A randomized two-stage iterative method for switched nonlinear systems identification

Federico Bianchi[a,*], Maria Prandini[a], Luigi Piroddi[a]

[a]*Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milano (Italy)*
(e-mail: {federico.bianchi, maria.prandini, luigi.piroddi}@polimi.it).

Abstract

This paper addresses the identification of discrete time switched nonlinear systems, which are collections of discrete time nonlinear continuous systems (modes) indexed by a finite-valued variable defining the current mode. In particular, we consider the class of Switched Nonlinear AutoRegressive eXogenous (Switched NARX, or SNARX) models, where the continuous dynamics are represented by NARX models. Given a set of input-output data, the identification of a SNARX model for the underlying system involves the simultaneous identification of the mode sequence and of the NARX model associated to each mode, configuring a mixed integer non-convex optimization problem, hardly solvable in practice due to the large combinatorial complexity. In this paper, we propose a black-box iterative identification method, where each iteration is characterized by two stages. In the first stage the identification problem is addressed assuming that mode switchings can occur only at predefined time instants, while in the second one the candidate mode switching locations are refined. This strategy allows to significantly reduce the combinatorial complexity of the problem, thus allowing an efficient solution of the optimization problem. The combinatorial optimization is addressed using a randomized method, whereby the sample-mode map and the SNARX model structure are characterized by a probability distribution, which is progressively tuned via a sample-and-evaluate strategy, until convergence to a limit distribution concentrated on the best SNARX model of the system generating the observed data.

Keywords: hybrid systems, switched systems, model identification, randomized algorithms.

1. Introduction

Hybrid systems (HSs) are dynamical systems whose behavior can be described by the interaction of time- and event-driven dynamics. HSs provide a unified framework for the representation of technological systems where continuous models such as differential or difference equations describe the physical and mechanical part, and discrete models such as finite-state machines or Petri nets describe the software and logical behavior. Also many real physical processes exhibiting both fast and slow changing behaviors can be described by HS models. When first principles modeling is too complicated, then, the model has to be identified based on experimental data collected from the real system.

Most research regarding the identification of hybrid systems (HSI) has focused on switched affine (SA) and piecewise affine (PWA) models due to their universal approximation properties and their simple interpretation. Indeed, they provide the simplest extensions of continuous systems that can handle hybrid phenomena. In SA systems, the discrete state is an exogenous finite-valued input which determines the switching between different continuous affine dynamics, whereas in PWA systems the switching mechanism is determined by a polyhedral partition of the (continuous) state-input domain. The input-output counterparts of these system classes are Switched ARX (SARX) and PieceWise affine ARX (PWARX), respectively. The optimization problem induced by the identification task is of the mixed-integer type, since it involves the identification of discrete variables (representing the mapping of the samples to the modes and the model structure associated to each mode), as well as continuous ones (the parameters of the

*Corresponding author at: Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milano (Italy)

models describing the continuous dynamics associated to the various system conditions). Many approaches have been proposed over the last two decades for the case of *affine* dynamics (see, *e.g.*, [29], [11], and [16], for a comprehensive review). These methods can be roughly classified into two categories, depending on how the optimization problem is tackled. Some methods adopt a solution strategy which addresses the full problem, optimizing simultaneously over both continuous and discrete variables, [3], [31], [23], [26], [1], [28], [27], [24], while other methods deal separately with the mode and structure classification and the parameter estimation tasks, [9], [13], [30], [12].

Surprisingly fewer works have tackled the case of *nonlinear* continuous dynamics associated to the modes, in spite of its importance in modeling complex applications. Indeed, if no *a priori* information on the number of modes is available, one can in principle identify an arbitrarily high number of local linear models (and switchings among them) in order to achieve a good model accuracy. However, this prevents the identification of the real dynamics of the hybrid system and hinders its physical interpretation. It also greatly aggravates the combinatorial complexity of the optimization problem, due to the increasing number of switchings. An attempt to deal with nonlinear HSs is documented in [14], where a method based on kernel regression and Support Vector Machines (SVMs) is discussed. In this setting, the number of variables over which the optimization is carried out grows rapidly with the number of data N and the number of modes N_M, according to $2N_M(N+1)$, and hence this method can deal only with relatively small problems. A reformulation of the optimization problem in a continuous framework is studied in [17] and [18], thus allowing the use of efficient solvers and enabling the solution of larger problems. The efficiency of this method is further improved in [19] by introducing fixed-size kernel submodels. In [15], the authors proposed an extension of the sum-of-norms approach described in [27] to piecewise systems with nonlinear dynamics, based again on kernel functional expansions. The method employs a convex cost function containing an accuracy term (quantifying the quality of fit of each local model on the assigned data samples), a term penalizing the local model complexity, and a variational term which controls the overall complexity as a function of the number of local models. Note that, in case of time-ordered and consecutive data, the proposed approach is similar to that in [6] which addresses the segmentation of ordered data getting from nonlinear dynamical systems. In [2] the identification problem is first formulated as a sparse optimization problem and then relaxed in a convex form by approximating the ℓ_0 norm with the ℓ_1 norm. A sufficient condition guaranteeing the optimality of the relaxed convex problem solution was provided only under a noiseless assumption. The notion of robust sparsity is introduced in [20] to extend the applicability of the previous method to the noisy case. On the down side, the method requires the careful setting of several parameters (*e.g.*, the factor that defines the trade-off between model complexity and accuracy, or the weights used to improve the sparsity of the solution), which appears to be far from trivial.

It is worth noticing that most of the aforementioned approaches are nonparametric in that they are based on kernel functional expansions. Instead, in this paper the identification problem has been addressed from a parametric perspective using nonlinear models of the NARX/NARMAX class [21, 22], where the nonlinear functions are represented as finite-dimensional parameterized polynomial expansions. Indeed, this is a very popular approach in black-box nonlinear model identification [5], provided the identification procedure includes a model structure selection (MSS) process to tackle the curse of dimensionality issue that is inherent to polynomial expansions. Polynomial nonlinear models of this type have several attractive features (see *e.g.* [5] for a more detailed discussion), among which the ability to represent a wide range of nonlinear systems using a small number of parameters, the easy interpretability, and the amenability to nonlinear frequency analysis using generalized frequency response functions.

This paper introduces an iterative randomized approach for the segmentation of time-ordered data observed from Switched Nonlinear ARX (SNARX) models, which extends our previous work in [4] where a randomized identification algorithm was proposed based on the assumption that the time instants at which mode switchings may occur is *a priori* known (although it remains to ascertain which of these correspond to actual switchings and between which modes). Here, such a restrictive assumption is removed by adopting an iterative procedure which starting from an initial guess of the candidate set of mode switching instants progressively refines it, ultimately allowing an accurate estimation of the actual switchings.

The proposed method consists of a two-stage procedure repeated at each iteration, the first stage addressing the SNARX identification problem based on the current set of candidate switching times, and the second aiming at the refinement of such set. The restriction of the candidate switchings is crucial in reducing the combinatorial complexity of the optimization problem associated to the identification task performed in the first stage, thus allowing its solvability. More in detail, it induces a partition of the data into (a small number of) sub-periods, each of which is associated to a mode, and the NARX model associated to each mode can be identified based on all the data segments labeled

with it.

A randomized method is adopted to address the identification task of the first stage. Specifically, a probability distribution is defined over the space of possible SNARX models (that are compatible with the current set of candidate switching times), representing the likelihood of each model being the actual one. This distribution is progressively refined through a sample-and-evaluate strategy, until convergence is obtained to a limit distribution, representing a specific SNARX model. In the second stage, the number and location of the candidate switching times is refined, based on the evidence gathered in the first stage. The rationale of the refinement stage is to sample more densely the time horizon in the proximity of the estimated switching times and adopt a sparser sampling elsewhere.

The sequence of the SNARX model identification and refinement stages is repeated until convergence, ideally, to the SNARX model that best describes the available data (target model). The described iterative two-stage method requires that the number of modes is *a priori* known (a commonly adopted assumption in the literature), whereas the structure of the NARX models associated with the modes is not assumed to be known. Though suboptimal, the proposed method is experimentally shown to be quite effective, and capable of operating with noisy data and dealing with relatively large data-sets.

The rest of the paper is organized as follows. Section 2 describes the SNARX model identification problem and provides a general overview of the proposed two-stage procedure, which is then detailed in Sections 3 and 4. Finally, some simulation examples are presented in Section 5, followed by some concluding remarks.

2. Identification of SNARX models

2.1. SNARX models: structure and parametrization

A SNARX model is represented by a set of N_M NARX models indexed by a finite-valued variable defining the modes, a NARX model [22] being a general input-output representation of a nonlinear model described by the following equation

$$y(t) = g\left(\boldsymbol{x}(t); \boldsymbol{\vartheta}\right) + e(t),$$

where $\boldsymbol{x}(t) = [y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u)]$ is a finite-dimensional vector of the most recent past observations (n_y and n_u being the model orders), $e(t)$ is an additive white noise signal, and $g(\cdot)$ is an unknown nonlinear function parameterized via a vector $\boldsymbol{\vartheta} = [\vartheta_1, \ldots \vartheta_n]^T$ of coefficients. The corresponding predictor is given by:

$$\hat{y}(t) = g\left(\boldsymbol{x}(t); \boldsymbol{\vartheta}\right).$$

The nonlinear mapping $g(\cdot)$ can be expressed as a linear combination of (nonlinear) basis functions $\varphi_j(\boldsymbol{x}(t))$, $j = 1, \ldots, n$:

$$g\left(\boldsymbol{x}(t); \boldsymbol{\vartheta}\right) = \sum_{j=1}^{n} \vartheta_j \varphi_j(\boldsymbol{x}(t)), \tag{1}$$

so that the predictor can be reduced to the following linear regression:

$$\hat{y}(t) = \boldsymbol{\varphi}(\boldsymbol{x}(t))^T \boldsymbol{\vartheta},$$

where all basis functions are collected in the *regression vector* $\boldsymbol{\varphi}(\boldsymbol{x}(t)) = [\varphi_1(\boldsymbol{x}(t)), \ldots \varphi_n(\boldsymbol{x}(t))]^T$.

Remark 1 (polynomial NARX models). *Among all the possible representations of $g(\cdot)$, one of the most common is the polynomial functional expansion, whereby the regressors are monomials of elements in $\boldsymbol{x}(t)$ up to a given order n_d,* i.e. *they are of the form $x_1^{k_1} x_2^{k_2} \cdots x_l^{k_l}$, where $l = |\boldsymbol{x}| = n_u + n_y$, with $\sum_{i=1}^{l} k_i \leq n_d$ and $k_i \geq 0$. Unfortunately, the regressor set grows rapidly with n_u, n_y, and n_d, a problem known as the "curse of dimensionality". However, in practical applications it is seldom necessary to employ full polynomial expansions and it is typically observed that few terms suffice to obtain highly accurate and robust models. This justifies the attention that the MSS problem has deserved in the NARX model identification literature.*

The structure of a NARX model can be coded in a vector $\boldsymbol{s} \in \{1,2\}^n$, where $s_j = 1$ if the j-th regressor belongs to the model structure (and $s_j = 2$ otherwise). If $s_j = 2$ the corresponding parameter ϑ_j in (1) is set to zero. Accordingly, the overall SNARX model structure can be encoded in a $n \times N_M$ matrix $S = [\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(N_M)}] \in \mathcal{S} = \{1,2\}^{n \times N_M}$, which is the collection of the structures of the NARX models that are associated with its N_M modes.

Given a data-set of time-ordered and consecutive input-output samples of a SNARX, a finite-valued switching signal assigns each sample to a specific mode. For the purpose of the SNARX model identification, the SNARX model structure thus needs to be extended to include the mode switching signal $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_N] \in \Sigma = \{1, \ldots, N_M\}^N$, such that $\sigma_t = i$ if sample t is attributed to mode i. A SNARX model structure is thus expressed by a pair $\lambda = (\boldsymbol{\sigma}, S)$ taking values in $\Lambda = \Sigma \times \mathcal{S}$. Given a SNARX model with structure $\lambda \in \Lambda$, its parametrization $\boldsymbol{\vartheta}^{(i)}$, $i = 1, \ldots, N_M$, can be obtained by minimizing the mean square prediction error on the available data. The quality of a SNARX model with structure λ is thus given by the value of the loss function corresponding to its optimal parameterization:

$$\begin{aligned} \mathcal{L}(\lambda) = \min_{\{\boldsymbol{\vartheta}^{(i)}\}_{i=1}^{N_M}} & \frac{1}{N} \sum_{t=1}^{N} \sum_{i=1}^{N_M} \beta_t^{(i)} \cdot \varepsilon_i^2(t), \\ & \text{subject to } \vartheta_j^{(i)} = 0 \text{ if } s_j^{(i)} = 2, j = 1, \ldots, n, i = 1, \ldots, N_M, \end{aligned} \tag{2}$$

where $\varepsilon_i(t) = y(t) - \hat{y}(t) = y(t) - \boldsymbol{\varphi}(\boldsymbol{x}(t))^T \boldsymbol{\vartheta}^{(i)}$ is the prediction error associated to mode i, and $\beta_t^{(i)}$ is a binary variable encoding the sample-mode mapping provided by $\boldsymbol{\sigma}$:

$$\sigma_t = i \Longleftrightarrow \beta_t^{(i)} = 1. \tag{3}$$

If we denote as $N_i = \sum_{t=1}^{N} \beta_t^{(i)}$ the number of samples in the data-set that are associated with mode i, then $\mathcal{L}(\lambda)$ in (2) can be explicitly expressed in terms of the contribution of each mode as

$$\mathcal{L}(\lambda) = \frac{1}{N} \sum_{i:\, N_i \neq 0} N_i \cdot \mathcal{L}^{(i)}(\boldsymbol{\sigma}, \boldsymbol{s}^{(i)}),$$

where $\mathcal{L}^{(i)}(\boldsymbol{\sigma}, \boldsymbol{s}^{(i)})$ measures the accuracy of the model of the i-th mode, with structure $\boldsymbol{s}^{(i)}$, when the switching signal is $\boldsymbol{\sigma}$. Index $\mathcal{L}^{(i)}(\boldsymbol{\sigma}, \boldsymbol{s}^{(i)})$ is well-defined if $N_i \neq 0$ and is given by:

$$\begin{aligned} \mathcal{L}^{(i)}(\boldsymbol{\sigma}, \boldsymbol{s}^{(i)}) = \min_{\boldsymbol{\vartheta}^{(i)}} & \frac{1}{N_i} \sum_{t=1}^{N} \beta_t^{(i)} \cdot \varepsilon_i^2(t) \\ & \text{subject to } \vartheta_j^{(i)} = 0 \text{ if } s_j^{(i)} = 2, j = 1, \ldots, n. \end{aligned} \tag{4}$$

Remark 2 (Redundancy of the parametrization). *Note that when performing the minimization in* (4) *the parameters associated to redundant regressors are set to* 0*. Regressor redundancy can be tackled* e.g. *by introducing a regularization term in $\mathcal{L}^{(i)}$, or by applying an* a posteriori *t-test on the estimated parameter vector to detect terms that are statistically indistinguishable from* 0*. The latter approach is the one adopted in our implementation.*

In the sequel, instead of $\mathcal{L}$ we will employ the following performance index (conveniently ranged in $[0, 1]$) to characterize λ:

$$\mathcal{J}(\lambda) = e^{-K_\lambda \mathcal{L}(\lambda)}, \tag{5}$$

where $K_\lambda > 0$ is a scaling parameter. Exponential indices can facilitate the discrimination between models with similar performance by amplifying their difference [32], thus improving the structure selection process.

2.2. Identification of SNARX models: a two-stage approach

A SNARX model identification problem consists in estimating from a data-set of N time-ordered and consecutive input-output data pairs the model structures $\boldsymbol{s}^{(i)}$ and parameterizations $\boldsymbol{\vartheta}^{(i)}$, $i = 1, \ldots, N_M$, of the mode dynamics, as well as the switching signal σ_t, $t = 1, \ldots, N$. Notice that the identification of σ_t amounts to segmenting the data in consecutive portions, attributing each subperiod to the appropriate mode. Assuming that the number of modes N_M is known, the SNARX identification problem can be reformulated as that of finding the λ value that maximizes the

performance index $\mathcal{J}(\lambda)$ in (5) and does not have redundant terms. If there exists only one such λ, this can be written as

$$\lambda^\star = \left(\boldsymbol{\sigma}^\star, S^\star\right) = \arg\max_{\lambda\in\Lambda} \mathcal{J}(\lambda). \tag{6}$$

The parameters of the NARX model associated to mode i are the solutions of the following LS problems:

$$\boldsymbol{\vartheta}^{(i)\star} = \arg\min_{\boldsymbol{\vartheta}^{(i)}} \sum_{t=1}^{N} \beta_t^{(i)\star} \cdot \left(y(t) - \boldsymbol{\varphi}(\boldsymbol{x}(t))^T \boldsymbol{\vartheta}^{(i)}\right)^2 \tag{7}$$
$$\text{subject to } \vartheta_j^{(i)} = 0 \text{ if } s_j^{(i)} = 2,\ j = 1,\dots,n,$$

where $\beta_t^{(i)\star}$, $i = 1,\dots,N_M$, $t = 1,\dots,N$, is retrieved from $\sigma_t^\star$, $t = 1,\dots,N$, based on (3). The set of parameters $\boldsymbol{\vartheta}^{(i)\star}$, $i = 1,\dots,N_M$, defines the *target SNARX model*.

The optimization problem (6) is a mixed integer program, which is typically computationally intractable due to its combinatorial complexity. Indeed, the SNARX structure λ involves $N \times N_M$ binary variables for $\boldsymbol{\sigma}$, plus $n \times N_M$ for S. Typically, N is the factor most affecting the combinatorial complexity of the problem, since $N \gg n, N_M$. As a consequence, the sample-mode mapping is the most critical aspect of the problem, since switchings can occur at arbitrary times. However, denoting by $\mathcal{T}_s^\circ \subseteq \{1,\dots,N\}$ the set of switching time instants in the observed data, it is typically true that $|\mathcal{T}_s^\circ| \ll N$.

In view of this, we address the SNARX identification problem (6) using an iterative two-stage approach, where at each iteration the identification is first carried out by restricting the possible switching occurrences at a limited (small) number of time instants (thus significantly reducing the combinatorial complexity of the problem), and then, based on the results of this operation, the set of allowed switching times is refined. In the first stage a randomized algorithm is employed for the estimation of the SNARX model best fitting the available data, given that the switching locations are restricted to be in the set $\mathcal{T}_s = \{t_k\}_{k=1}^{N_s}$, with $1 < t_1 < t_2 < \ldots < t_{N_s} \leq N$ and $N_s \ll N$. The information resulting from the first stage is used to refine the switchings positioning defined by $\mathcal{T}_s$ before a new execution of the first stage is carried out. This is done by means of a split-and-merge procedure designed to finitely tune the number and the location of the candidate switching times. The rationale is to add further possible switching times in the neighborhood of detected switchings, while at the same time removing candidate switching locations that were not identified as such. By iterating this two-stage procedure, one can progressively improve the identification of the switching locations as well as that of the NARX models associated to the modes. The two stages are described in detail in the next two sections.

3. First stage of the SNARX identification approach: identification for a given $\mathcal{T}_s$

The first stage of the method (preliminarily presented in [4]) is an extension to the SNARX model class of the RaMSS method for NARX model identification described in [7], under the assumption that switchings can occur only at specific time instants. The RaMSS method is a randomized model structure selection approach based on a probabilistic representation of the model structure, whereby a probability distribution representing the likelihood of each model structure to be the true one is iteratively refined by a sample-and-evaluate procedure until convergence to a limit distribution that can be associated to a specific parameterized NARX model structure. Specifically, a collection of independent Bernoullian distributions is employed in the RaMSS to account for the presence (or absence) of each regressor in the model. However, the structure of the combinatorial problem considered here is more complex since some decision variables (namely the elements of the switching signal) can take more than two values, and thus cannot be modeled by plain Bernoullian distributions. Accordingly, we first show (Section 3.1) that the RaMSS can be extended to a more general class of combinatorial optimization problems with non-binary decision variables, and then revisit such a generalization within our SNARX identification context. Notice that this analysis was not part of [4], that focused on algorithmic and implementation aspects.

Remark 3. *In a parametric framework as the one adopted here, it is important to test the ability of an identification algorithm to retrieve the exact model structure, in the ideal condition that the system generating the data actually belongs to the considered class of model structures. Obviously, if this is not the case, there is no such thing as a "true" or "exact" model structure and one can only search for the optimal one in the considered class.*

3.1. Extension of the RaMSS method to combinatorial optimization problems with non-binary decision variables

Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be a collection of n discrete variables with $x_j \in \mathcal{X}_j = \{1, \ldots, m_j\}$, $j = 1, \ldots, n$. Consider a combinatorial optimization problem where the goal is to find a value of $\boldsymbol{x}$ that maximizes a given performance index $\mathcal{J} : \mathcal{X} \to \mathbb{R}^+$, with $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$. If such a value is unique, we can define it as:

$$\boldsymbol{x}^\star = (x_1^\star, \ldots, x_n^\star) = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \mathcal{J}(\boldsymbol{x}). \tag{8}$$

Let us introduce a random variable $\gamma_j \sim \texttt{Categorical}(\boldsymbol{\pi}_j)$[1] for each term x_j, where $\boldsymbol{\pi}_j = (\pi_j^{(1)}, \ldots, \pi_j^{(m_j)})$ and $\pi_j^{(i)}$ represents the probability that x_j takes the i-th value ($\sum_{i=1}^{m_j} \pi_j^{(i)} = 1$). If we assume that the γ_j variables are independent[2], then the probability that the collection of random variables $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)$ takes value $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{X}$ is uniquely defined by $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_n)$. More precisely, we have that

$$\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}) = \prod_{j=1}^{n} \prod_{i=1}^{m_j} \left(\pi_j^{(i)}\right)^{\beta_j^{(i)}}, \tag{9}$$

where $\beta_j^{(i)} = 1$ if $x_j = i$, and 0 otherwise.

The expected performance of $\boldsymbol{\gamma}$ can then be computed as follows:

$$\mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma})] = \sum_{\boldsymbol{x} \in \mathcal{X}} \mathcal{J}(\boldsymbol{x}) \mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}).$$

The value of $\mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma})]$ is a function of $\mathbb{P}_{\boldsymbol{\gamma}}$, and its maximum is obtained if the distribution $\mathbb{P}_{\boldsymbol{\gamma}}$ is such that all the probability mass is concentrated on $\boldsymbol{x}^\star$, which can be obtained for an appropriate choice of the parameters in $\boldsymbol{\pi}$. In view of this, the original optimization problem (8) is equivalent to

$$\boldsymbol{x}^\star = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{P}^\star_{\boldsymbol{\gamma}}(\boldsymbol{x}), \tag{10}$$

where

$$\mathbb{P}^\star_{\boldsymbol{\gamma}} = \arg\max_{\mathbb{P}_\gamma} \mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma})]$$

is called the target limit distribution. Now, let

$$\delta_j^{(i)} = \mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma}) | \gamma_j = i] - \mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma}) | \gamma_j \neq i] \tag{11}$$

for $i = 1, \ldots, m_j$, $j = 1, \ldots, n$, where the conditional expectations are set equal to 0 if the conditional event has 0 probability to happen.

Theorem 3.1. *Let $\mathbb{P}_{\boldsymbol{\gamma}}$ be the probability distribution over $\mathcal{X}$ defined according to* (9)*. Then, there exists $\varrho \in (0, 1)$ such that if $\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}^\star) \geq \varrho > \max_{\boldsymbol{x} \in \mathcal{X} \setminus \{\boldsymbol{x}^\star\}} \frac{\mathcal{J}(\boldsymbol{x})}{\mathcal{J}(\boldsymbol{x}^\star)}$, it holds that $\delta_j^{(i)} > 0$ if $x_j^\star = i$ and $\delta_j^{(i)} < 0$ otherwise, $i = 1, \ldots, m_j$, $j = 1, \ldots, n$.*

Proof 3.1.1. *See Appendix A.1.*

Theorem 3.1 suggests that, when $\mathbb{P}_{\boldsymbol{\gamma}}$ associated with $\boldsymbol{\pi}$ is sufficiently close to $\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}^\star)$, then the sign of $\delta_j^{(i)}$ provides a reliable information for tuning the $\pi_j^{(i)}$ parameters towards those in $\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}^\star)$. This information can then be used to iteratively refine $\pi_j^{(i)}(k)$ (where k is the iteration index) according to the following update rule:

$$\pi_j^{(i)}(k+1) = \pi_j^{(i)}(k) + \chi \delta_j^{(i)}, \tag{12}$$

where $\chi > 0$. In order for the Categorical distribution to be well defined, a normalization step is required after the application of (12), so that $0 \leq \pi_j^{(i)}(k+1) \leq 1$ and $\sum_{i=1}^{m_j} \pi_j^{(i)}(k+1) = 1$.

[1] A categorical random variable can take one of n possible values (or categories), with the probability of each category separately specified. The outcomes are often numbered for convenience, *e.g.* from 1 to n. The parameters specifying the probabilities of each possible outcome must be in the range $[0, 1]$, and must sum to 1. The categorical distribution is the generalization of the Bernoulli distribution for $n > 2$.

[2] The introduced probability distribution quantifies our belief regarding the fact that $x_j^\star$ takes a specific value. By assuming the independence of the γ_j variables, we are not letting our belief regarding one specific variable affect the belief for the remaining ones.

Theorem 3.2. *Let $\mathbb{P}_{\boldsymbol{\gamma}}$ be the probability distribution over $\mathcal{X}$ defined according to* (9)*, and assume that $\boldsymbol{\pi}$ is such that $\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}^\star) \geq \varrho$, where ϱ is a value for which Theorem 3.1 holds. Then, the local convergence to the target limit distribution $\mathbb{P}^\star_{\boldsymbol{\gamma}}$ is guaranteed by the iterative application of* (12) *starting from $\boldsymbol{\pi}$.*

Proof 3.2.1. *See Appendix A.2.*

3.2. Application of the extended RaMSS method to SNARX identification

Let $\mathcal{T}_s = \{t_k\}_{k=1}^{N_s}$ with $1 < t_1 < t_2 < \ldots < t_{N_s} \leq N$, be the candidate switching locations, and let $I_1 = \{t \mid 1 \leq t < t_1\}$, $I_k = \{t \mid t_{k-1} \leq t < t_k\}$, $k = 2, \ldots, N_s$, and $I_{N_s+1} = \{t \mid t_{N_s} \leq t \leq N\}$, be the $N_s + 1$ time intervals induced by $\mathcal{T}_s$. Define also the corresponding set of admissible switching signals:

$$\Sigma_{\mathcal{T}_s} = \{\boldsymbol{\sigma} : \sigma_{t'} = \sigma_{t''},\ \forall t', t'' \in I_k,\ k = 1, \ldots, N_s + 1\}.$$

One can associate a mode κ_k to each time interval I_k, $k = 1, \ldots, N_s + 1$, and define vector $\boldsymbol{\kappa} = [\kappa_1, \ldots, \kappa_{N_s+1}] \in \{1, \ldots, N_M\}^{N_s+1}$. Then, with a slight abuse of notation, the SNARX model structure λ can be re-parameterized as $\lambda = (\kappa, S)$. Accordingly, our goal is to find the best SNARX model with switching signal in $\Sigma_{\mathcal{T}_s}$:

$$\lambda^\star = \left(\boldsymbol{\kappa}^\star, S^\star\right) = \arg\max_{\lambda \in \Lambda} \mathcal{J}(\lambda). \tag{13}$$

where we set $\Lambda = \{1, \ldots, N_M\}^{N_s+1} \times \mathcal{S}$.

Note that problem (13) has the same structure of (8) and therefore can be addressed in the explained probabilistic framework. To this purpose, let $\boldsymbol{\gamma} = (\boldsymbol{\xi}, \boldsymbol{\rho})$, where $\boldsymbol{\xi}$ is a discrete random variable taking values in $\{1, \ldots, N_M\}^{N_s+1}$ according to $\mathbb{P}_{\boldsymbol{\xi}}$ that accounts for the mode switchings, and $\boldsymbol{\rho}$ is a discrete variable taking values in $\mathcal{S}$ according to $\mathbb{P}_{\boldsymbol{\rho}}$ that accounts for the structures of the N_M modes.

If we assume that the mode switching and the local model structures are independent[3], we can express $\mathbb{P}_{\boldsymbol{\gamma}}$ as:

$$\mathbb{P}_{\boldsymbol{\gamma}}(\lambda) = \mathbb{P}_{\boldsymbol{\xi}}(\boldsymbol{\kappa}) \cdot \mathbb{P}_{\boldsymbol{\rho}}(S), \tag{14}$$

where $\lambda = (\boldsymbol{\kappa}, S) \in \Lambda$.

3.2.1. Parametrization of $\mathbb{P}_{\boldsymbol{\xi}}$

The random variable $\boldsymbol{\xi}$ is a vector of $N_s + 1$ random variables ξ_k, $k = 1, \ldots, N_s + 1$, each one representing the mode associated to the corresponding time interval I_k. We can then introduce vector $\boldsymbol{\eta}_k = [\eta_k^{(1)}, \ldots, \eta_k^{(N_M)}]$, where $\eta_k^{(i)}$ represents the probability of assigning mode i to sub-period I_k and is denoted as Mode Extraction Probability (MEP) in the following. Clearly, $\sum_{i=1}^{N_M} \eta_k^{(i)} = 1$.

If we assume independence between the random variables ξ_k, $k = 1, \ldots, N_s + 1$, then, the probability distribution of $\boldsymbol{\xi}$ is given by

$$\mathbb{P}_{\boldsymbol{\xi}}(\boldsymbol{\kappa}) = \mathbb{P}_{\boldsymbol{\xi}}([\kappa_1, \ldots, \kappa_{N_s+1}]) = \prod_{k=1}^{N_s+1} \prod_{i=1}^{N_M} \left(\eta_k^{(i)}\right)^{b_k^{(i)}}, \tag{15}$$

where $b_k^{(i)} = 1$ if $\kappa_k = i$, and 0 otherwise. $\mathbb{P}_{\boldsymbol{\xi}}$ is uniquely defined by matrix

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_{N_s+1} \end{bmatrix} \in \mathbb{R}^{(N_s+1) \times N_M}. \tag{16}$$

[3] As done previously, we assume that the elements of the model structure λ are all independent, in the sense that our belief regarding the value of one element should not affect the others. The usage of probability in this framework is not meant to describe any correlation structure existing in the system, but it is only instrumental to the functioning of the proposed method.

3.2.2. Parametrization of $\mathbb{P}_{\boldsymbol{\rho}}$

Similarly to the RaMSS method, we associate each regressor $\varphi_j(\boldsymbol{x}(t))$ in each mode i to a Categorical distribution[4] $\rho_{j,i} \sim \texttt{Categorical}(\boldsymbol{\mu}_{j,i})$, where $\boldsymbol{\mu}_{j,i} = (\mu_{j,i}^{(1)}, \mu_{j,i}^{(2)})$. The outcome 1 encodes the case that $\varphi_j(\boldsymbol{x}(t))$ is present in the ith local model structure ($s_j^{(i)} = 1$), while the outcome 2 encodes the case that $\varphi_j(\boldsymbol{x}(t))$ is absent ($s_j^{(i)} = 2$). Clearly, $\mu_{j,i}^{(1)} + \mu_{j,i}^{(2)} = 1$. In the following, the probabilities $\mu_{j,i}^{(1)}$ will be denoted as Regression Inclusion Probabilities (RIPs).

The collection of all parameters $\mu_{j,i}^{(1)}$, and $\mu_{j,i}^{(2)}$, $j = 1, \ldots, n$, $i = 1, \ldots, N_M$ defines a matrix

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1^{(1)} & \boldsymbol{\mu}_1^{(2)} \\ \vdots & \vdots \\ \boldsymbol{\mu}_{N_M}^{(1)} & \boldsymbol{\mu}_{N_M}^{(2)} \end{bmatrix} \in \mathbb{R}^{(N_M \cdot n) \times 2}, \tag{17}$$

where $\boldsymbol{\mu}_i^{(l)} = [\mu_{1,i}^{(l)}, \ldots, \mu_{n,i}^{(l)}]^T \in \mathbb{R}^n$, $i = 1, \ldots, N_M$, $l = 1, 2$. If the random variables $\rho_{j,i}$, $j = 1, \ldots, n$, are independent, then the probability distribution $\mathbb{P}_{\boldsymbol{\rho}^{(i)}}$ of $\boldsymbol{\rho}^{(i)} = [\rho_{1,i}, \ldots, \rho_{n,i}]^T$ is given by

$$\mathbb{P}_{\boldsymbol{\rho}^{(i)}}(\boldsymbol{s}^{(i)}) = \mathbb{P}_{\boldsymbol{\rho}^{(i)}}([s_1^{(i)}, \ldots, s_n^{(i)}]) = \prod_{j:\varphi_j \in \boldsymbol{s}^{(i)}} \mu_{j,i}^{(1)} \prod_{j:\varphi_j \notin \boldsymbol{s}^{(i)}} \mu_{j,i}^{(2)} = \prod_{j=1}^{n} \prod_{l=1}^{2} \left(\mu_{j,i}^{(l)}\right)^{\zeta_{j,i}^{(l)}},$$

where $\zeta_{j,i}^{(l)} = 1$ if $s_j^{(i)} = l$, and 0 otherwise.
Under the assumption of independence between mode structures, we then have that the probability distribution of the random vector $\boldsymbol{\rho}$ associated with the SNARX model structure is given by

$$\mathbb{P}_{\boldsymbol{\rho}}(S) = \prod_{i=1}^{N_M} \mathbb{P}_{\boldsymbol{\rho}^{(i)}}(\boldsymbol{s}^{(i)}) = \prod_{i=1}^{N_M} \prod_{j=1}^{n} \prod_{l=1}^{2} \left(\mu_{j,i}^{(l)}\right)^{\zeta_{j,i}^{(l)}}. \tag{18}$$

Therefore, $\mathbb{P}_{\boldsymbol{\rho}}$ is uniquely defined by matrix $\boldsymbol{\mu}$ in (17).

3.2.3. Tuning of $\mathbb{P}_{\boldsymbol{\gamma}}$

The overall probability distribution $\mathbb{P}_{\boldsymbol{\gamma}}$ in (14) is parameterized by $\eta_k^{(i)}$ and $\mu_{j,i}^{(l)}$, $k = 1, \ldots, N_s + 1$, $j = 1, \ldots, n$, $i = 1, \ldots, N_M$, and $l = 1, 2$. By setting

$$\boldsymbol{\pi} = \left(\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{N_s+1}, \boldsymbol{\mu}_{1,1}, \ldots, \boldsymbol{\mu}_{n,1}, \ldots, \boldsymbol{\mu}_{1,N_M}, \ldots, \boldsymbol{\mu}_{n,N_M}\right),$$

$\mathbb{P}_{\boldsymbol{\gamma}}$ can be succinctly written in the form of (9) as

$$\mathbb{P}_{\boldsymbol{\gamma}}(\lambda) = \prod_{j=1}^{N_s+1+N_M \cdot n} \prod_{i=1}^{m_j} \left(\pi_j^{(i)}\right)^{\beta_j^{(i)}}, \tag{19}$$

where

$$m_j = \begin{cases} N_M, & j \leq N_s + 1 \\ 2, & \text{otherwise} \end{cases},$$

and the $\beta_j^{(i)}$ values are the element of a vector $\boldsymbol{\beta}$ defined as

$$\boldsymbol{\beta} = \left[b_1^{(1)}, \ldots, b_1^{(N_M)}, \ldots, b_{N_s+1}^{(1)}, \ldots, b_{N_s+1}^{(N_M)}, \zeta_{1,1}^{(1)}, \zeta_{1,1}^{(2)}, \ldots, \zeta_{n,1}^{(1)}, \zeta_{n,1}^{(2)}, \ldots, \zeta_{1,N_M}^{(1)}, \zeta_{1,N_M}^{(2)}, \ldots, \zeta_{n,N_M}^{(1)}, \zeta_{n,N_M}^{(2)}\right].$$

In this view, the results stemming from Theorems 3.1 and 3.2 can be used to develop suitable tuning rules for $\mathbb{P}_{\boldsymbol{\gamma}}$, as follows.

[4]Notice that a Categorical distribution with only two outcomes is equivalent to a Bernoullian distribution.

The randomized procedure involves extracting and evaluating samples $\lambda = (\boldsymbol{\kappa}, S)$ of the random variable $\boldsymbol{\gamma} = (\boldsymbol{\xi}, \boldsymbol{\rho})$, according to the distribution $\mathbb{P}_{\boldsymbol{\gamma}}$, to gather information for tuning $\mathbb{P}_{\boldsymbol{\gamma}}$. To update the MEP $\eta_k^{(i)}$ we employ a sampled version of the index:

$$\delta_k^{(i)} = \mathbb{E}_{\mathbb{P}_\gamma}\left[\mathcal{J}(\boldsymbol{\gamma})|\xi_k = i\right] - \mathbb{E}_{\mathbb{P}_\gamma}\left[\mathcal{J}(\boldsymbol{\gamma})|\xi_k \neq i\right], \tag{20}$$

which compares the average performance of $\boldsymbol{\gamma}$ in case mode i is assigned to time period I_k with the average performance of $\boldsymbol{\gamma}$ in the opposite case. If $\delta_k^{(i)} > 0$ it pays off to apply the mentioned mode assignment. Since, in practice, index $\delta_k^{(i)} > 0$ can only be calculated in an approximate sampled version, we use this information in a conservative way, defining the following tuning rule:

$$\eta_k^{(i)} \leftarrow \eta_k^{(i)} + \chi\delta_k^{(i)}, \tag{21}$$

where the step size $\chi > 0$ is a design parameter.

Similarly, we update $\mu_{j,i}^{(l)}$ based on an aggregate index that weighs the advantages of picking regressor $\varphi_j(\boldsymbol{x}(t))$ for mode i:

$$\ell_{j,i}^{(l)} = \mathbb{E}_{\mathbb{P}_\gamma}\left[\mathcal{J}(\boldsymbol{\gamma})|\rho_{j,i} = l\right] - \mathbb{E}_{\mathbb{P}_\gamma}\left[\mathcal{J}(\boldsymbol{\gamma})|\rho_{j,i} \neq l\right]. \tag{22}$$

Index $\ell_{j,i}^{(l)}$ compares the average performance of $\boldsymbol{\gamma}$ in case $\varphi_j(\boldsymbol{x}(t))$ is included in the model structure for mode i with the average performance of $\boldsymbol{\gamma}$ in the opposite case. As with $\delta_k^{(i)}$, only an approximate sampled version of $\ell_{j,i}^{(l)}$ can be calculated in practice, which motivates the use of an update law which balances the prior knowledge with the new estimate of the index:

$$\mu_{j,i}^{(l)} \leftarrow \mu_{j,i}^{(l)} + \chi\ell_{j,i}^{(l)}. \tag{23}$$

The above update rules guarantee (local) convergence to the target limit distribution

$$\mathbb{P}_{\boldsymbol{\gamma}}^\star = \arg\max_{\mathbb{P}_\gamma} \mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma})]$$

as stated in the theorem below, which follows directly from Theorems 3.1 and 3.2.

Theorem 3.3. *Let $\mathbb{P}_{\boldsymbol{\gamma}}$ be the probability distribution over Λ defined according to* (19)*, which depends on $\boldsymbol{\eta}$ in* (16) *and $\boldsymbol{\mu}$ in* (17)*. Then there exists $\varrho \in (0, 1)$, such that if $\mathbb{P}_{\boldsymbol{\gamma}}(\lambda^\star) \geq \varrho$ the iterative application of* (21) *and* (23) *will make $\mathbb{P}_{\boldsymbol{\gamma}}$ converge to the target limit distribution $\mathbb{P}_{\boldsymbol{\gamma}}^\star$.*

3.3. Guidelines for parameter settings

Choosing the correct step size χ in the update of the MEPs and RIPs is crucial for the convergence speed of the algorithm, as discussed in [8, 7] with reference to the RaMSS algorithm. In the early stages, the algorithm should be allowed to freely explore the solution space in order to gather as much information as possible. In this exploration phase, however, the correction terms $\delta_k^{(i)}$ and $\ell_{j,i}^{(l)}$ may vary erratically, and thus their influence in the update equations has to be limited. Later on, when the suggested corrections become more stable, the step size should be incremented to accelerate convergence. In the light of these remarks, we adaptively tune χ taking into account the performance dispersion of the associated SNARX models. Specifically,

$$\chi = \frac{1}{10\left(\mathcal{J}_{\text{best}} - \overline{\mathcal{J}}\right) + 0.1} \tag{24}$$

where $\mathcal{J}_{\text{best}}$ and $\overline{\mathcal{J}}$ are, respectively, the best value and the mean value for $\mathcal{J}$ evaluated on the extracted samples for $\boldsymbol{\gamma}$.

The convergence speed of the algorithm is also influenced by the choice of K_λ in (5). As an alternative to a classical trial-and-error approach (as suggested in [4]), we here provide a simple tuning procedure for K_λ, designed to allow a better discrimination between models with similar performance. Let us denote by $OM(x) = \lfloor \log_{10}(x) \rfloor$ the order of magnitude of a nonnegative number x. Parameter K_λ is tuned at the first iteration of the algorithm according to the minimum $OM(\mathcal{L}(\lambda))$, computed based on the extracted SNARX models. Specifically,

$$K_\lambda = 10^{-(\min(OM(\mathcal{L}(\lambda)))+1)}. \tag{25}$$

Regarding the initialization of the probability distribution, we set the parameters $\mu_{j,i}^{(l)}$ to equal small values, to encourage the extraction of small models at the early stages of the algorithm, see also [8]. As for the $\eta_k^{(i)}$, in the absence of any a-priori assumption on the switching signal, we attribute equal probabilities $\eta_k^{(i)} = 1/N_M$ to all modes in each sub-period I_k.

Concerning the choice of $\mathcal{T}_s$, we initially place the candidate switching time instants uniformly over $\{1, N\}$, dividing the time horizon in sub-periods of equal length. In choosing this placement, one can take advantage from the *a priori* knowledge on the minimum dwell time of the system in a mode. Indeed, in practical applications, where the mode switching is caused by activation/deactivation of devices and system reconfiguration, switchings cannot generally occur at consecutive time steps and a certain time must be allowed to pass between switchings. If such information is available, the maximum sub-period length should be upper bounded by the minimum dwell time, so that at most one switching can occur inside a given sub-period, thus reducing the number of mixed sub-periods, as discussed in Section 4. Notice also that a significant reduction of the combinatorial complexity can be leveraged for what concerns the switching signal (only switching signals that do not violate the minimum dwell time are acceptable).

3.4. An heuristic implementation

The convergence speed of the algorithm can be improved by updating the probability distribution associated to the model structures of the modes (see equation (22)) *separately* for each mode, based on a local performance index of the following type:

$$\mathcal{J}^{(i)}(\boldsymbol{\kappa}, \boldsymbol{s}^{(i)}) = e^{-K_i \mathcal{L}^{(i)}(\boldsymbol{\kappa}, \boldsymbol{s}^{(i)})}, \tag{26}$$

as opposed to the full $\mathcal{J}(\lambda)$. In expression (26) $K_i > 0$ is a design parameter that can be tuned similarly to (25):

$$K_i = 10^{-\left(\min\left(OM\left(\mathcal{L}^{(i)}(\boldsymbol{\kappa}, \boldsymbol{s}^{(i)})\right)\right)+1\right)}. \tag{27}$$

As a result, the update term $\ell_{j,i}^{(l)}$ is modified as follows:

$$\tilde{\ell}_{j,i}^{(l)} = \mathbb{E}_{\mathbb{P}_\gamma}\left[\mathcal{J}^{(i)}(\boldsymbol{\kappa}, \boldsymbol{\rho}^{(i)}) | \rho_{j,i} = l, \boldsymbol{\xi} = \boldsymbol{\kappa}\right] - \mathbb{E}_{\mathbb{P}_\gamma}\left[J^{(i)}(\boldsymbol{\kappa}, \boldsymbol{\rho}^{(i)}) | \rho_{j,i} \neq l, \boldsymbol{\xi} = \boldsymbol{\kappa}\right], \tag{28}$$

which, with reference to mode i, compares the average performance of model structures that include $\varphi_j(\boldsymbol{x}(t))$ with that of the remaining structures. Observe that the performance evaluation depends on the switching signal as well, which defines the segments of the data-set that are assigned to mode i. The resulting RIP update law is:

$$\mu_{j,i}^{(l)} \leftarrow \mu_{j,i}^{(l)} + \nu_i \tilde{\ell}_{j,i}^{(l)} \tag{29}$$

where $\nu_i > 0$ is the step size for mode i defined (similarly to (24)) as:

$$\nu_i = \frac{1}{10\left(\mathcal{J}_{\text{best}}^{(i)} - \overline{\mathcal{J}}^{(i)}\right) + 0.1} \tag{30}$$

with $\mathcal{J}_{\text{best}}^{(i)}$ and $\overline{\mathcal{J}}^{(i)}$ being respectively, the best value and the mean value for $\mathcal{J}^{(i)}$ evaluated on the extracted samples for $\boldsymbol{\gamma}$.

In this case, the local convergence of $\mathbb{P}_{\boldsymbol{\gamma}}$ to the target limit distribution $\mathbb{P}_{\boldsymbol{\gamma}}^\star$ is not guaranteed, essentially due to possible sign differences between $\ell_{j,i}^{(l)}$ and $\tilde{\ell}_{j,i}^{(l)}$ (see Appendix B). However, as discussed in Section 3.5, the experimental evidence indicates that this occurs relatively seldom and scarcely affects the overall identification results (see Table 2 and Figure 2). This justifies the adoption of this heuristic version of the algorithm in view of its more favorable computational characteristics.

3.5. Example 1: $\mathcal{T}_s^\circ \subseteq \mathcal{T}_s$

Recalling that $\mathcal{T}_s^\circ$ identifies the set of true switching time instants, it can happen that $\mathcal{T}_s^\circ \subseteq \mathcal{T}_s$ or, more frequently, $\mathcal{T}_s^\circ \nsubseteq \mathcal{T}_s$. We discuss here the former condition, while the latter one is the subject of the next subsection.

Consider the following SNARX system [14], which switches between a linear mode 1:

$$y(t) = -0.905y(t-1) + 0.9u(t-1) + e(t),$$

and a nonlinear mode 2:

$$y(t) = -0.4y(t-1)^2 + 0.5u(t-1) + e(t),$$

where $e(t)$ is a zero mean Gaussian noise of variance 0.012 and $u(t)$ is uniformly distributed in the interval $[0, 1]$. An observation window of $N = 2000$ samples has been collected, which contains 4 switchings, at $t = 400$ (from mode 1 to mode 2), $t = 1500$ (from mode 2 to mode 1), $t = 1600$ (from mode 1 to mode 2), and $t = 1700$ (from mode 2 to mode 1), so that $\mathcal{T}_s^\circ = \{400, 1500, 1600, 1700\}$. In the absence of any *a priori* information regarding the candidate switching times, we uniformly divide the time horizon in 20 sub-periods of length 100, setting $t_k = 100k$, $k = 1, \ldots, 19$. Notice that, while this hugely simplifies the combinatorial complexity of the problem, more than 1 million different possible switching signals are nevertheless compatible with the defined $\mathcal{T}_s$. In this case study, the set of pre-defined candidate switchings includes the true ones. The design parameters have been set to $n_y = n_u = n_d = 2$ (for a total of 15 possible regressors for each NARX model, *i.e.,* $\{1,\ y(t-1),\ y(t-2),\ u(t-1),\ u(t-2),\ y(t-1)^2,\ y(t-1)y(t-2),\ y(t-1)u(t-1),\ y(t-1)u(t-2),\ y(t-2)^2,\ y(t-2)u(t-1),\ y(t-2)u(t-2),\ u(t-1)^2,\ u(t-1)u(t-2),\ u(t-2)^2\}$). Furthermore, the initial MEPs are all set to 0.5, and the initial RIPs to 0.0667.

One of the nice features of the presented approach is that it is capable of extracting useful information on the model from partially correct extracted models. To emphasize this property, consider Figure 1 which shows the probability distribution state, in terms of the scalar parameters $\eta_k^{(i)}$ and $\mu_{j,i}^{(l)}$, $k = 1, \ldots, N_s + 1$, $j = 1, \ldots, n$, $i = 1, \ldots, N_M$, $l = 1, 2$, obtained by interrupting the algorithm well before convergence, at the iteration when the correct model structure is first extracted. All the information gathered up to this point to tune the probability distribution is based on extracted SNARX models none of which has the correct structure. All the same, this information appears to be sufficient to drive the algorithm toward the true model structure $\lambda^\star$. Indeed, some of the sub-periods have been already mapped on the correct mode with high confidence and the algorithm is looking for the model structure S on a restricted area of the solution space $\mathcal{S}$ which actually contains $S^\star$. This confirms the effectiveness of the chosen parametrization of $\mathbb{P}_{\boldsymbol{\gamma}}$ and of the proposed tuning rules. It proves also that the result in Theorem 3.3 is somewhat conservative, since in this example the algorithm is converging toward the target limit distribution even if it has been initialized with $\mathbb{P}_{\boldsymbol{\gamma}}(\lambda^\star) \cong 0$.

Table 1 reports some aggregate results obtained from 100 runs of the algorithm on the same data realization. The proposed algorithm performs well in both the sample-mode assignment and the local NARX model identification, and it does so by exploring a small fraction of the total number of possible switching signals and models. As for the nonlinear mode, the algorithm sporadically (2 times out of 100) fails to select the nonlinear term $y(t-1)^2$ in favor of $y(t-1)$, for a slight performance loss. Indeed, $\mathcal{L}^{(2)}$ takes the value 0.0119 for the wrong local model and the value 0.0118 for the correct one, causing the algorithm to be trapped in the found local minimum due to the almost negligible difference between them. It is worth noticing that despite the occasional failures in identifying mode 2, the algorithm has always been able to capture from the data the existence of two different modes, and to assign them correctly to the sub-periods.

A similar MC analysis has been carried out by considering this time the heuristic implementation introduced in Section 3.4. As one can note from Table 2, which reports the aggregated results of this analysis, the heuristic implementation provides comparable results in terms of accuracy, albeit at a lower computational cost. Figure 2 compares the two versions of the proposed algorithm, by enumerating the occurrences of a sign difference between the two update factors $\ell_{j,i}^{(l)}$ and $\tilde{\ell}_{j,i}^{(l)}$ over the MC runs. The frequency of these events decreases with iterations, so that no significant differences are expected in the algorithm outcomes at convergence. Based on this evidence, the heuristic implementation has been employed in the rest of the paper for computational convenience.

3.6. Example 1 (contd.): $\mathcal{T}_s^\circ \nsubseteq \mathcal{T}_s$

Suppose now that the switchings occur at $t = 350$ (from mode 1 to mode 2), $t = 1450$ (from mode 2 to mode 1), $t = 1600$ (from mode 1 to mode 2), and $t = 1750$ (from mode 2 to mode 1). Notice that using the previously defined uniform placement of the switching times, only one of the true switchings is encompassed, while the others occur exactly in the middle of the 4th, 15th, and 18th sub-periods.

Table 3 reports the results of a single run of the identification method. Apparently, the presence of sub-periods assigned to mode 1 but containing also samples associated to mode 2 prevents the algorithm from correctly identifying

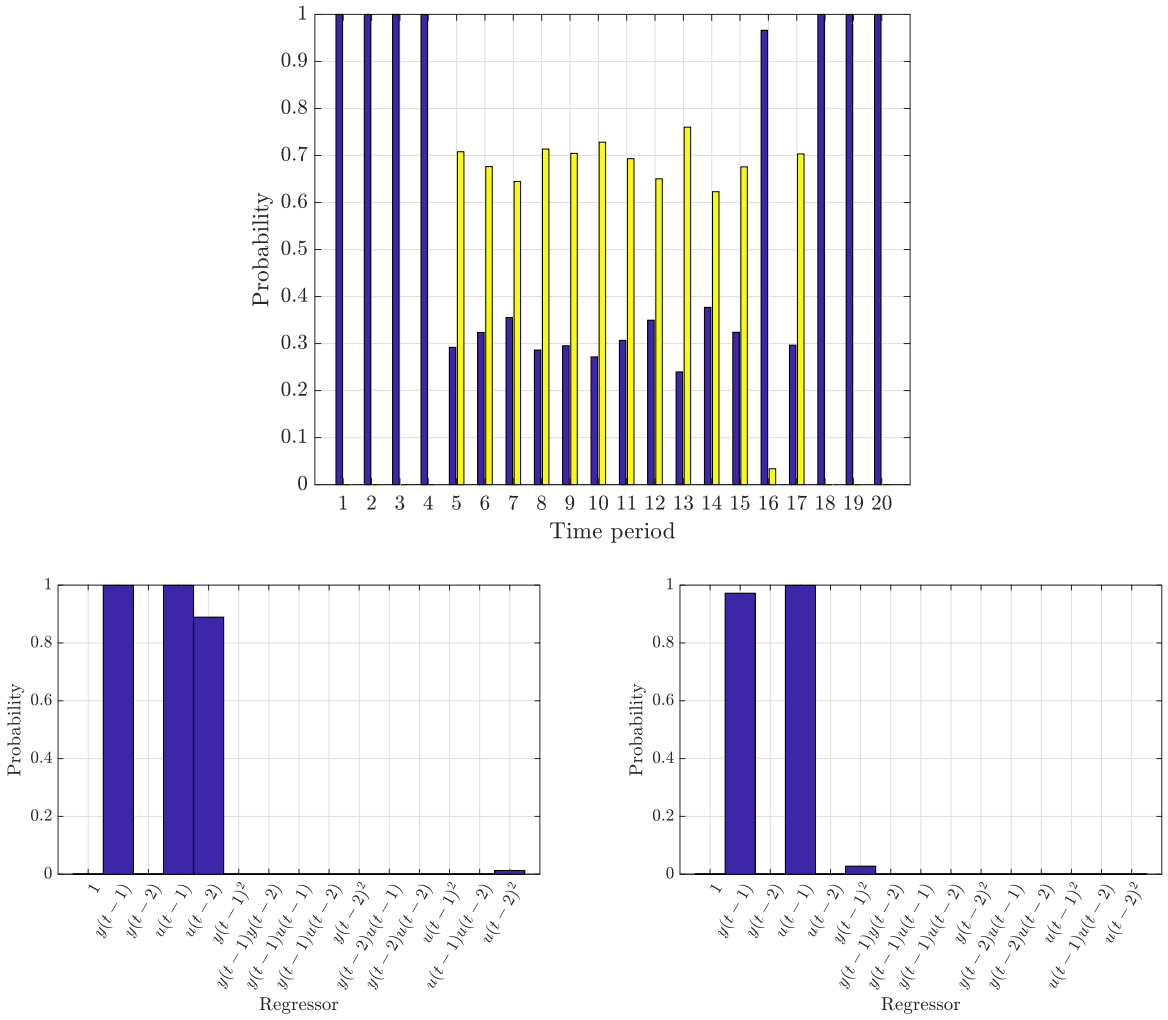

Figure 1: Example 1: MEP and RIP values at the iteration when the correct model structure is first extracted. Top: MEPs of modes 1 (blue) and 2 (yellow). Bottom, from left to right: RIPs of mode 1, RIPs of mode 2.

Table 1: Example 1: $\mathcal{T}_s^\circ \subseteq \mathcal{T}_s$. Monte Carlo simulation results.

Average elapsed time [s]	41.25
Percentage of correct selection of $\boldsymbol{\kappa}$	100%
Average # of explored switching sequences	12520
Total # of allowed switching sequences	1048576
Percentage of correct selection of $\boldsymbol{s}^{(1)}$	100%
Average # of explored model structures for mode 1	790.62
Total # of possible model structures for mode 1	32768
Percentage of correct selection of $\boldsymbol{s}^{(2)}$	98%
Average # of explored model structures for mode 2	1005.9
Total # of possible model structures for mode 2	32768

the local model assigned to the first mode (a redundant regressor is added to the model, although with a very small coefficient, indicating its relatively smaller importance). Despite this failure in estimating the linear local model, the method performs well in assigning the samples to the modes. Indeed, the obtained $\boldsymbol{\kappa}^\star$ is correct in 17 out of 20 periods

Table 2: Example 1: $\mathcal{T}_s^\circ \subseteq \mathcal{T}_s$. Monte Carlo simulation results - heuristic implementation.

Average elapsed time [s]	30.16
Percentage of correct selection of $\boldsymbol{\kappa}$	100%
Average # of explored switching sequences	11156
Total # of allowed switching sequences	1048576
Percentage of correct selection of $\boldsymbol{s}^{(1)}$	100%
Average # of explored model structures for mode 1	646.27
Total # of possible model structures for mode 1	32768
Percentage of correct selection of $\boldsymbol{s}^{(2)}$	95%
Average # of explored model structures for mode 2	752.42
Total # of possible model structures for mode 2	32768

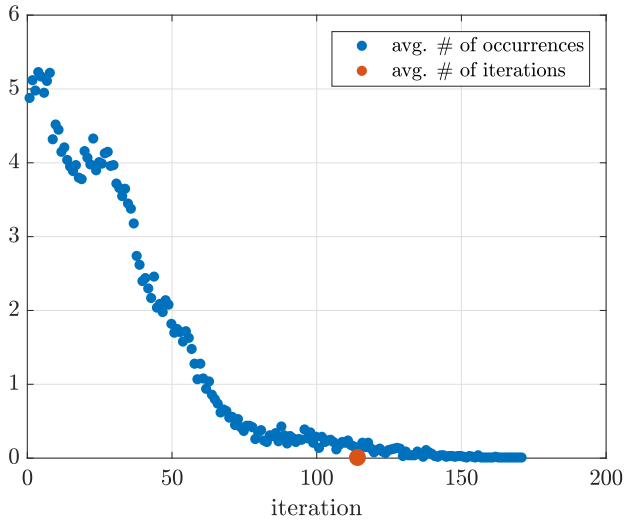

Figure 2: Example 1: average number of occurrences of a sign difference between $\epsilon_{j,i}^{(l)}$ and $\tilde{\epsilon}_{j,i}^{(l)}$ in the MC runs at each iteration. The red marker indicates the average number of iterations required to solve the identification problem.

and yields a 50% correct classification of the samples in the remaining three sub-periods. This error (which involves 150 out of 2000 samples, *i.e.* 7.5% of the data) is unavoidable given the placement of the true switchings exactly in the middle of the allowed sub-periods.

Table 3: Example 1: $\mathcal{T}_s^\circ \nsubseteq \mathcal{T}_s$. Single run results.

$\mathcal{L}(\lambda)$	0.0154
$\mathcal{L}^{(1)}(\lambda)$	0.0198 ($N_1 = 900$)
$\mathcal{L}^{(2)}(\lambda)$	0.0119 ($N_2 = 1100$)
Detected switching times	400, 1400, 1600, 1700
sub-periods assigned to mode 1	I_k, $k = 1, \dots, 4, 15, 16, 18, \dots, 20$
sub-periods assigned to mode 2	I_k, $k = 5, \dots, 14, 17$
Sample classification error	7.5%
Regressors mode 1	$y(t-1)$, $u(t-1)$, $u(t-2)$
Parameters mode 1	-0.9041, 0.8363, 0.0566
Regressors mode 2	$u(t-1)$, $y(t-1)^2$
Parameters mode 2	0.5093, -0.4137

3.7. Discussion

The presented first stage identification method is effective in both mode assignment and model estimation, provided that $\mathcal{T}_s^\circ \subseteq \mathcal{T}_s$, while an unavoidable approximation error is experienced otherwise. In general, no *a priori* information on the switching times is available and, in principle, a switching could occur at any time instant in $\{1, 2, \ldots, N\}$. In order to encompass this case one could arbitrarily enlarge $\mathcal{T}_s$ towards $\{1, \ldots, N\}$. However, the complexity of the resulting combinatorial problem rapidly increases with the cardinality of $\mathcal{T}_s$ that is employed, making it computationally intractable to sample the set $\{1, 2, \ldots, N\}$ too densely. This poses a practical limit on the modeling accuracy that can be achieved with the method described in this section, since with a sparse $\mathcal{T}_s$ a poor resolution on the switching times is typically obtained, which in turn influences the quality of the identified models (that are tuned on data not fully belonging to the appropriate modes), and motivates the introduction of the second stage.

4. Second stage of the SNARX identification approach: refinement of $\mathcal{T}_s$

Rather than extending $\mathcal{T}_s$ to improve the accuracy of the model, we here suggest to refine it based on the outcome of the identification procedure and then iterate the process. The refinement stage is aimed at improving the resolution of $\mathcal{T}_s$ where required, at the same time keeping its size under control. This is achieved by adopting a denser sampling of the time horizon in the vicinity of the detected switchings and a sparser sampling elsewhere. Notice that, besides improving the resolution of the estimated switching instants, it is also expected that the improvement in the sample-mode assignment will also positively impact the accuracy of the identified local models.

The rationale behind the refinement of $\mathcal{T}_s$ follows from the observations listed below:

- Let two adjacent sub-periods be assigned to different modes, say $\kappa_k = 1$ and $\kappa_{k+1} = 2$. This suggests that the majority of the samples of the first period can be ascribed to mode 1 and similarly that most of the samples in the second period indeed belong to mode 2. This indicates that there is at least one switching between modes 1 and 2 in the time interval spanned by the set $I_k \cup I_{k+1}$, but not necessarily at the common boundary (t_k). Therefore, adding new candidate switching times in the vicinity of t_k may improve the resolution of the algorithm.

- Let two adjacent sub-periods be assigned to the same mode, say $\kappa_k = \kappa_{k+1} = 1$. Then, in the same assumptions as before, no switching from mode 1 to another one can occur in the vicinity of the intermediate point t_k. It is therefore possible to disregard t_k altogether as a candidate switching time.

- Occasionally, the identification procedure may fail to converge to a limit distribution regarding a specific sub-period, so that multiple MEPs have non-zero values. This typically occurs when the sub-period contains data of different modes. In these situations, splitting further the sub-period into smaller sub-periods may facilitate the algorithm in taking its decisions.

Let $\mathcal{T}_s^{(r)}$ be the set of allowed switching time instants at the rth iteration of the overall procedure. Then, after the execution of the identification phase in the first stage, the refinement phase of the mode switching times consists in defining $\mathcal{T}_s^{(r+1)}$ based on the results of the rth identification. $\mathcal{T}_s^{(r+1)}$ is calculated according to the following steps, starting from an empty set:

1. *Detection of switchings*. A switching is detected at t_k if $\kappa_k \neq \kappa_{k+1}$ (*i.e.* two consecutive sub-periods have been assigned to different modes). Accordingly, let $\mathcal{V} = \{t_k \in \mathcal{T}_s^{(r)} | \kappa_k \neq \kappa_{k+1}\}$ be the set of detected switchings.
2. *Detection of unresolved sub-periods*. Sub-period I_k is marked as *unresolved* if the identification algorithm was unable to converge to a limit distribution for ξ_k, within the allotted iterations (although the MEP of one mode could still be significantly larger than the others to allow for a meaningful mode assignment). The auxiliary set $\mathcal{U} \subseteq \mathcal{T}_s^{(r)}$ includes the starting times of such unresolved sub-periods.
3. *Split phase: part a*. For each $t \in \mathcal{V}$, three candidate switching locations are added to $\mathcal{T}_s^{(r+1)}$. More precisely, $\mathcal{T}_s^{(r+1)} \leftarrow \mathcal{T}_s^{(r+1)} \cup \{t^-, t, t^+\}$, with $t^- = t - w$ and $t^+ = t + w$, where w is a design parameter.
4. *Split phase: part b*. For each $t \in \mathcal{U}$, let $t' = \min_{\{t_k \in \mathcal{T}_s^{(r)} | t_k > t\}} t_k$. Now, if $d = t' - t \geq 2$, then $\mathcal{T}_s^{(r+1)} \leftarrow \mathcal{T}_s^{(r+1)} \cup \{t, t^+, t'\}$, where $t^+ = t + \lceil \frac{d}{2} \rceil$. Otherwise, $\mathcal{T}_s^{(r+1)} \leftarrow \mathcal{T}_s^{(r+1)} \cup \{t, t'\}$.

5. *Merge phase*. The elements of $\mathcal{T}_s^{(r)}$ not in $\mathcal{V}$ or $\mathcal{U}$ are not carried over to $\mathcal{T}_s^{(r+1)}$, and are therefore discarded. By doing so, we are implicitly merging consecutive sub-periods, which are assumed not to include mode switchings, according to the current model.

Regarding the split procedure, a possible choice is to use the same w value for each detected switching, setting $w(r+1) = \alpha \min_k |I_k^{(r)}|$, where $I_k^{(r)}$, $k = 1, \ldots, N_s + 1$ are the sub-periods induced by $\mathcal{T}_s^{(r)}$ and $0 < \alpha < 1$ (*e.g.*, $\alpha = 0.5$ to get new sub-periods half as large as the smallest sub-periods of the previous iteration).

The rationale behind the processing of the unresolved sub-periods is as follows. Since the absence of convergence is typically due to the simultaneous presence in a sub-period of an initial portion associated to a mode followed by samples from a different one, the time interval is split into two equal parts to increase the mode unbalance in both time intervals and thus facilitate the mode assignment. However, if the original unresolved sub-period is too short, the time interval is not further divided, trusting that the progressive improvements in the identification of the local models (thanks to the refined positioning of the switchings) will allow the full convergence to a limit distribution in the subsequent iterations.

4.1. Guidelines for parameter settings

The results of the previous identification phase can also be used to set the initial MEPs and RIPs more appropriately before repeating the identification procedure. Indeed, if a sub-period was previously assigned to a specific mode with high confidence (*i.e.*, the corresponding MEP was close to 1 at the previous iteration), then this information should be preserved in the new execution, by setting the corresponding MEP to a large value. All the same, we apply a discounting factor to allow the identification algorithm some flexibility to consider also alternative mode assignments. On the other hand, the MEPs associated to unresolved sub-periods or to newly generated sub-periods (from t^- to t and from t to t^+) are set to be equal for all modes. The following rules formalize these considerations:

- *Detected switchings.* For each $t \in \mathcal{V}$, $\eta_{t^-}^{(i)} = \eta_t^{(i)} = 1/N_M$, $i = 1 \ldots N_M$, while $\eta_{t^+}^{(i)} = p$ for $i = \sigma_{t^+}$ and $\eta_{t^+}^{(j)} = \frac{1-p}{N_M-1}$, for all other modes, where p is a design parameter (*e.g.*, $p = 0.7$) representing the desired confidence level.
- *Unresolved switchings.* For each $t \in \mathcal{U}$, $\eta_t^{(i)} = 1/N_M$, $i = 1 \ldots N_M$, while $\eta_{t'}^{(i)} = p$ for $i = \sigma_{t'}$ and $\eta_{t'}^{(i)} = \frac{1-p}{N_M-1}$, for all other modes. Furthermore, if t^+ exists, $\eta_{t^+}^{(i)} = 1/N_M$, $i = 1 \ldots N_M$.

Regarding the RIPs, they are all set to $1/n$ at each iteration, where n is the number of regressors.

4.2. Example 1 (contd.): Refinement stage

A typical execution of the refinement stage is illustrated in Figure 3, as a continuation of the last example discussed in Section 3.6. In the identification stage, as already discussed, mode switchings were identified at times 400, 1400, 1600, and 1700, three of which being approximations of the true ones, given the coarse division of the time horizon in $\mathcal{T}_s$. The refinement stage halves the 4th, 5th, 14th, 15th, 16th, 17th, and 18th intervals ($w = 50$), and removes the redundant time points separating equal mode assignments, yielding $\mathcal{T}_s^{(2)} = \{350, 400, 450, 1350, 1400, 1450, 1550, 1600, 1650, 1700, 1750\}$. Notice that the total number of switching times has decreased from 19 to 11, thanks to the merging phase. The smallest time sub-periods generated by the refinement are initialized with MEPs assigning the same *a priori* probability to all modes, while the MEPs of the other ones (where a clear decision was made in the first run) are only partially discounted to allow some further flexibility to the algorithm. Notice that based on $\mathcal{T}_s^{(2)}$ a much sharper detection of the true switching times is indeed possible.

5. Simulation results

In this section several simulation examples are discussed to show the effectiveness of the proposed iterative method. First, the presented procedure is applied to the example introduced in Section 3.6 to illustrate the effect of repeatedly iterating stages 1 and 2 (Section 5.1). Some robustness and computational load analyses have also been carried out on the same example. Then, a linear parameter-varying (LPV) system identification problem is discussed in Section 5.3, which is not trivial due to the presence of local models with the same structure but different parameterizations. A third, more complex case study is also considered.

All tests have been performed in a MATLAB 2017a environment [25], exploiting the Parallel Computing Toolbox, on an HP ProBook 650 G1 CORE i7-4702MQ CPU @2.20 GHz with 8GB of RAM.

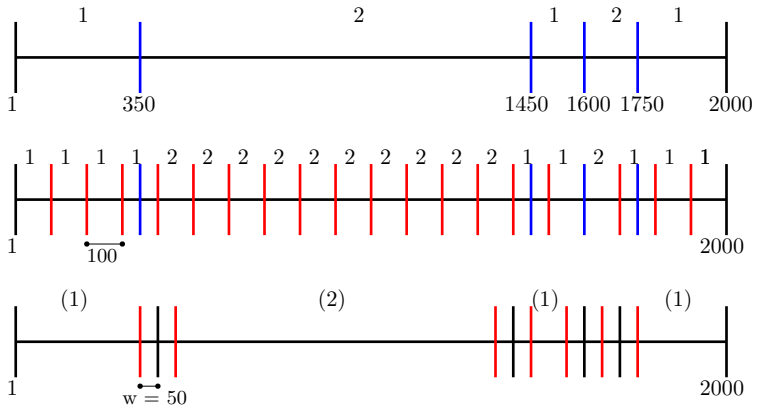

Figure 3: Example 1, refinement stage: real switching signal (top), identified switching instants (middle) and updated set of allowed switchings (bottom). Blue bars indicate actual switchings, black bars the detected switching instants, and red bars the candidate switchings. The mode corresponding to each sub-period is reported on the top of each plot: modes indicated in brackets are those whose MEP will be set to a larger value in that sub-period for the next identification stage.

5.1. Example 1 (contd.): Two-stage procedure

Let us apply the iterative two-stage procedure to the illustrative example discussed in Section 3.6. The design parameters of the identification phase, as well as the initial MEPs, RIPs, and candidate switching locations are set as done previously (20 sub-periods of 100 samples are initially defined). The design parameters for the refinement stage have been set to $\alpha = 0.5$ and $p = 0.7$.

Table 4 presents the aggregated results of 100 Monte Carlo (MC) runs. Notice, first of all, that both local model structures have been estimated correctly 100% of the times. Furthermore, the low accuracy in the selection of the switching sequence selection (see Table 4) is only apparent, the errors in the estimation of the switching time instants being in fact rather small. This can be a appreciated by inspection of Figure 4 (top), which shows the distribution of the detected switchings over the MC runs, indicating that the number and position of the switchings are in fact quite accurately estimated, thanks to the refinement procedure. Figure 4 shows also the aggregated results in terms of classification error rate on the training set (percentage of misclassified samples), and the normalized accuracy criterion

$$FIT = 100\,(1 - \|\hat{\boldsymbol{y}} - \boldsymbol{y}\|_2 / \|\boldsymbol{y} - \bar{y}\mathbb{1}\|_2)\,, \tag{31}$$

where $\boldsymbol{y}$ is the vector containing the target outputs, $\bar{y}$ being the mean value, and $\hat{\boldsymbol{y}}$ is the vector of the outputs predicted using the estimated mode switching signal $\boldsymbol{\sigma}$.

With reference to the same example we also ran a comparative analysis with the non-parametric approach of [20], which extends the method presented in [14] from which the SNARX system used in this example has been taken. In particular, among the four methods proposed in [20] to fix the submodel size and limit the number of optimization variables, we chose the Feature Vector Selection (FVS) method. To describe the two modes we considered a linear kernel and a RBF kernel, respectively, exploiting (as done in [14]) the prior knowledge that one submodel is linear and the other is nonlinear. To produce the results presented in the paper we tested various combinations of the design parameters σ (the STD of the RBF kernels) and C (which governs the trade-off between model complexity and model accuracy), obtaining the best results for $\sigma = 0.1$ and $C = 100$. An MC analysis was carried out and the aggregate results are reported in Figure 5. The values of the FIT criterion are roughly in the same range as with the proposed algorithm, albeit with a much larger variance. However, the more striking difference is in the sample classification accuracy, which is significantly larger than with the proposed algorithm. This is a remarkable aspect, considering also that with the non-parametric approach we have taken advantage of the *a priori* knowledge about the linearity of one of the submodels. One reason for this performance difference lies in the fact that the non-parametric method operates on a sample-by-sample basis, resulting in a very fragmented mode mapping of the time history (unless some sort of post-processing is applied). This does not happen with our method, since it exploits the time-ordering of the collected data to solve the sample-mode mapping process, by applying a segmentation in a relatively small number of subperiods. In the light of the large classification error, the occasional high FIT models obtained with the non-parametric approach

might be interpreted as a manifestation of overfitting behavior. Finally, the considered non-parametric approach on average required 123.9 seconds to solve the identification task.

Table 4: Example 1 (contd.): $\mathcal{T}_s^\circ \nsubseteq \mathcal{T}_s$. MC analysis.

Average elapsed time [s]	284.26
Percentage of correct selection of $\boldsymbol{\kappa}$	61.62%
Average # of explored sequences	11791
Total # of allowed switching sequences	1048576
Percentage of correct selection of $\boldsymbol{s}^{(1)}$	100%
Average # of explored models for mode 1	654
Total # of possible model structures for mode 1	32768
Percentage of correct selection of $\boldsymbol{s}^{(2)}$	100%
Average # of explored model structures for mode 2	758
Total # of possible model structures for mode 2	32768

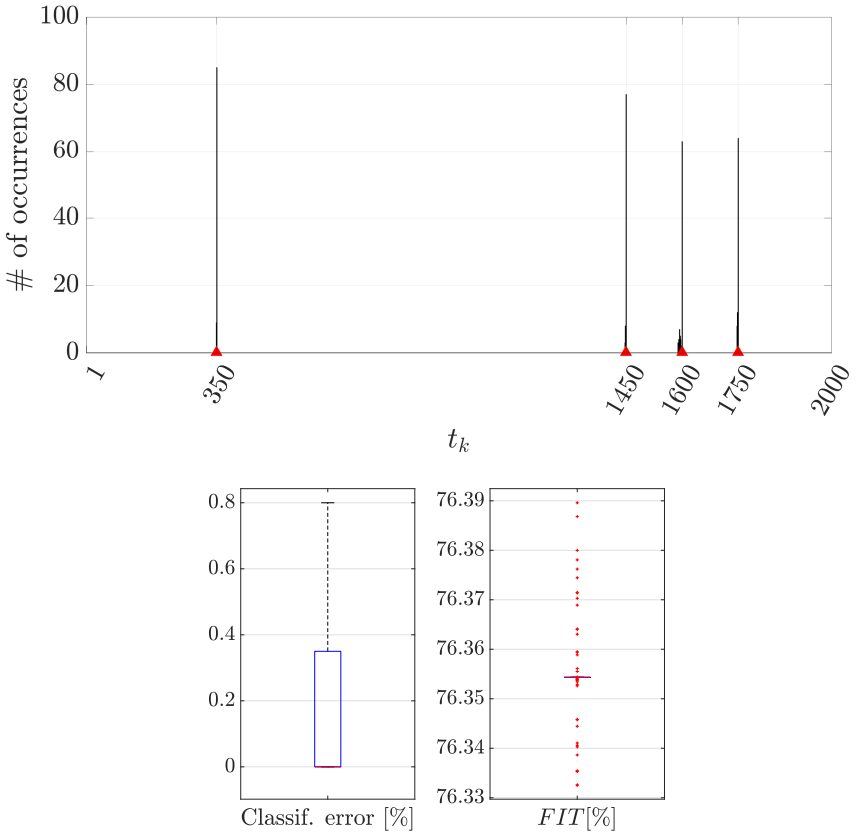

Figure 4: Example 1 (contd.): $\mathcal{T}_s^\circ \nsubseteq \mathcal{T}_s$, proposed method. Top: distribution of the detected switching time instants for the proposed method (red markers represent the true switching instants). Bottom: boxplots showing the distributions of the classification error rate and the FIT criterion on the training set.

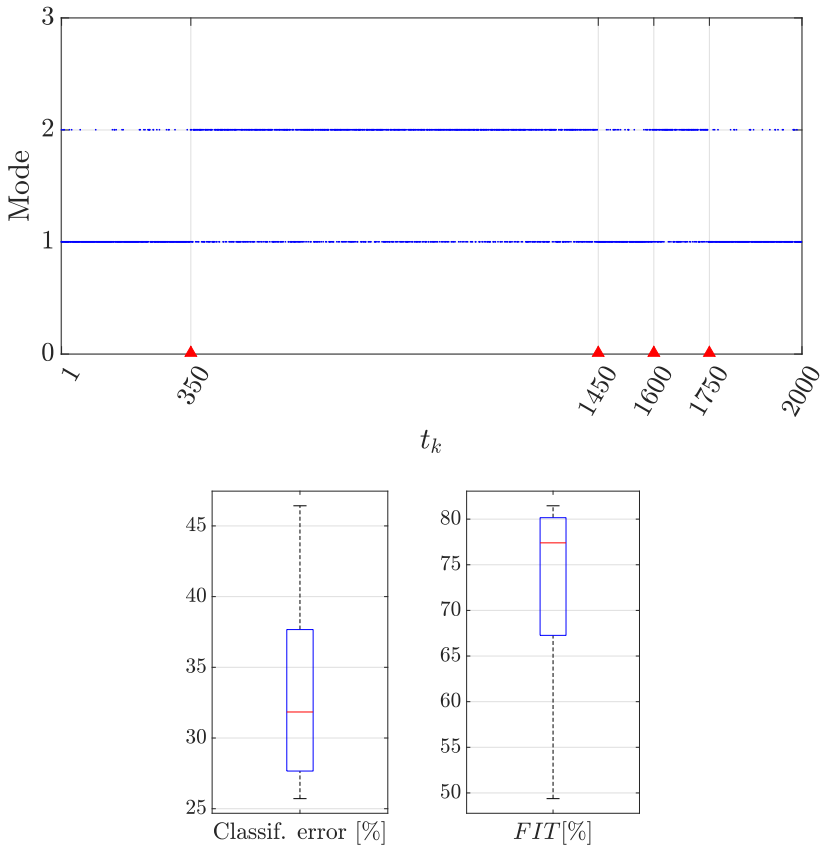

Figure 5: Example 1 (contd.): $\mathcal{T}_s^\circ \nsubseteq \mathcal{T}_s$, non-parametric approach described in [20]. Top: Sample-mode mapping (single run). Bottom: boxplots showing the distributions of the classification error rate and the FIT criterion on the training set.

5.2. Example 1 (contd.): Robustness and computational load analysis

To show the robustness of the proposed method with respect to the initial choice of the switching instants, a MC simulation was carried out on Example 1[5], initializing $\mathcal{T}_s^{(0)}$ randomly. Specifically, at each run the candidate switching instants are set to $t_k = 100k + v_k$, $k = 1, \ldots, 19$, where v_k is a zero mean white gaussian noise with standard deviation 10. As can be noticed from Figure 6, the classification error rate is generally below 1% and in any case lower than 5%, and the obtained distribution of the detected switching instants shows that the algorithm performs reasonably well in the data segmentation task, leading to accurate models. Indeed, the overall accuracy as described by the FIT index is not distant from what found previously.

We also analyzed the robustness of the proposed approach as the noise level increases (using a fixed $\mathcal{T}_s^{(0)}$, with $t_k = 100k$, $k = 1, \ldots, 19$, as done originally). For each data realization (*i.e.* different SNR level), 10 runs were carried out, the aggregated results being summarized in Table 5. As expected, the performance of the method in terms of FIT decreases significantly as the noise variance increases. Interestingly enough, the classification error rate increases very slowly and remains well below 1% in all the examined range.

Finally, a computational load analysis for an increasing number of switchings was carried out, by analyzing datasets of different length obtained from the system of Example 1. Assuming that the system switches between the two modes every 100 instants (starting from $\kappa_1^\circ = 1$), the number of switching instants t_k in $\mathcal{T}_s^\circ$ grows proportionally to N. As done previously, we initialized $\mathcal{T}_s^{(0)}$ randomly. An MC simulation was carried out by running the algorithm

[5]Where the values of the design parameters are not reported explicitly, those used in Section 5.1 are considered.

30 times for each data realization with different random initializations of $\mathcal{T}_s^{(0)}$, and repeating for different N values. Figure 7 shows how the elapsed time varies with the number of switching time instants. As expected, this is the most critical factor which affects the computational burden.

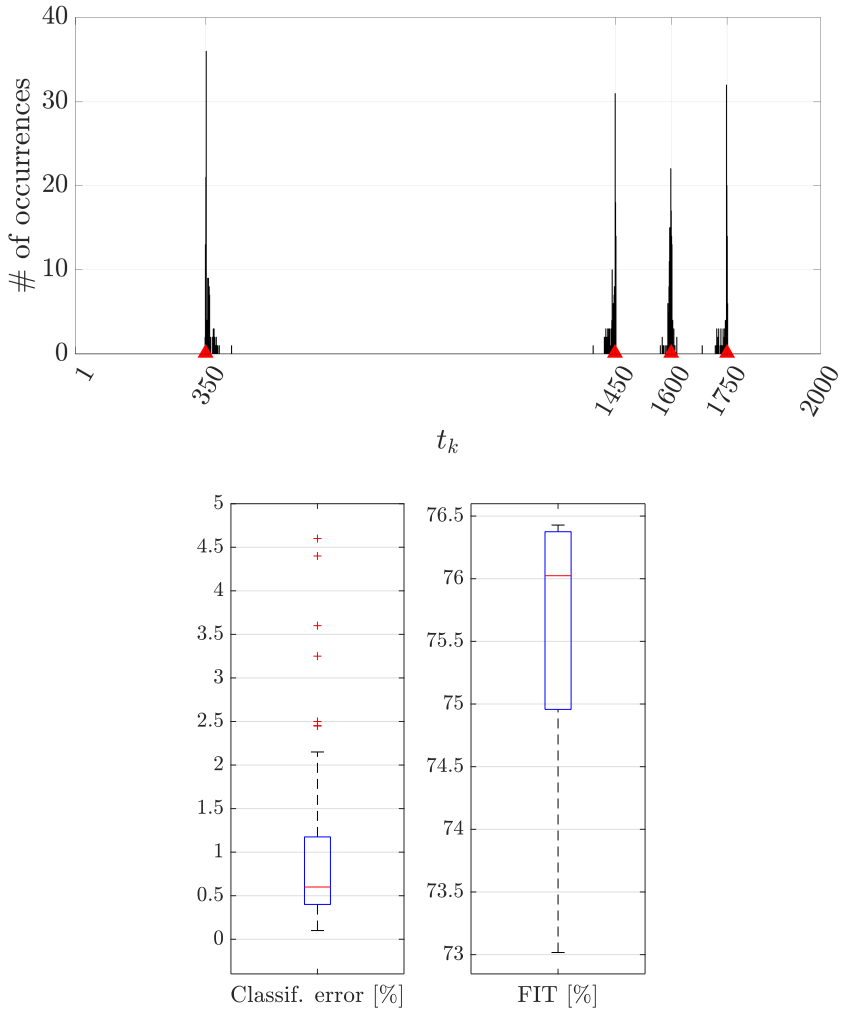

Figure 6: Example 1 (contd.): robustness w.r.t. the initial choice of the switching instants. Distribution of the detected switching time instants (red markers represent the true switching instants), and boxplots demonstrating the classification error rate on the training set and the FIT criterion.

Table 5: Example 1 (contd.): robustness w.r.t. the noise level. MC analysis, mean values and variances.

Noise σ	0.01	0.0422	0.0744	0.1067	0.1389	0.1711	0.2033	0.2356
Train Cl. Err. [%]	0 (0)	0 (0)	0 (0)	0.044 (0.018)	0.23 (0.043)	0.34 (0.042)	0.34 (0.043)	0.42 (0.016)
FIT[%]	96.44 (1.36)	89.85 (0)	82.98 (0)	76.85 (2.97E-5)	71.59 (6.57E-5)	67.19 (2.84E-4)	63.54 (3.29E-4)	60.55 (2.57E-5)

5.3. Example 2: an LPV system

The aim of this example is to assess how the method fares in the identification of the overall process model when the local models have the same structure, as happens *e.g.* for LPV systems. Consider thus the system presented in [12]:

$$y(t) = \vartheta^{(i)} y(t-1) - 0.7y(t-2) + u(t-1) - 0.5u(t-2) + e(t), \tag{32}$$

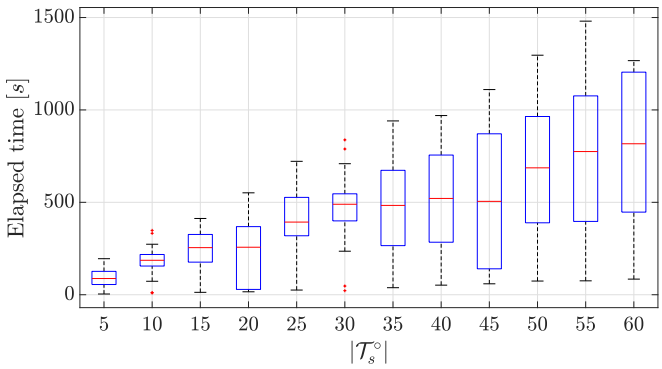

Figure 7: Example 1 (contd.): elapsed time as a function of the number of switchings.

which consists of $N_M = 4$ local models, that are almost identical apart from one parameter that takes the values $\vartheta^{(1)} = -1.5$, $\vartheta^{(2)} = -1$, $\vartheta^{(3)} = -0.5$, and $\vartheta^{(4)} = 0.5$, respectively. The input signal $u(t)$ is a ± 1 Pseudo-Random Binary Sequence (PRBS), while the noise is an i.i.d. Gaussian process, $e(t) \sim \mathcal{N}(0, \sigma^2)$, with $\sigma = 0.5$. A data-set of 2500 input-output samples is available during which 6 mode switchings occur, according to $\mathcal{T}_s^\circ = \{400,\ 810,\ 1270,\ 1500,\ 1830,\ 2150\}$ and following the mode sequence $\boldsymbol{\kappa}^\circ = [1,\ 2,\ 3,\ 2,\ 3,\ 4,\ 1]$.

We compare our method with the SON-EM method described in [12], which turned out to fare well w.r.t. some of the latest developments in identification for linear switched systems (for details see [12]). Among others, the SON-EM outperforms (on the considered examples) the RANdom SAmple Consensus (RANSAC) method [10] which has been adapted in [12] for hybrid systems. In order to have a fair comparison, we here assume that the model structure of the modes is fixed as for the SON-EM method (the correct regressors $y(t-1)$, $y(t-2)$, $u(t-1)$, and $u(t-2)$, are employed and the NARX model structure selection part is skipped). Both methods address the estimation of all 4 parameters (not just $\vartheta^{(i)}$), for each mode. The initial set of candidate switching locations is defined as $\mathcal{T}_s^{(0)} = \{100, 200, \ldots, 2400\}$, inducing a uniform subdivision of the data-set in 25 sub-periods of 100 samples. The design parameters for the refinement stage are set to $\alpha = 0.5$ and $p = 0.25$.

An MC analysis has been carried out, running the algorithm 100 times on the same data realization. It turned out that 92% of the detected switching sequences contained the correct number of time instants. The distribution of the detected switching time instants for these sequences is reported in Figure 8. These results show that the proposed method performs well in detecting the switchings, in fact the best run yields $\mathcal{T}_s^\star = \{400,\ 797,\ 1250,\ 1500,\ 1830,\ 2146\}$ which proves to be quite close to the real one $\mathcal{T}_s^\circ$. Overall, the maximum and the mean sample classification error are respectively 5.96% and 1.54% for the MC runs resulting in a $\mathcal{T}_s^\circ$ with the correct cardinality. In the remaining 8% of detected switching sequences, 6 of them missed only the switching at time $t = 400$, while the other 2 cases resulted in a completely wrong $\mathcal{T}_s^\star$.

Figure 9 compares the estimates of $\vartheta^{(i)}$ on a single run obtained with the proposed method and the SON-EM method [12]. It is noteworthy that both methods captured well all the switching time instants and provided good parameter estimates, thus showing that the proposed method equals in terms of performance one of the most recent and promising methods. For the considered run, Table 6 reports the performance of the identified hybrid model at each iteration, the detected switchings, the sample-mode classification for each sub-period and the corresponding classification error on the training set. From a computational complexity viewpoint, we compared the two methods in terms of the time required to solve the identification task. It turned out that our method is more demanding w.r.t. the SON-EM, *i.e.,* on average our method lasted 238.56 seconds against 20.6063.

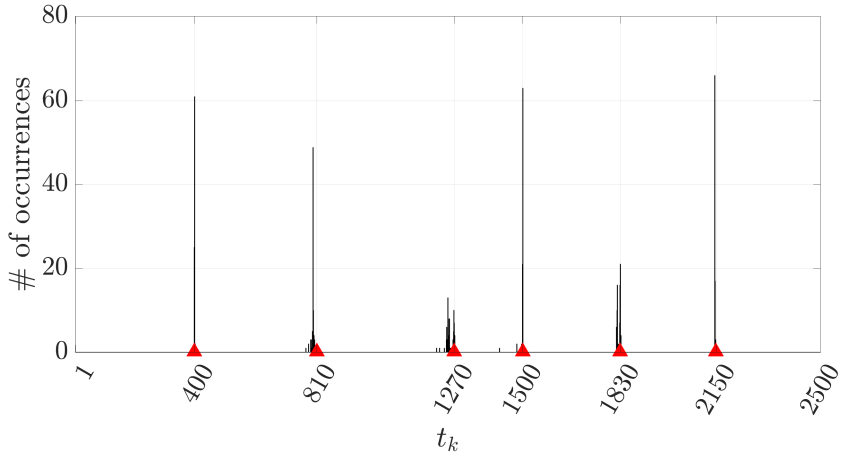

Figure 8: Example 2 - LPV system: Distribution of the detected switching time instants (red markers represent the true switching instants).

Table 6: Example 2 - LPV system: Performance of a single run over iterations.

r	$\mathcal{L}^{(1)}$	$\mathcal{L}^{(2)}$	$\mathcal{L}^{(3)}$	$\mathcal{L}^{(4)}$	$\mathcal{L}$	t_1	t_2	t_3	t_4	t_5	t_6	t_7	$\boldsymbol{\kappa}$	Classif. error
1	0.2467	0.9951	0.2838	0.4848	0.4569	400	800	1300	1500	1800	2100	2200	[1,2,3,2,3,4,2,1]	6.8%
2	0.2469	0.2603	0.3217	0.2459	0.2742	400	800	1250	1500	1850	2150	–	[1,2,3,2,3,4,1]	1.4%
3	0.2469	0.2275	0.2615	0.2934	0.2527	400	800	1275	1500	1825	2150	–	[1,2,3,2,3,4,1]	0.8%
4	0.2469	0.2275	0.2615	0.2934	0.2527	400	800	1275	1500	1825	2150	–	[1,2,3,2,3,4,1]	0.8%
5	0.2469	0.2315	0.2542	0.2934	0.2513	400	813	1269	1500	1825	2150	–	[1,2,3,2,3,4,1]	0.36%
6	0.2529	0.2315	0.2542	0.2951	0.2533	400	813	1269	1500	1826	2147	–	[1,2,3,2,3,4,1]	0.48%
7	0.2829	0.2317	0.2570	0.2735	0.2514	400	810	1269	1500	1827	2147	–	[1,2,3,2,3,4,1]	0.28%

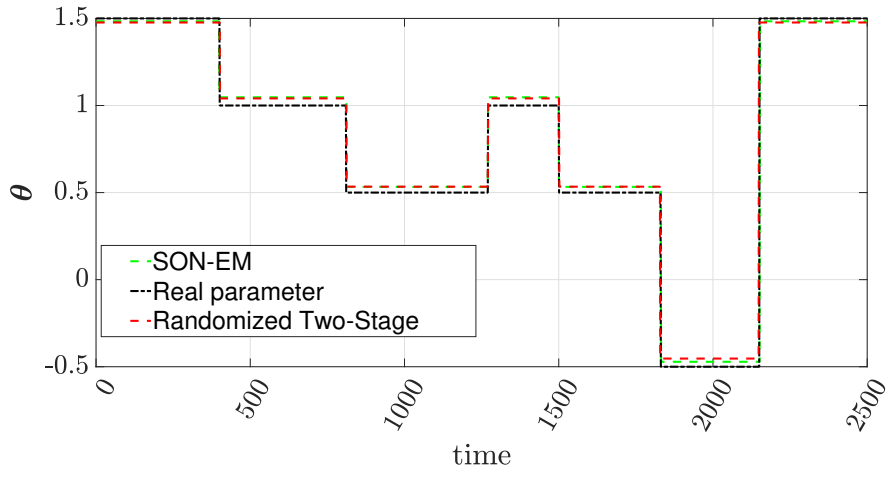

Figure 9: Example 2 - LPV system: Identification of parameter $\vartheta^{(i)}$ with the proposed method and the SON-EM method.

5.4. *Example 3: A 3-mode SNARX case, with nonlinear modes*

In this study, the following system has been considered:

$$\begin{aligned}
\text{mode 1 }: y(t) &= 0.5y(t-1) + 0.8u(t-2) \\
&\quad + u(t-1)^2 - 0.3y(t-2)^2 + e(t) \\
\text{mode 2 }: y(t) &= 0.2y(t-1)^3 - 0.5y(t-2) \\
&\quad - 0.7y(t-2)u(t-2)^2 + 0.6u(t-2)^2 + e(t) \\
\text{mode 3 }: y(t) &= 0.4y(t-1)^3 + 0.5y(t-2) \\
&\quad - 0.7y(t-2)u(t-2)^2 + 0.6u(t-2)^2 + e(t)
\end{aligned}$$

where $e(t)$ is a zero mean Gaussian noise of variance 0.01 and $u(t)$ is uniformly distributed in the interval $[-1, 1]$. Notice that two of the three nonlinear local models have the same model structure (but one different parameter). An observation window of $N = 3400$ samples has been collected, which contains 5 switchings, at locations $\mathcal{T}_s^\circ = \{500, 1030, 2115, 2740, 3000\}$, and corresponding to the mode switching sequence $\kappa^\circ = [1, 2, 1, 3, 2, 3]$.

An MC analysis has been carried out considering an initial set of candidate switchings defined as $\mathcal{T}_s^{(0)} = \{200, 400, \ldots, 3200\}$, which induces 17 sub-periods of 200 samples. Furthermore, the initial MEPs are all set to 0.33, and the initial RIPs to $\frac{1}{n} = 0.0061$. Regarding the NARX model structure selection, the candidate regressor set is defined by $n_d = 3$, $n_y = n_u = 4$, which makes it abundantly oversized (the model orders are overestimated), amounting to $n = 165$ regressors. Finally, $\alpha = 0.5$ and $p = 0.7$, for the refinement stage.

Table 7 reports the aggregated results of 50 MC runs. Apparently, the model structures of all the modes have been detected with a quite high accuracy (over 94%), despite the large combinatorial complexity of the involved model selection problems. Furthermore, Figure 10 illustrates the robustness of the algorithm in estimating the switching locations. Indeed, in the best case, a $\mathcal{T}_s^\star = \{499, 1029, 2112, 2739, 2998\}$ was obtained, whereas an error of only 1.6% was obtained regarding the sample classification in the worst run of the MC study.

Table 8 reports the results of a single run, indicating specifically the performance of the identified hybrid model at each iteration, the detected switchings, the sample-mode classification (for each sub-period) and the corresponding percentage error. Furthermore, Table 9 reports for each mode the percentage of misclassified samples. As can be noticed, the first identification stage results in an inaccurate model mainly because of the initial coarse uniform placement of the switching candidate time instants, which leads to a large sample classification error mainly for the first and third mode (see $r = 1$ in Table 9). The algorithm adapts the structure selection by extracting the correct terms plus some extra ones in order to take into account for the misclassified samples (see $r = 1$ in Table 10). Notwithstanding this, the first identified switching signal $\boldsymbol{\sigma}$ is already close to the real discrete dynamics. The subsequent refinement stages (and the identification phases) progressively improve both the local and the global performance leading to a very accurate final hybrid model (both in terms of the continuous and the discrete dynamics). It is apparent that as the switching signal is more accurately estimated, the accuracy of the local models also improves, since they are estimated on more appropriate data sets. Indeed, from the fourth iteration on the sample classification errors are lower than 1% for all modes (see Table 9) and the extracted structures are correct (see Table 10).

Table 7: Example 3: MC analysis.

Average elapsed time [s]	1247
Percentage of $\boldsymbol{\kappa}$ of correct length	100%
Average # of explored sequences	5042
Total # of allowed switching sequences	131072
Percentage of correct selection of $\boldsymbol{s}^{(1)}$	94%
Average # of explored models for mode 1	6634.3
Total # of possible model structures for mode 1	$4.6768 \cdot 10^{49}$
Percentage of correct selection of $\boldsymbol{s}^{(2)}$	96%
Average # of explored model structures for mode 2	6377.2
Total # of possible model structures for mode 2	$4.6768 \cdot 10^{49}$
Percentage of correct selection of $\boldsymbol{s}^{(3)}$	94%
Average # of explored model structures for mode 3	6144.4
Total # of possible model structures for mode 3	$4.6768 \cdot 10^{49}$

6. Conclusions

We consider the identification of switched nonlinear autoregressive exogenous (SNARX) models, and propose an iterative method that addresses the challenge of the simultaneous identification of the mode switching sequence and of the NARX model associated to each mode. The proposed method alleviates the combinatorial complexity of the problem by adopting a two-stage approach. More precisely, in the first stage, candidate mode switching instants are fixed and adopted to segment the input/output data and jointly solve mode assignment and NARX structure and parameter

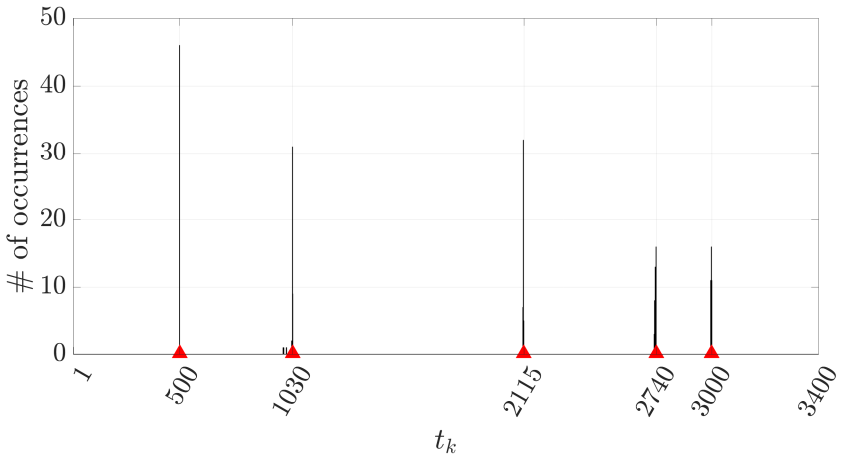

Figure 10: Example 3: Distribution of the detected switching time instants (red markers represent the true switching instants).

Table 8: Example 3: Performance over iterations on a single run.

r	$\mathcal{L}^{(1)}$	$\mathcal{L}^{(2)}$	$\mathcal{L}^{(3)}$	$\mathcal{L}$	t_1	t_2	t_3	t_4	t_5	κ	Classif. error
1	0.0395	0.0097	0.0117	0.0257	600	1000	2200	2800	3000	[1,2,1,3,2,3]	7.86%
2	0.0274	0.0097	0.0106	0.0190	550	1000	2150	2750	3000	[1,2,1,3,2,3]	3.57%
3	0.0179	0.0097	0.0107	0.0138	525	1025	2125	2750	3000	[1,2,1,3,2,3]	2.86%
4	0.0135	0.0104	0.0097	0.0116	512	1025	2112	2737	3000	[1,2,1,3,2,3]	0.66%
5	0.0121	0.0105	0.0097	0.0110	505	1032	2112	2737	3000	[1,2,1,3,2,3]	0.37%
6	0.0110	0.0104	0.0097	0.0104	501	1028	2112	2737	3000	[1,2,1,3,2,3]	0.26%
7	0.0098	0.0106	0.0096	0.0098	499	1030	2114	2739	3000	[1,2,1,3,2,3]	0.09%

Table 9: Example 3: Sample classification error over iterations on a single run.

r	Mode 1	Mode 2	Mode 3
1	11.94%	0%	5.45%
2	8.85%	0%	0.91%
3	2.42%	0%	0.89%
4	1.06%	0%	0.89%
5	0.32%	0.63%	0.27%
6	0.19%	0.38%	0.27%
7	0%	0.25%	0.09%

identification; in the second stage, the candidate mode switching instants are refined. As for the combinatorial optimization problem in the first stage, it is addressed using a computationally attractive randomized method where mode assignment and SNARX model structure are modeled through discrete probability distributions that are progressively tuned via a sample-and-evaluate strategy, until convergence to a limit distribution concentrated on the best SNARX model of the system generating the observed data. Numerical examples show the efficacy of the proposed method.

References

[1] Laurent Bako. Identification of switched linear systems via sparse optimization. *Automatica*, 47(4):668–677, 2011.

[2] Laurent Bako, Khaled Boukharouba, and Stéphane Lecoeuche. An l_0–l_1 norm based optimization procedure for the identification of switched nonlinear systems. In 49^{th} *IEEE Conference on Decision and Control*, pages 4467–4472, 2010.

[3] Alberto Bemporad, Andrea Garulli, Simone Paoletti, and Antonio Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, 2005.

Table 10: Example 3: Model structure selection over iterations (true regressors in bold face).

r	Regressors of mode 1	Regressors of mode 2	Regressors of mode 3
1	$\boldsymbol{y(t-1)}$, $\boldsymbol{u(t-2)}$, $y(t-1)u(t-2)$, $\boldsymbol{y(t-2)^2}$, $y(t-3)u(t-2)$, $\boldsymbol{u(t-1)^2}$, $u(t-2)^2$, $u(t-2)u(t-3)$, $u(t-2)u(t-4)$, $y(t-2)u(t-2)u(t-4)$, $y(t-2)u(t-4)^2$, $y(t-3)^2u(t-2)$, $y(t-4)^2u(t-2)$, $u(t-2)^3$, $u(t-2)u(t-3)^2$	$\boldsymbol{y(t-2)}$, $\boldsymbol{u(t-2)^2}$, $\boldsymbol{y(t-1)^3}$, $\boldsymbol{y(t-2)u(t-2)^2}$	$\boldsymbol{y(t-2)}$, $y(t-2)y(t-4)$, $\boldsymbol{u(t-2)^2}$, $\boldsymbol{y(t-1)^3}$, $y(t-2)y(t-4)^2$, $\boldsymbol{y(t-2)u(t-2)^2}$, $u(t-4)^2$
2	$\boldsymbol{y(t-1)}$, $\boldsymbol{u(t-2)}$, $y(t-1)u(t-2)$, $\boldsymbol{y(t-2)^2}$, $y(t-3)u(t-2)$, $\boldsymbol{u(t-1)^2}$, $u(t-2)u(t-3)$, $u(t-2)u(t-4)$, $y(t-2)u(t-4)^2$, $y(t-3)^2u(t-2)$, $y(t-4)^2u(t-2)$, $u(t-2)^3$, $u(t-2)u(t-3)^2$	$\boldsymbol{y(t-2)}$, $\boldsymbol{u(t-2)^2}$, $\boldsymbol{y(t-1)^3}$, $\boldsymbol{y(t-2)u(t-2)^2}$	$\boldsymbol{y(t-2)}$, $\boldsymbol{u(t-2)^2}$, $\boldsymbol{y(t-1)^3}$, $\boldsymbol{y(t-2)u(t-2)^2}$
$3\cdots7$	$\boldsymbol{y(t-1)}$, $\boldsymbol{u(t-2)}$, $\boldsymbol{y(t-2)^2}$, $\boldsymbol{u(t-1)^2}$	$\boldsymbol{y(t-2)}$, $\boldsymbol{u(t-2)^2}$, $\boldsymbol{y(t-1)^3}$, $\boldsymbol{y(t-2)u(t-2)^2}$	$\boldsymbol{y(t-2)}$, $\boldsymbol{u(t-2)^2}$, $\boldsymbol{y(t-1)^3}$, $\boldsymbol{y(t-2)u(t-2)^2}$

[4] Federico Bianchi, Maria Prandini, and Luigi Piroddi. A randomized approach to switched nonlinear systems identification. *18th IFAC Symposium on System Identification, SYSID 2018*, 2018.

[5] S. A. Billings. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Wiley, 2013.

[6] Tillmann Falck, Henrik Ohlsson, Lennart Ljung, Johan AK Suykens, and Bart De Moor. Segmentation of time series from nonlinear dynamical systems. *IFAC Proceedings Volumes*, 44(1):13209–13214, 2011.

[7] A. Falsone, L. Piroddi, and M. Prandini. A randomized algorithm for nonlinear model structure selection. *Automatica*, 60:227–238, 2015.

[8] Alessandro Falsone, Luigi Piroddi, and Maria Prandini. A novel randomized approach to nonlinear system identification. In 53^{rd} *IEEE Conference on Decision and Control*, pages 6516–6521, 2014.

[9] Giancarlo Ferrari-Trecate, Marco Muselli, Diego Liberati, and Manfred Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.

[10] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[11] Andrea Garulli, Simone Paoletti, and Antonio Vicino. A survey on switched and piecewise affine system identification. In 16^{th} *IFAC Symposium on System Identification*, pages 344–355, Brussels, Belgium, July 11-13 2012.

[12] András Hartmann, João M Lemos, Rafael S Costa, João Xavier, and Susana Vinga. Identification of switched ARX models via convex optimization and expectation maximization. *Journal of Process Control*, 28:9–16, 2015.

[13] Aleksandar Lj Juloski, Siep Weiland, and WPMH Heemels. A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10):1520–1533, 2005.

[14] Fabien Lauer and Gérard Bloch. Switched and piecewise nonlinear hybrid system identification. In *International Workshop on Hybrid Systems: Computation and Control*, pages 330–343, 2008.

[15] Fabien Lauer and Gérard Bloch. Piecewise smooth system identification in reproducing kernel hilbert space. In 53^{rd} *IEEE Conference on Decision and Control*, pages 6498–6503, 2014.

[16] Fabien Lauer and Gérard Bloch. *Hybrid System Identification*. Springer International Publishing, 2019.

[17] Fabien Lauer, Gérard Bloch, and René Vidal. Nonlinear hybrid system identification with kernel models. In 49^{th} *IEEE Conference on Decision and Control*, pages 696–701, 2010.

[18] Fabien Lauer, Gérard Bloch, and René Vidal. A continuous optimization framework for hybrid system identification. *Automatica*, 47(3): 608–613, 2011.

[19] Van Luong Le, Gérard Bloch, and Fabien Lauer. Reduced-size kernel models for nonlinear hybrid system identification. *IEEE Transactions on Neural Networks*, 22(12):2398–2405, 2011.

[20] Van Luong Le, Fabien Lauer, Laurent Bako, and Gérard Bloch. Learning nonlinear hybrid systems: from sparse optimization to support vector regression. In *Proceedings of the* 16^{th} *International Conference on Hybrid systems: computation and control*, pages 33–42, 2013.

[21] I.J. Leontaritis and S.A. Billings. Input-output parametric models for non-linear systems part I: deterministic non-linear systems. *International Journal of Control*, 41(2):303–328, 1985.

[22] I.J. Leontaritis and S.A. Billings. Input-output parametric models for non-linear systems part II: stochastic non-linear systems. *International Journal of Control*, 41(2):329–344, 1985.

[23] Yi Ma and René Vidal. Identification of deterministic switched ARX systems via identification of algebraic varieties. In *International Workshop on Hybrid Systems: Computation and Control*, pages 449–465, 2005.

[24] Ichiro Maruta, Toshiharu Sugie, and Tae-Hyoung Kim. Identification of multiple mode models via distributed particle swarm optimization. In *Proceedings of the* 18^{th} *IFAC World Congress*, pages 7743–7748, Milano, Italy, Aug. 28 – Sept. 2 2011.

[25] MATLAB. *Version 2017b*. The MathWorks Inc., Natick (MA), USA, 2017.

[26] Sohail Nazari, Qing Zhao, and Biao Huang. An improved algebraic geometric solution to the identification of switched ARX models with noise. In *Proceedings of the American Control Conference*, pages 1230–1235, 2011.

[27] Henrik Ohlsson and Lennart Ljung. Identification of switched linear regression models using sum-of-norms regularization. *Automatica*, 49 (4):1045–1050, 2013.

[28] Necmiye Ozay, Mario Sznaier, Constantino M Lagoa, and Octavia I Camps. A sparsification approach to set membership identification of switched affine systems. *IEEE Transactions on Automatic Control*, 57(3):634–648, 2012.

[29] Simone Paoletti, Aleksandar Lj Juloski, Giancarlo Ferrari-Trecate, and René Vidal. Identification of hybrid systems a tutorial. *European*

Journal of Control, 13(2–3):242–260, 2007.
[30] Gianluigi Pillonetto. A new kernel-based approach to hybrid system identification. *Automatica*, 70:21–31, 2016.
[31] Jacob Roll, Alberto Bemporad, and Lennart Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, 2004.
[32] J. Speyer, J. Deyst, and D. Jacobson. Optimization of stochastic linear systems with additive measurement and process noise using exponential performance criteria. *IEEE Transactions on Automatic Control*, 19(4):358–366, 1974.

Appendix A. Theorem proofs

Appendix A.1. Proof of Theorem 3.1

The proof goes along the lines of that reported in Appendix A.1 in [7], where the special case of categorical distributions with only two outcomes (Bernoullian distributions) is discussed. The proof is here reported for the sake of clarity within the notation introduced in this paper.

Consider first the case $x_j^\star = i$. Then, the index $\delta_j^{(i)}$ (11) can be lower bounded as follows:

$$\delta_j^{(i)} \geq \mathcal{J}(\boldsymbol{x}^\star)\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}^\star) - \bar{\mathcal{J}}_j^{(i)}, \tag{A.1}$$

where $\bar{\mathcal{J}}_j^{(i)} = \max_{\boldsymbol{x}\in\mathcal{X}:x_j\neq i}\mathcal{J}(\boldsymbol{x})$. Indeed, for the first term in the RHS of (11),

$$\mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma})|\gamma_j = i] = \sum_{\boldsymbol{x}\in\mathcal{X}:x_j=i}\mathcal{J}(\boldsymbol{x})\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}) \geq \mathcal{J}(\boldsymbol{x}^\star)\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}^\star), \tag{A.2}$$

where the inequality follows upon observing that $\boldsymbol{x}^\star \in \{\boldsymbol{x}\in\mathcal{X} : x_j = i\}$ and that $\mathcal{J}(\boldsymbol{x}) \geq 0$.
On the other hand, the second term in the RHS of (11),

$$\mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma})|\gamma_j \neq i] \leq \bar{\mathcal{J}}_j^{(i)}, \tag{A.3}$$

by definition. Therefore, applying the bounds A.2 and A.3 in (11), one obtains (A.1).

A similar reasoning applies for the case $x_j^\star \neq i$, leading to the following bound:

$$\delta_j^{(i)} \leq \tilde{\mathcal{J}}_j^{(i)} - \mathcal{J}(\boldsymbol{x}^\star)\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}^\star), \tag{A.4}$$

where $\tilde{\mathcal{J}}_j^{(i)} = \max_{\boldsymbol{x}\in\mathcal{X}:x_j=i}\mathcal{J}(\boldsymbol{x})$. Indeed, for the first term in the RHS of (11),

$$\mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma})|\gamma_j = i] \leq \tilde{\mathcal{J}}_j^{(i)}, \tag{A.5}$$

by definition.
The second term can be bounded as

$$\mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma})|\gamma_j \neq i] = \sum_{\boldsymbol{x}\in\mathcal{X}:x_j\neq i}\mathcal{J}(\boldsymbol{x})\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}) \geq \mathcal{J}(\boldsymbol{x}^\star)\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}^\star). \tag{A.6}$$

Therefore, applying the bounds A.5 and A.6 in (11), one obtains (A.4).

Now, under the assumption that $\boldsymbol{x}^\star$ is unique, if one sets

$$\varrho > \max_{\boldsymbol{x}\in\mathcal{X}\setminus\{\boldsymbol{x}^\star\}}\frac{\mathcal{J}(\boldsymbol{x})}{\mathcal{J}(\boldsymbol{x}^\star)}$$

and $\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{x}^\star) \geq \varrho$, one obtains that $\delta_j^{(i)} > 0$ if $x_j^\star = i$, from bound A.1. On the other side, $\delta_j^{(i)} < 0$ if $x_j^\star \neq i$, from bound A.4.

Appendix A.2. Proof of Theorem 3.2

Let $\mathbb{P}_{\boldsymbol{\gamma}}^{(k)}$ be the probability distribution associated with the probability matrix $\boldsymbol{\pi}$ at iteration k. Assuming that $\mathbb{P}_{\boldsymbol{\gamma}}^{(k)}(\boldsymbol{x}^{\star}) \geq \varrho$, where ϱ makes the condition of Theorem (3.1) valid. Then one obtains that:

$$\begin{cases} \delta_j^{(i)} > 0 & \forall j : x_j^{\star} = i \\ \delta_j^{(i)} < 0 & \forall j : x_j^{\star} \neq i \end{cases}$$

and therefore, according to (12) and recalling that $\chi > 0$:

$$\begin{cases} \pi_j^{(i)}(k+1) = \pi_j^{(i)}(k) + \chi\delta_j^{(i)} > \pi_j^{(i)}(k) & \forall j : x_j^{\star} = i \\ \pi_j^{(i)}(k+1) = \pi_j^{(i)}(k) + \chi\delta_j^{(i)} < \pi_j^{(i)}(k) & \forall j : x_j^{\star} \neq i \end{cases}$$

Recalling that:

$$\mathbb{P}_{\boldsymbol{\gamma}}^{(k)}(\boldsymbol{x}) = \prod_{j=1}^{n}\prod_{i=1}^{m}\left(\pi_j^{(i)}(k)\right)^{\beta_j^{(i)}}$$

it follows that

$$\mathbb{P}_{\boldsymbol{\gamma}}^{(k+1)}(\boldsymbol{x}^{\star}) > \mathbb{P}_{\boldsymbol{\gamma}}^{(k)}(\boldsymbol{x}^{\star}) > \varrho.$$

We then have a sequence of strictly monotonically increasing scalars that are upper bounded by 1, which entails that $\lim_{k\to\infty}\mathbb{P}_{\boldsymbol{\gamma}}^{(k)}(\boldsymbol{x}^{\star}) = 1$.

Appendix B. Digression on local convergence of the heuristic implementation

The absence of local convergence is showed upon observing that the sign of $\tilde{\ell}_{j,i}^{(l)}$ in (28) could be different from that of $\ell_{j,i}^{(l)}$ (22). To show that, consider first the relation between $\mathcal{J}(\lambda)$ and $\mathcal{J}^{(i)}\left(\boldsymbol{\sigma}, \boldsymbol{s}^{(i)}\right)$, $i = 1, \ldots, N_M$:

$$\mathcal{J}(\lambda) = e^{-K_\lambda \mathcal{L}(\lambda)} = e^{-K_\lambda \cdot [\frac{1}{N}\sum_{i:\, N_i \neq 0} N_i \cdot \mathcal{L}^{(i)}(\boldsymbol{\kappa}, \boldsymbol{s}^{(i)})]} = \prod_{i=1}^{N_M}\left[\mathcal{J}^{(i)}\left(\boldsymbol{\kappa}, \boldsymbol{s}^{(i)}\right)\right]^{\frac{N_i}{N}\frac{K_\lambda}{K_i}} \tag{B.1}$$

Based on (B.1) and under the assumption of independence between modes, and between regressors, one can reformulate the first term in the RHS of (22) as:

$$\mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma}) | \rho_{j,i} = l] = \sum_{\kappa}\mathbb{P}_{\boldsymbol{\xi}}(\kappa)\left[\mathbb{E}\left[\left(\mathcal{J}^{(i)}\left(\kappa, \boldsymbol{s}^{(i)}\right)\right)^{\frac{N_i}{N}\frac{K_\lambda}{K_i}} | \rho_{j,i} = l, \boldsymbol{\xi} = \kappa\right] \cdot \prod_{n\neq i}^{N_M}\mathbb{E}\left[\left(\mathcal{J}^{(i)}\left(\kappa, \boldsymbol{s}^{(i)}\right)\right)^{\frac{N_i}{N}\frac{K_\lambda}{K_i}} | \boldsymbol{\xi} = \kappa\right]\right]. \tag{B.2}$$

Similarly,

$$\mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma}) | \rho_{j,i} \neq l] = \sum_{\kappa}\mathbb{P}_{\boldsymbol{\xi}}(\kappa)\left[\mathbb{E}\left[\left(\mathcal{J}^{(i)}\left(\kappa, \boldsymbol{s}^{(i)}\right)\right)^{\frac{N_i}{N}\frac{K_\lambda}{K_i}} | \rho_{j,i} \neq l, \boldsymbol{\xi} = \kappa\right] \cdot \prod_{n\neq i}^{N_M}\mathbb{E}\left[\left(\mathcal{J}^{(i)}\left(\kappa, \boldsymbol{s}^{(i)}\right)\right)^{\frac{N_i}{N}\frac{K_\lambda}{K_i}} | \boldsymbol{\xi} = \kappa\right]\right]. \tag{B.3}$$

By substituting (B.2) and (B.3) in (22), one obtains:

$$\begin{aligned} &\mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma}) | \rho_{j,i} = l] - \mathbb{E}_{\mathbb{P}_\gamma}[\mathcal{J}(\boldsymbol{\gamma}) | \rho_{j,i} \neq l] = \\ &= \sum_{\kappa}\mathbb{P}_{\boldsymbol{\xi}}(\kappa)\left[\mathbb{E}\left[\left(\mathcal{J}^{(i)}\left(\kappa, \boldsymbol{s}^{(i)}\right)\right)^{\frac{N_i}{N}\frac{K_\lambda}{K_i}} | \rho_{j,i} = l, \boldsymbol{\xi} = \kappa\right] - \mathbb{E}\left[\left(\mathcal{J}^{(i)}\left(\kappa, \boldsymbol{s}^{(i)}\right)\right)^{\frac{N_i}{N}\frac{K_\lambda}{K_i}} | \rho_{j,i} \neq l, \boldsymbol{\xi} = \kappa\right]\right] \cdot \prod_{n\neq i}^{N_M}\mathbb{E}\left[\left(\mathcal{J}^{(i)}\left(\kappa, \boldsymbol{s}^{(i)}\right)\right)^{\frac{N_i}{N}\frac{K_\lambda}{K_i}} | \boldsymbol{\xi} = \kappa\right] \end{aligned} \tag{B.4}$$

That is, the sign of each single $\tilde{\ell}_{j,i}^{(l)}$ (the inner part of the summation over κ) may be different from that of $\ell_{j,i}^{(l)}$.