha_j$ for all $j = 1, \cdots, K$, where $k_j = \text{card}\left(\{i : K_i \geq K_1 - j + 1\}\right)$. In view of this analysis, it can shown that the set of monomials to be removed can be indexed by the set $\mathscr{I}_\rho$ of exponents $(\alpha_1, \cdots, \alpha_K)$ satisfying $\alpha_1 + \cdots + \alpha_j > k_j$ for $j \leq K_1 - K_s$.

With this notation, we define a new embedded data matrix in $\mathbb{R}^{(N-n)\times(M_s(K_1)-|\mathscr{I}_\rho|)}$

$$V_\rho := \left[\eta_\rho\big(x_n(n+1)\big), \cdots, \eta_\rho\big(x_n(N)\big)\right]^\top \tag{11}$$

that is simply the matrix $L(n, s)$ with $|\mathscr{I}_\rho|$ missing columns ($n = \rho(1)$). As before, the null space of V_ρ contains the coefficients of the set of vanishing polynomials. However, we may not compute such coefficients directly, because we neither know the system orders ρ nor the number of models s. As it turns out, both ρ and s can be computed from the data under the assumption that the data are rich enough. More specifically:

Definition 1. *We say that the data* $\{u(t), y(t)\}_{t=1}^N$ *are sufficiently exciting for the SARX system (1) if the null space of* V_ρ *in* (11) *is of dimension exactly equal to* n_h*, i.e.*

$$\text{rank}(V_\rho) = M_s(K_1) - n_h - |\mathscr{I}_\rho|\,. \tag{12}$$

Notice that Definition 1 assumes implicitly that all the discrete states have been sufficiently visited. If we denote the matrix of data vectors related to the discrete state j by $\bar{X}_j = \left[x_n(t_1^j) \cdots x_n(t_{N_j}^j)\right]$, where the t_k^j, $k = 1, \ldots, N_j$, are the time instants t such that $\lambda_t = j$, then $\bar{X}_j$ must span completely $\text{null}(P_j)$. Otherwise, $\text{null}(P_j)$ may not be identifiable from $\cup_{j=1}^s \text{null}(P_j)$. We have the following result.

Theorem 1 *Let* $\bar{s} \geq s$ *be an upper bound on the number of submodels and let* r *be an integer. Assume that the data are sufficiently exciting in the sense of Definition 1. Assume further that* $N_j \gg M_{\bar{s}}(K_1)$ *for all* $j = 1, \ldots, s$*. Then* $\dim\left(\text{null}(L(r, \bar{s}))\right) = 0$ *if and only if* $r < \max(n_j)$*.*

Proof. Assume $r < n_1$ and let q be the number of submodels whose orders are less than or equal to r. Let $\mathcal{X} = \left[x_r(t_1^o), \cdots, x_r(t_{N_o}^o)\right] \in \mathbb{R}^{f\times N_o}$, with $f = (r+1)(n_u+n_y)$, be a matrix whose columns are regressor vectors formed by data generated by the $(s-q)$ submodels of orders $n_j > r$. Since the data are sufficiently exciting, $\mathcal{X}$ must be full row rank. It follows from Lemma 5 in [20] that $\text{rank}(\nu_{\bar{s}}(\mathcal{X})) = \min(N_o, M_{\bar{s}}(f)) = M_{\bar{s}}(f)$, where $\nu_{\bar{s}}\left(\mathcal{X}\right) = \left[\nu_{\bar{s}}(x_r(t_1^o)), \cdots, \nu_{\bar{s}}\left(x_r(t_{N_o}^o)\right)\right]$. Consequently, $L(r, \bar{s})$ is full column rank, because it is equal to a row permutation of $\left[\nu_{\bar{s}}(\mathcal{X}),\, \nu_{\bar{s}}(\mathcal{X}_{s-q+1}),\, \cdots,\, \nu_{\bar{s}}(\mathcal{X}_s)\right]^\top$.

Assume now that $r \geq \max(n_j)$. Then the row nullity of each data matrix X_j is at least one. This means that, for all $j = 1, \ldots, s$, there exists a nonzero $b_j \in \mathbb{R}^f$ satisfying $b_j^\top \mathcal{X}_j = 0$. One can then verify that $\text{Sym}(b_1 \otimes \cdots \otimes b_s \otimes a_{s+1} \otimes \cdots \otimes a_{\bar{s}}) \in \text{null}(L(r, \bar{s}))$ for some $a_i \in \mathbb{R}^f$. Hence, $\dim(\text{null}(L(r, \bar{s}))) \geq 1$. $\square$

Let $\bar{s} \geq s$ and $\bar{n} \geq \max(n_j)$ be upper bounds on the number of submodels and their orders respectively. Thanks to Theorem 1, we can estimate both the number of submodels s and the orders $\{n_j\}$ from the rank of the embedded data matrix $L(r, \bar{s})$. The basic idea is that, whenever r is less than one of the orders, there is no polynomial of degree $\bar{s} \geq s$ vanishing on the entire data set, provided that $N \gg s$ and that the data is sufficiently exciting. Therefore, as shown in Algorithm 2, we can obtain the first order n_1 by setting $\bar{\rho} = \begin{bmatrix} r \cdots r \end{bmatrix} \in \mathbb{N}^{\bar{s}}$, so that $V_{\bar{\rho}} = L(r, \bar{s})$, and then start decreasing r from $r = \bar{n}$ to $r = 0$ until $\mathrm{null}(V_{\bar{\rho}}) = \{0\}$ for some r^*. We then have $n_1 = r^* + 1$. Given n_1, we can set $\bar{\rho} = \begin{bmatrix} n_1 \; r \cdots r \end{bmatrix} \in \mathbb{N}^{\bar{s}}$ and repeat the procedure starting from $r = n_1$ and so on, until all the orders of all the s submodels are identified. Notice that, once all the orders of the s submodels have been correctly estimated, r will go to zero for the $\bar{s} - s$ remaining presumed submodels. Therefore, if one assumes that $n_j > 0$ for all $j = 1, \ldots, s$, then the number of submodels can be estimated as the number of orders n_j strictly greater than zero.

One advantage of Algorithm 2 is that it does not require prior knowledge of the dimension n_h of the space of vanishing polynomials. If all the orders are correctly identified, then the sufficiency of excitation condition in Definition 1 guarantees that the dimension of the null space of V_ρ is exactly equal to n_h. Given n_h, we can use Algorithm 1 to compute a basis H_ρ of $\mathrm{null}(V_\rho)$. We can then complete that basis with zeros to form a matrix $H \in \mathbb{R}^{M_s(K_1) \times n_h}$ such that the rows indexed by $\mathscr{I}_\rho$ are null. The remaining steps of Algorithm 1 are then performed without additional change.

Algorithm 2 (Identification of the orders and the number of submodels)

Set $j_o \leftarrow 1$, $n_j \leftarrow \bar{n}$ for $j = 1, \ldots, \bar{s}$,
$K \leftarrow (\bar{n} + 1)\,(n_u + n_y)$, $V \leftarrow L(\bar{n}, \bar{s})$,

1. Determine the maximum order n_1 using Theorem 1
   - **While** $\mathrm{rank}(V) < M_{\bar{s}}(K)$, do
     - $n_j \leftarrow n_1 - 1$ for $j = 1, \ldots, \bar{s}$
     - $K \leftarrow (n_1 + 1)\,(n_u + n_y)$
     - $V \leftarrow$ last $M_{\bar{s}}(K)$ columns of V
   - **EndWhile**
   - Obtain the maximum order as $n_1 \leftarrow n_1 + 1$ and then set $n_j \leftarrow n_1$ for $j = 1, \ldots, \bar{s}$
   - Set $V \leftarrow L(n_1, \bar{s})$ and $K \leftarrow (n_1 + 1)\,(n_u + n_y)$
2. Find the remaining orders n_j, $j = 2, \ldots, \bar{s}$ using Theorem 1
   - $j_o \leftarrow j_o + 1$
   - **While** $\mathrm{rank}(V) < M_{\bar{s}}(K) - |\mathscr{I}_{\bar{\rho}}|$
     - $n_j \leftarrow n_{j_o} - 1$ for $j = j_o, \ldots, \bar{s}$
     - $\bar{\rho} \leftarrow \begin{bmatrix} n_1 \cdots n_{\bar{s}} \end{bmatrix}$
     - Compute $\mathscr{I}_{\bar{\rho}}$ and remove the corresponding columns of V
   - **EndWhile**
   - Obtain the order n_{j_o}: $n_j \leftarrow n_{j_o} + 1$ for $j = j_o, \ldots, \bar{s}$
   - Set $V \leftarrow L(n_1, \bar{s})$
3. Go to step 2 until $j_o = \bar{s}$ or until one gets $n_{j_o} = 0$
4. Determine the number of submodels $s = \mathrm{card}\,(\{j : n_j > 0\})$

3.3 Implementation of Algorithm 2 with Noisy Data

The algorithm proposed in the previous subsection will operate correctly in the absence of noise. When dealing with noisy data, however, the multiple rank tests required may cause Algorithm 2 to fail, because the involved matrices may always be full rank. In this subsection, we discuss some possible improvements of the algorithm in order to enhance its ability to deal with noisy data.

Recall first that the purpose of the rank test is to check whether or not the dimension of the null space of $V_{\bar{\rho}}$ is zero for a given vector of orders $\bar{\rho}$. Therefore, we do not need to know the rank of $V_{\bar{\rho}}$ exactly. We just need a measure of how likely it is that there exists a nonzero vector $h_{\bar{\rho}}$ satisfying $V_{\bar{\rho}} h_{\bar{\rho}} = 0$.

One possible way of approaching this problem is to inspect the smallest singular value of $V_{\bar{\rho}}$ for different vectors $\bar{\rho}$. For example, to compute n_1, let $\bar{\rho}_{1,l} = \begin{bmatrix} l \cdots l \end{bmatrix} \in \mathbb{N}^{1\times\bar{s}}$, $l = 0, \ldots, \bar{n}$, and define $W_{\bar{\rho}_{1,l}} \doteq \frac{1}{N-\bar{n}} V_{\bar{\rho}_{1,l}} V_{\bar{\rho}_{1,l}}^\top$ as the matrix obtained from $\frac{1}{N-\bar{n}} L(\bar{n},\bar{s})^\top L(\bar{n},\bar{s})$ by removing its columns and rows indexed by $\mathscr{I}_{\bar{\rho}_{1,l}}$. Denote by $\sigma_{1,l}$, the smallest eigenvalue of the matrix $W_{\bar{\rho}_{1,l}}$ for $l = 0, \cdots, \bar{n}$. According to Theorem 1, $W_{\bar{\rho}_{1,l}}$ has at least one nonzero vector in its null space for all $l \geq n_1$ and hence, $\sigma_{1,n_1} \approx \cdots \approx \sigma_{1,\bar{n}} \approx \varepsilon_{1,n_1} \doteq \frac{1}{\bar{n}-n_1} \left(\sigma_{1,n_1+1} + \cdots + \sigma_{1,\bar{n}}\right)$ and are small compared to $\sigma_{1,0}, \cdots, \sigma_{1,n_1-1}$. Therefore, to determine n_1, one needs to look for the smallest integer $l \in \{0, \cdots, \bar{n}\}$ for which $\sigma_{1,l} \approx \varepsilon_{1,l}$ in a certain sense.

Following this procedure, Algorithm 2 can be implemented in a more efficient way for determining the orders. With $\hat{n}_0 = \bar{n}$, and given a user-defined decision threshold ε_0, the following algorithm directly computes the orders starting from $j = 1$ through $j = \bar{s}$, by avoiding the rank tests required in Algorithm 2.

$$
\begin{aligned}
&\bar{\rho}_{j,l} = \begin{bmatrix} \hat{n}_1 \cdots \hat{n}_{j-1}\ l \cdots l \end{bmatrix}, \ l = 0, \cdots, \hat{n}_{j-1}, \\
&\sigma_{j,l} = \min \lambda \left(W_{\bar{\rho}_{j,l}}\right), \ l = 0, \cdots, \hat{n}_{j-1}, \\
&\varepsilon_{j,l} = \frac{1}{\hat{n}_{j-1} - l} \left(\sigma_{j,l+1} + \cdots + \sigma_{j,\hat{n}_{j-1}}\right), \ l = 0, \cdots, \hat{n}_{j-1}, \\
&\mathcal{S}_j = \left\{ l = 0, \cdots, \hat{n}_{j-1} : |\sigma_{j,l} - \varepsilon_{j,l}| < \varepsilon_o \right\}, \\
&\hat{n}_j = \begin{cases} \min \{ l : l \in \mathcal{S}_j \}, & \text{if} \quad \mathcal{S}_j \neq \emptyset \\ \hat{n}_{j-1} & \text{otherwise,} \end{cases} \\
&j \leftarrow j + 1,
\end{aligned}
$$

where $\lambda(W_{\bar{\rho}_{j,l}})$ is the set of all eigenvalues of the matrix $W_{\bar{\rho}_{j,l}}$. In the notation such as $\bar{\rho}_{j,l}$, the index j indicates which submodel's order is being estimated while l is a possible value of the order sought.

4 Complexity reduction using a projection approach

The algebraic algorithm proposed in the previous section becomes computationally prohibitive when the dimensions of the SARX system are large. This is because the regressor $x_n(t) \in \mathbb{R}^{K_1}$ constructed from all n_y outputs is large, and so it induces an exponential increase in $M_s(K_1)$, the dimension of the space of homogeneous polynomials space of degree s in K_1 variables. Moreover, the number n_h of polynomials to be estimated is unknown, even when the orders and the number of submodels are given, unless one makes certain assumptions.

In this section, instead of attempting to compute a potentially large and unknown number of polynomials, we propose a computationally simpler method to identify the model parameters. The idea is to transform the MIMO system into a multiple-input single-output (MISO) system, and hence use only one decoupling polynomial to partition the data according to the different ARX submodels. Once all the data are correctly partitioned, the SARX system identification problem reduces to a standard regression problem for each discrete state.

To that end, notice that, without loss of generality, system (1) can transformed into the MISO system[3]

$$y(t) = \sum_{i=1}^{n_{\lambda_t}} a^i_{\lambda_t} y(t-i) + \sum_{i=0}^{n_{\lambda_t}} F^i_{\lambda_t} u(t-i) + e(t), \tag{13}$$

where the $\left\{a^i_j\right\}_{i=1,\cdots,n_j}^{j=1,\cdots,s}$ are the coefficients of the polynomial $z^{n_j} - a^1_j z^{n_j-1} - \cdots - a^{n_j}_j$ that encodes the poles of the jth submodel as its roots.

Let $\gamma = \left[\gamma_1 \cdots \gamma_{n_y}\right]^\top$ be a vector of real nonzero numbers and let $y_o(t) = \gamma^\top y(t) \in \mathbb{R}$ be a weighted combination of all the system outputs. Then, (13) can be transformed into the following single output system

$$y_o(t) = \sum_{i=1}^{n_{\lambda_t}} a^i_{\lambda_t} y_o(t-i) + \sum_{i=0}^{n_{\lambda_t}} \gamma^\top F^i_{\lambda_t} u(t-i) + \gamma^\top e(t). \tag{14}$$

Remark 1 *To the purpose of separating the data according to their generating submodels, one may be tempted to consider a single output $y_j(t)$ from (13) instead of a combination of all the n_y outputs. The problem with proceeding in this way is that, after pole-zero cancellation, the MISO system with output $y_j(t)$ may be common to many different modes and so, we may not be able to differentiate between those modes. By choosing a random linear combination of the outputs, such degenerate situations can be avoided almost surely.*

By introducing the blended output $y_o(t)$, we obtain only one hybrid decoupling polynomial $g(z)$ that is easier to deal with. However, at the same time the parameters of different submodels are combined. This raises the question of whether this combination of outputs preserves the distinguishability of the different submodels that constitute the SARX system. In fact, depending on the weights vector γ, two submodels which were initially distinct may reduce to the same submodel in (14). To analyze this risk, let

$$F_j = \left[F^{n_j}_j \ \cdots \ F^1_j \ F^0_j\right] \in \mathbb{R}^{n_y \times (n_j+1)n_u} \quad \text{and} \quad a_j = \left[a^{n_j}_j \ \cdots \ a^1_j\right]^\top \in \mathbb{R}^{n_j}. \tag{15}$$

It follows from (14) that two different modes i and j become indistinguishable after the previous transformation by γ, if they have the same order ($n_i = n_j$), the same

[3] Note that the orders n_j in (13) may be larger than the ones in (1). By an abuse of notation, we will keep using the same notation for the orders.

dynamics ($a_i = a_j$) and $\left(F_i^\top - F_j^\top\right)\gamma = 0$, i.e. when γ lies in null$(F_i^\top - F_j^\top)$. If the F_j were known one could readily select a γ which does not satisfy this condition. But these matrices are precisely what we are looking for. The question is, without knowing the F_j, how can we choose γ in such a way that for any $i \neq j$, $\gamma \notin$ null$(F_i^\top - F_j^\top)$. In fact, it is not hard to show that when γ is drawn randomly, this condition is satisfied with probability one. Therefore, two submodels that are distinct in the original system (13) remain so after the transformation. However, the separability of the modes, which is a measure of how close the different submodels are, may be affected.

From (14), let us redefine the parameter vector $\bar{\theta}_j$ and the regressor $\bar{x}_n(t)$ as

$$\bar{\theta}_j = \begin{bmatrix} 0_{q_j}^\top & \gamma^\top F_j^{n_j} & a_j^{n_j} & \cdots & \gamma^\top F_j^1 & a_j^1 & \gamma^\top F_j^0 & 1 \end{bmatrix}^\top \in \mathbb{R}^K,\ j = 1, \cdots, s \tag{16}$$

$$\bar{x}_n(t) = \begin{bmatrix} u(t-n)^\top & y_o(t-n) & \cdots & u(t)^\top & -y_o(t) \end{bmatrix}^\top \in \mathbb{R}^K, \tag{17}$$

where $K = (n+1)(n_u+1)$. One can view the smallest singular value $\sigma_0(X(\gamma))$ of $X(\gamma) = \begin{bmatrix} \bar{x}_{\bar{n}}(\bar{n}+1) & \cdots & \bar{x}_{\bar{n}}(N) \end{bmatrix}$, as a certain measure of how likely the data can be fitted to one subspace of $\mathbb{R}^{\bar{K}}$. It is in fact intuitive that the more distinguishable the subspaces are, the larger $\sigma_0(X(\gamma))$ should be. Therefore, to preserve the separability of the modes, we suggest to choose γ for example as $\gamma^* = \arg\max_{\gamma:\|\gamma\|\leq 1} \frac{\sigma_0(X(\gamma))}{\sigma_{max}(X(\gamma))}$, where $\sigma_{max}(X(\gamma))$ is the largest singular value of $X(\gamma)$. Since this could be a hard optimization problem, an alternative is to choose several candidate γs in such a way that $\sigma_0(X(\gamma))$ is in a certain proportion of $\sigma_{max}(X(\gamma))$.

Once γ has been chosen, we can proceed with the identification procedure. As before, we eliminate the dependency of the system equation on the switches by considering the following decoupling polynomial which vanishes on the data independently of their generating submodel:

$$g\big(\bar{x}_n(t)\big) = \prod_{j=1}^{s} \left(\bar{\theta}_j^\top \bar{x}_n(t)\right) = h^\top \nu_s\big(\bar{x}_n(t)\big) = 0. \tag{18}$$

Solving (18) is a particular and simpler case ($n_y = 1$) of the case studied in section 3. The procedure for the determination of $\bar{\theta}_j$ is roughly the same:

1. Solve for the orders and number of submodels using Algorithm 2.
2. Obtain h_ρ as any nonzero element in $\mathrm{null}(V_\rho)$ (which is expected to be one dimensional when the data are sufficiently exciting), and
3. Complete h_ρ with zeros to form a $h \in \mathbb{R}^{M_s(K)}$ so that the entries of h defined by $\mathscr{I}_\rho$ are zero.

Given h, the parameters may be obtained from the derivative of g as shown in [9]:

$$\bar{\theta}_j = \frac{\nabla g(z_j)}{e_K^\top \nabla g(z_j)}, j = 1, \ldots, s, \tag{19}$$

where z_j is a point in $S_j \setminus \cup_{i\neq j}^{s} S_i$, $S_j = \left\{x \in \mathbb{R}^K : \bar{\theta}_j^\top x = 0\right\}$, e_K is a vector of length K with 1 in its last entry and 0 everywhere else.

4.1 Classification of the Data

The computation of $\bar{\theta}_j$ for each submodel, involves finding a point lying in S_j but not in any other S_i, $i \neq j = 1, \ldots, s$. We find a point in S_j as $z_j = \bar{x}_n(\tau_j)$, where

$$\tau_j = \arg\min_{t \in \mathcal{D}_j} \left| \frac{\nabla g\left(\bar{x}_n(t)\right)^\top \bar{x}_n(t)}{e_K^\top \nabla g\left(\bar{x}_n(t)\right)} \right|, \tag{20}$$

$\mathcal{D}_1 = \{t : \nabla g(\bar{x}_n(t)) \neq 0\}$ and $\mathcal{D}_j = \{t : \nabla g(\bar{x}_n(t)) \neq 0, \bar{\theta}_i^\top \bar{x}_n(t) \neq 0, i = 1, ..., j{-}1\}$, for $j > 1$. Then one can compute the parameters by (19) using $z_j = \bar{x}_n(\tau_j)$.

Recall that recovering the vectors $\left\{\bar{\theta}_j\right\}_{j=1}^s$ associated with the blended output $y_o(t)$ is only an intermediate step in achieving the goal of computing the parameters a_j and F_j that define each subsystem of the original system (13). Now, from the parameters $\bar{\theta}_j$ obtained, we can determine the discrete state of (14) which is the same as that of (13) and then, compute finally the system sought. In order to discard possible outliers in the data we set up a performance bound $\varepsilon < 1$ to define the following decision rules:

$$\text{If} \quad \min_j \Delta(\bar{\theta}_j, \bar{x}_n(t)) > \varepsilon \left\|\bar{x}_n(t)\right\|, \quad \text{then} \quad \lambda_t \quad \text{is undecidable.}$$

$$\text{If} \quad \min_j \Delta(\bar{\theta}_j, \bar{x}_n(t)) \leq \varepsilon \left\|\bar{x}_n(t)\right\|, \quad \text{then} \quad \lambda_t = \arg\min_j \Delta\left(\bar{\theta}_j, \bar{x}_n(t)\right).$$

Here $\Delta(\bar{\theta}_j, \bar{x}_n(t)) = \dfrac{\left|\bar{\theta}_j^\top \bar{x}_n(t)\right|}{\left\|\bar{\theta}_j\right\|}$ is the distance from the point $\bar{x}_n(t)$ to the linear hyperplane S_j defined by its normal vector $\bar{\theta}_j$. We define $\mathscr{X}_j = \{t > \bar{n} : \lambda_t = j\}$ $= \left\{t_1^j, \cdots, t_{N_j}^j\right\}$, $j = 1, \ldots, s$ as the set of time instances in which the regressors are generated by the submodel j.

4.2 Estimation of the Submodel Parameters

Based on the results of the previous classification, we know which data correspond to each generating mode. Therefore, we are left with determining the parameters of each mode j from the data indexed by $\mathscr{X}_j$. To begin with, consider a single linear submodel j of order n_j from (13). For any $t \in \mathscr{X}_j$, let us define

$$\Phi_j^y(t) := \left[y(t-1) \cdots \; y(t-n_j)\right] \in \mathbb{R}^{n_y \times n_j}, \tag{21}$$

$$\phi_j^u(t) := \left[u(t)^\top \cdots \; u(t-n_j)^\top\right]^\top \in \mathbb{R}^{(n_j+1)n_u}. \tag{22}$$

The parameters of the submodels of system (13) can be computed as the solution to the following linear regression problem

$$y(t) = \left[\,\Phi_j^y(t) \;\phi_j^u(t)^\top \otimes I_{n_y}\,\right] \begin{bmatrix} a_j \\ \text{vec}(F_j) \end{bmatrix} + e(t), \; t \in \mathscr{X}_j. \tag{23}$$

This equation is obtained by making use of the identity $\text{vec}(AXB) = (B^\top \otimes A)\text{vec}(X)$, where the symbol $\otimes$ refers to the Kronecker product and $\text{vec}(\cdot)$ is the vectorization operator. Notice that in the whole procedure, the vectors a_j, $j = 1, \ldots, s$, are estimated twice. The first estimate (obtained from $\bar{\theta}_j$) is considered as a raw estimate that is required here just to be able to discriminate among the different modes. The second estimate from (23) is expected to be more accurate.

5 Numerical results

We test the performance of the proposed approach on an SARX system composed of two submodels of orders 2 and 1, with $n_u = 1$ input and $n_y = 2$ outputs. The system equations are given by

$$y(t) = a_j^1 I_{n_y} y(t-1) + a_j^2 I_{n_y} y(t-2) + b_j^0 u(t) + b_j^1 u(t-1) + b_j^2 u(t-2), \quad (24)$$

where a_j^1 and a_j^2, $j = 1, 2$, are scalar coefficients and b_j^0, b_j^1, b_j^2 are vectors of dimension $n_y = 2$. The coefficients a_j^2 and b_j^2 are zero for the second submodel.

The system is driven by a zero-mean white Gaussian noise input with unit standard deviation and switches periodically from one discrete state to another every 10 samples. The output is corrupted with additive noise with a signal-to-noise ratio (SNR) of 30 dB.

The parameters of the two ARX models are given by the matrices

$$P_1 = \left[\begin{array}{cc|cc|c|c|c} 1.3561, & 0 & 0.6913, & 0, & 0 & 0.3793 & 0.2639 \\ 0, & 1.3561 & 0, & 0.6913, & 1.3001 & 1.8145 & 0.7768 \end{array}\right], \quad (25)$$

$$P_2 = \left[\begin{array}{cc|cc|c|c|c} 0.9485, & 0 & 0, & 0 & 1.7661 & 2.9830 & 0 \\ 0, & 0.9485 & 0, & 0 & 0 & 0.9106 & 0 \end{array}\right], \quad (26)$$

which are defined with respect to the regressor vector

$$\left[y(t-1)^\top \,\middle|\, y(t-2)^\top \,\middle|\, u(t) \,\middle|\, u(t-1) \,\middle|\, u(t-2) \right]^\top .$$

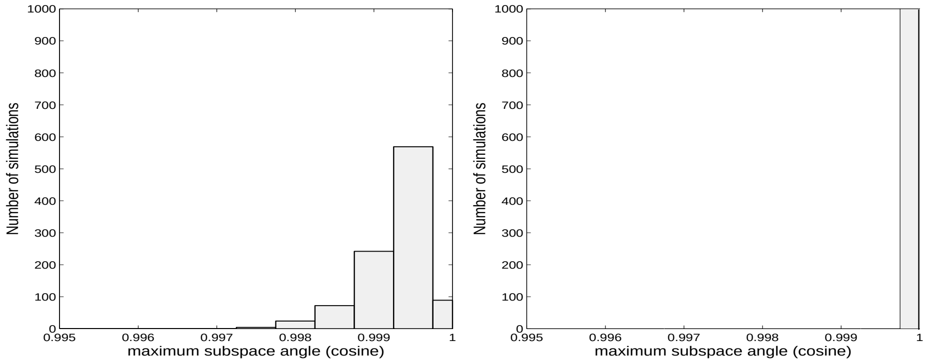
Given input-output data generated by this system on a time window of size 1500, we are interested in extracting the number of constituent submodels, the orders of these submodels and the parameters that describe them. To demonstrate the performance of our algorithm we carried out a Monte-Carlo simulation of size 1000 with the following user-defined set of parameters: $\bar{n} = 3$ and $\bar{s} = 3$. For a threshold of $\varepsilon_0 = 10^{-3}$ in the algorithm of §3.3, the estimation of the orders of both submodels is realized with 100% of successes. Since we provided $\bar{s} = 3$, the vector of orders is obtained as $\hat{\rho} = \begin{bmatrix} 2 & 1 & 0 \end{bmatrix}$. The means of the estimates $\hat{P}_1$ and $\hat{P}_2$ obtained across all the simulations are given by:

$$\hat{P}_1 = \left[\begin{array}{cc|cc|c|c|c} 1.3558, & 0.0043 & 0.6897, & 0.0036 & 0.0056 & 0.3937, & 0.2639 \\ -0.0012,, & 1.3558 & -0.0021, & 0.6907 & 1.3031 & 1.8208 & 0.7753 \end{array}\right], \quad (27)$$

$$\hat{P}_2 = \left[\begin{array}{cc|cc|c|c|c} 0.9480, & 0.0045 & -0.0005, & 0.0050 & 1.7710, & 2.9869, & 0.0050 \\ -0.0003, & 0.9479 & -0.0001, & -0.0006 & -0.0012 & 0.9081 & -0.0018 \end{array}\right]. \quad (28)$$

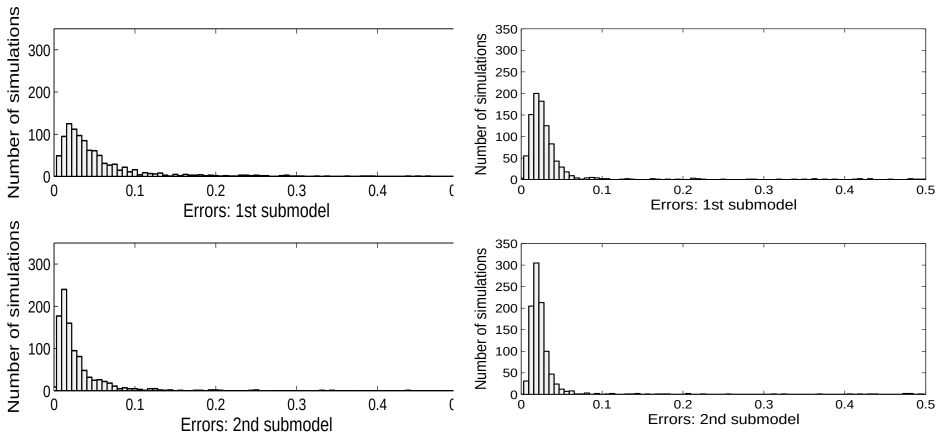
Figure 1 shows a histogram with the maximum angle between the column space of the hybrid parameter matrix H and that of its estimate $\hat{H}$. Notice that for all simulations the cosine of this angle is larger than 0.99, implying a strong correlation between H and its estimate. For the second identification method, the result is much better since H consists of only one vector.

Figure 2 shows the relative errors between the true parameter matrices P_j and the estimates $\hat{P}_j$ obtained by our algorithm. Observe that the percentage of simulations that give errors less than 0.05 is about 66% for the first submodel and about 85% for the second submodel. These percentages improve significantly (86% and 93%) when we use the classification approach described in Section 4.



(a) MIMO GPCA method (b) MISO GPCA method

Fig. 1: Histograms of the maximum subspace angle between $\mathrm{span}(H)$ and $\mathrm{span}(\hat{H})$.



(a) MIMO GPCA method (b) MISO GPCA method

Fig. 2: Histograms of the errors $\left\|P_1 - \hat{P}_1\right\|_2 / \left\|P_1\right\|_2$ and $\left\|P_2 - \hat{P}_2\right\|_2 / \left\|P_2\right\|_2$.

6 Conclusions

We have presented an algebraic approach to the identification of MIMO SARX models with unknown number of submodels of unknown and possibly different orders. The number of submodels and their orders are estimated from a rank constraint on the input-output data, and the model parameters using a subspace clustering technique called GPCA. As the complexity of the method is exponential on the number of outputs and submodels, we proposed a simpler approach that applies GPCA to a MISO system built by projecting the original data. Future work includes developing recursive identification algorithms for MIMO SARX systems, such as the one in [20] for SISO systems.

Acknowledgements. The authors thank Mr. Dheeraj Singaraju for his help in proofreading this paper. This work has been funded by BOURSE-MOBILITE from the Regional Council of Nord-Pas-de-Calais (France), by Johns Hopkins startup funds, and by grants NSF EHS-05-09101, NSF CAREER IIS-04-47739 and ONR N00014-05-1083.

References

1. Roll, J., Bemporad, A., Ljung, L.: Identification of piecewise affine systems via mixed-integer programming. Automatica **40**(1) (2004) 37–50
2. Ferrari-Trecate, G., Muselli, M., Liberati, D., Morari, M.: A clustering technique for the identification of piecewise affine systems. Automatica **39**(2) (2003) 205–217
3. Ferrari-Trecate, G., Muselli, M.: Single-linkage clustering for optimal classification in piecewise affine regression. In: IFAC Conference on the Analysis and Design of Hybrid Systems. (2003)
4. Nakada, H., Takaba, K., Katayama, T.: Identification of piecewise affine systems based on statistical clustering technique. Automatica **41**(5) (2005) 905–913
5. Juloski, A., Weiland, S., Heemels, M.: A Bayesian approach to identification of hybrid systems. IEEE Transactions on Automatic Control **50**(10) (2005) 1520–1533
6. Bemporad, A., Garulli, A., Paoletti, S., Vicino, A.: A bounded-error approach to piecewise affine system identification. IEEE Transactions on Automatic Control **50**(10) (2005) 1567–1580
7. Paoletti, S., Juloski, A., Ferrari-Trecate, G., Vidal, R.: Identification of hybrid systems: A tutorial. European Control Journal (2007)
8. Ragot, J., Mourot, G., Maquin, D.: Parameter estimation of switching piecewise linear systems. In: Conference on Decision and Control. (2003)
9. Vidal, R., Soatto, S., Ma, Y., Sastry, S.: An algebraic geometric approach to the identification of a class of linear hybrid systems. In: Conference on Decision and Control. (2003) 167–172
10. Ma, Y., Vidal, R.: Identification of deterministic switched ARX systems via identification of algebraic varieties. In: Hybrid Systems: Computation and Control. Springer Verlag (2005) 449–465
11. Bako, L.., Mercere, G., Lecoeuche, S.: Online subspace identification of switching systems with possibly varying orders. In: European Control Conference. (2007)
12. Huang, K., Wagner, A., Ma, Y.: Identification of hybrid linear time-invariant systems via subspace embedding and segmentation. In: Conference on Decision and Control. Volume 3. (2004) 3227–3234
13. Verdult, V., Verhaegen, M.: Subspace identification of piecewise linear systems. In: Proceedings of the 43rd IEEE Conference on Decision and Control. (2004) 3838–3843
14. Münz, E., Krebs, V.: Identification of hybrid systems using a priori knowledge. In: IFAC World Congress. (2002)
15. Verdult, V., Verhaegen, M.: Subspace identification of piecewise linear systems. In: IEEE Conference on Decision & Control. (2004) 3838–3843
16. Münz, E., Krebs, V.: Continuous optimization approaches to the identification of piecewise affine systems. In: IFAC World Congress. (2005)
17. Vidal, R., Ma, Y., Sastry, S.: Generalized Principal Component Analysis (GPCA). IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(12) (2005) 1–15
18. Ma, Y., Yang, A., Derksen, H., Fossum, R.: Estimation of subspace arrangements with applications in modeling and segmenting mixed data. SIAM Review (To appear) (2008)
19. Derksen, H.: Hilbert series of subspaces arrangements. Journal of Pure and Applied Algebra **209**(1) (2007) 91–98
20. Vidal, R.: Recursive identification of switched ARX systems. Automatica (To appear) (2008)