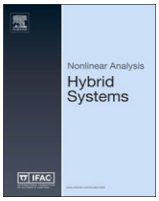
Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

journal homepage: www.elsevier.com/locate/nahs



A randomized method for the identification of switched NARX systems

Miao Yu [*], Federico Bianchi, Luigi Piroddi

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133, Milano, Italy

ARTICLE INFO

Article history:
Received 22 September 2022
Received in revised form 1 February 2023
Accepted 7 April 2023
Available online 6 May 2023

Keywords:
Nonlinear ARX systems
Switched systems
Structure selection
Randomized algorithms
System identification

ABSTRACT

The identification of switched systems is a complex optimization problem that involves both continuous (parametrizations of the local models, a.k.a. modes) and discrete variables (model structures, switching signal). In particular, the combinatorial complexity associated with the estimation of the switching signal grows exponentially with the number of samples, which makes data segmentation (*i.e.* estimating the number and location of mode switchings, and the mode sequence) a challenging problem. In this work, we extend a previously developed randomized approach for the identification of switched systems to encompass the estimation of the switching locations. The method operates by extracting samples from a probability distribution of switched models, and gathering information from the associated model performances to update the distribution, until convergence to a limit distribution associated to a specific model. A suitable probability distribution is employed to represent the likelihood of a mode switching at a certain time, and the update process is designed to correct the switching locations and remove redundant switchings. The proposed algorithm has been compared to existing state-of-the-art methods and has been tested on various benchmark examples, to demonstrate its effectiveness.

1. Introduction

In many modeling problems, the heterogeneity of the system behavior is hardly captured by a single model and, rather, different dynamics are observed, the system switching from one operational mode to another. Examples range from computer vision [1] to DC motors [2], from diesel engines [3] to pick-and-place machines [4], just to mention a few. To model such complex dynamic behaviors one can resort to a switched system, which is characterized by a set of local models that capture the individual modes of operation, and a switching mechanism governing the transition from one mode to another. Mode switching is sometimes associated to the crossing of the boundaries between the operating regions of the local models (piecewise affine (PWA) models) or is otherwise represented as an exogenous switching signal (switched systems).

The identification of a (discrete-time) switched system from data is a particularly challenging problem, that involves the optimization of both continuous (parameterizations of the local models) and discrete variables (model structures, switching signal). Various methodologies have been proposed in the literature to address this problem (see *e.g.* [5,6] for a survey). The major source of complexity is associated with the data segmentation task, *i.e.* estimating the number and location of mode switchings, as well as the mode transition sequence. Indeed, the combinatorial complexity associated

[*] Corresponding author.
E-mail addresses: miao.yu@polimi.it (M. Yu), federico.bianchi@polimi.it (F. Bianchi), luigi.piroddi@polimi.it (L. Piroddi).

https://doi.org/10.1016/j.nahs.2023.101364

with the estimation of the switching signal grows exponentially with the number of samples. The modeling and parametrization of the switching mechanism is therefore crucial. Many approaches use a Markov chain to model the switching signal, introducing the notion of jump Markov models. The identification of state space jump Markov models has been addressed in the Bayesian framework in [7,8], in the linear and nonlinear case, respectively. A solution for the identification of jump linear input–output models was proposed in [9,10], and later extended to jump Box–Jenkins models [11] and jump polynomial nonlinear AutoRegressive with eXogenous input (nonlinear ARX or NARX, [12,13]) models [14]. Other methods deal with a fully random switching mechanism, assuming that switchings can occur at any time, involving any pair of modes. A Bayesian framework (see, *e.g.*, [6,15,16]) has been developed to segment the training data into an *a priori* known number of subsets, and next the local models are independently estimated, typically in a non-parametric setting using kernel-based methods (with a few exceptions, such as [17]). Algebraic methods are discussed in [18–20]. Finally, randomized methods have been developed as well, such as [21,22].

Regarding the last class of methods, the combinatorial part of the optimization problem is addressed by means of a continuous relaxation approach, by resorting to a probability distribution designed to cover the solution space of all discrete variables. As a result, the combinatorial problem is transformed into the continuous optimization of the distribution parameters, until convergence to a limit distribution with probability mass concentrated at an optimum. The key to this approach is an efficient representation and parametrization of the probability distribution. This relaxation approach was originally applied in a nonlinear parametric identification setting to address the combinatorial problem associated with the model structure selection task [23]. In that context, a joint Bernoulli distribution is employed to parameterize the model structure, each distribution defining the probability that a specific regressor is selected in the model. The joint distribution is updated by means of a *sample-and-evaluate* strategy. This work was later extended to the identification of switched nonlinear systems, by adding a further combinatorial layer to the problem, associated to the selection of the switching signal [20]. However, to avoid an excessive increase of the combinatorial complexity, switching is assumed to occur only at a reduced subset of the data samples. A joint Categorical distribution is employed to associate modes to each time interval between two subsequent candidate switching locations. A refinement stage [21,22] was later added to correct the switching locations, thus providing a more reliable segmentation on the switching signal. Indeed, alternating the updates of the randomized method with the corrections of the refinement stage yields sufficiently accurate identification results, at the cost of an increased computational load.

We here follow the same philosophy, this time fully incorporating the switching signal in the probability distribution (that is, removing the limiting assumption that switchings can occur only at specific times), and avoiding completely the need for a separate refinement stage. The key to this improvement is the introduction of a suitable probabilistic description of the switching locations, which complements the distributions associated to the switching sequence and the nonlinear model structures. More precisely, a switching location is described by a discrete Gaussian distribution, and a number of such distributions is initially assumed. Then, the centers and variances of these distributions are progressively tuned by the randomized algorithm using an update policy that drives the centers towards the switching locations, and diminishes the variance as the uncertainty on the switching locations is reduced. The update process is carried out until convergence is obtained to limit distributions concentrated at specific locations. Redundant distributions are eliminated in the process. The performance of the proposed method has been compared with existing state-of-the-art methods and analyzed in various benchmark examples.

The rest of this paper is organized as follows. The switched NARX (SNARX) model class and the corresponding identification problem are formalized in Section 2. The probabilistic reformulation of the discrete variables in the randomized algorithm framework is discussed in Section 3, followed by the presentation of the identification algorithm in Section 4. Various simulation examples are discussed in Section 5, followed by some concluding remarks in Section 6.

2. Problem statement

A (single-input single-output) switched nonlinear system with K modes is defined as

$$y_t = f^{\sigma_t}(\boldsymbol{x}_t) + e_t \tag{1}$$

where $u_t \in \mathbb{R}$ and $y_t \in \mathbb{R}$ are time-ordered input and output sequences, e_t is a disturbance generally assumed to be a zero-mean Gaussian white noise, and $\boldsymbol{x}_t = [y_{t-1}, \ldots, y_{t-n_y}, u_{t-1}, \ldots, u_{t-n_u}] \in \mathcal{X} \subseteq \mathbb{R}^{n_u+n_y}$ is the vector collecting previous inputs and outputs, $n_u,\ n_y \in \mathbb{N}$ being the (assigned) dynamic orders. The dynamics of the kth mode are described by (the possibly nonlinear) function $f^k : \mathcal{X} \to \mathbb{R}$. The switching signal $\sigma_t \in \{1, \ldots, K\}$ defines the active mode at time t.

The identification of (1) consists in the estimation of both the switching signal $\{\sigma_t\}_{t=1}^{N}$ and the local models $\{f^k(\cdot)\}_{k=1}^{K}$, given an observation training set $\{u_t, y_t\}_{t=1}^{N}$.

2.1. Switching signal

The identification of switched systems entails a significant combinatorial complexity, mainly associated with the correct attribution of the training samples to the modes, which is crucial for a successful identification of the local models. Indeed, *a priori* there are K^N possible switching signals. This complexity is reduced in [20–22] by assuming that the system can switch at most $N_s \ll N$ times at specific locations in the observation horizon, collected in the time-ordered set $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{N_s}\}$, where $\mathcal{T}_i < \mathcal{T}_{i+1}$, for $i = 1, \ldots, N_s - 1$. This reduces the size of the solution space for the switching signal to K^{N_s+1}. The resulting extended set

$$\overline{\boldsymbol{\mathcal{T}}} = [1,\ \mathcal{T}_1,\ \ldots,\ \mathcal{T}_{N_s},\ N+1], \tag{2}$$

induces a segmentation of the observation set into $N_s + 1$ sub-intervals $I_i = [\overline{\mathcal{T}}_i, \overline{\mathcal{T}}_{i+1} - 1]$, each being associated to a single mode $\kappa_i \in \{1, \ldots, K\}$. Accordingly, one can define the corresponding mode sequence

$$\boldsymbol{\kappa} = [\kappa_1,\ \ldots,\ \kappa_{N_s+1}]. \tag{3}$$

Notice that the switching signal can be easily retrieved as $\sigma_t = \kappa_i$, for $t \in I_i$. In this way, $(2N_s + 1)$ discrete variables are employed to fully characterize a switching signal instead of N, and the segmentation of the training set induces K sub-training sets $\{\mathcal{D}_k\}_{k=1}^K$ corresponding to K local models, where $\mathcal{D}_k = \{(y_t, u_t) \,|\sigma_t = k\}$.

2.2. Modes and model structure

Regarding the structure of the local models, this work focuses on the polynomial NARX model class, whereby $f^k(\boldsymbol{x}(\cdot))$ is defined as a polynomial functional expansion of its arguments, and therefore configures a linear regression with respect to monomials obtained from $\boldsymbol{x}$:

$$f^k(\boldsymbol{x}) = \boldsymbol{\varphi}(\boldsymbol{x})^\top \boldsymbol{\vartheta}^k, \tag{4}$$

where $\boldsymbol{\vartheta}^k$ is the parameter vector and $\boldsymbol{\varphi}(\boldsymbol{x}) = [\varphi_1(\boldsymbol{x}),\ \ldots,\ \varphi_n(\boldsymbol{x})]$ is the regressor vector in which, $\varphi_j(\boldsymbol{x}) : \mathbb{R}^{n_y+n_u} \to \mathbb{R}$, $j = 1, \ldots, n$, are the monomials of $\boldsymbol{x}$ up to a given order n_d. So, the orders n_y, n_u and n_d account for the flexibility as well as the complexity of the local nonlinearity.

The identification of a model of type (4) involves the selection of the regressors and the estimation of the corresponding parameters. The first task, a.k.a. model structure selection (see, *e.g.*, [23]), is essential to reduce the complexity of the model and avoid overparametrization issues. Regarding the multi-model structure of the switched system, we assume that all local models share the same set of potential regressors (n_y, n_u and n_d are fixed), although the regressors actually included in each local model may be different. A binary vector $\boldsymbol{s} \in \{0, 1\}^n$ is used to encode the structure of a local model, such that $s_j = 1$ indicates that the jth regressor is included in it, and conversely if $s_j = 0$ the corresponding parameter ϑ_j^k is set to 0. Accordingly, the structure of the modes of the switched system is encoded by a $n \times K$ binary matrix $\boldsymbol{S} = [\boldsymbol{s}^1,\ \ldots,\ \boldsymbol{s}^K]$, where $\boldsymbol{s}^k$ encodes the model structure of the kth mode.

Overall, the structure of the switched model is defined by a discrete variable $\lambda = (\boldsymbol{\mathcal{T}}, \boldsymbol{\kappa}, \boldsymbol{S}) \in \Lambda$, where $\Lambda = \{1, \ldots, N\}^{N_s} \times \{1, \ldots, K\}^{N_s+1} \times \{0, 1\}^{n\times K}$.

2.3. Problem setting

Given a time-ordered data set $\mathcal{D} = \{(y_t, u_t)\}_{t=1}^N$, the identification of the switched model involves the selection of the discrete variable $\lambda \in \Lambda$ and the estimation of the local model parameters $\boldsymbol{\Theta} = \{\boldsymbol{\vartheta}^k\}_{k=1}^K \in \mathbb{R}^{n\times K}$, that minimize the loss function:

$$\mathcal{L}(\lambda,\ \boldsymbol{\Theta}) = \frac{1}{N} \sum_{k=1}^{K} \sum_{t\in\mathcal{D}_k} \left(y_t - \boldsymbol{\varphi}(\boldsymbol{x}_t)^\top \boldsymbol{\vartheta}^k\right)^2 . \tag{5}$$

Note that, for a fixed λ, the local data sets $\{\mathcal{D}_k\}_{k=1}^K$ are completely defined, and the parameters ϑ_j^k such that $S_{j,k} \neq 0$, for $k = 1, \ldots, K$, are estimated by minimizing (5):

$$\boldsymbol{\Theta}^\lambda = \arg\min_{\boldsymbol{\Theta}} \mathcal{L}\left(\lambda,\ \boldsymbol{\Theta}\right). \tag{6}$$

Therefore, the loss function can actually be seen as a function of the discrete variable λ only, which effectively decouples the optimization of the continuous and discrete variables, and justifies the following notation:

$$\mathcal{L}\left(\lambda,\ \boldsymbol{\Theta}^\lambda\right) = \mathcal{L}\left(\lambda\right). \tag{7}$$

Accordingly, the identification problem can be reformulated as the minimization of the loss function with respect to λ:

$$\lambda^\star = \arg\min_{\lambda\in\Lambda} \mathcal{L}\left(\lambda\right). \tag{8}$$

The next sections explain how to address this discrete optimization problem.

For practical reasons, we employ instead of $\mathcal{L}(\lambda)$ a normalized performance index that evaluates the fitting accuracy in a $[0, 1]$ range, namely $\mathcal{J}\left(\lambda\right) = \exp\left(-\mathcal{K}\mathcal{L}\left(\lambda\right)\right)$ where $\mathcal{K}$ is a scaling variable. An exponential index as $\mathcal{J}$ can facilitate the discrimination between models with similar performance by amplifying their difference, thus improving the structure selection process. Accordingly, (8) is equivalently reformulated as the maximization of $\mathcal{J}\left(\lambda\right)$.

3. Probabilistic reformulation of the identification problem

Problem (8) is challenging since it is not feasible in practice to test all possible $\lambda \in \Lambda$. We next discuss a randomized approach to its solution based on a probabilistic reformulation, which can be interpreted as a continuous relaxation. Let ψ be a random variable that takes values in Λ according to a distribution $\mathbb{P}_{\Psi}$. Then, the optimization problem defined in (8) is equivalent to maximizing the mean performance of $\mathbb{P}_{\Psi}$:

$$\mathbb{P}^{\star}_{\Psi} = \arg\max_{\mathbb{P}_{\Psi}} \sum_{\lambda \in \Lambda} \mathbb{P}_{\Psi}(\lambda)\mathcal{J}(\lambda). \tag{9}$$

To see this, assume that $\lambda^{\star}$ is an optimal solution, *i.e.* $\mathcal{J}(\lambda^{\star}) \geq \mathcal{J}(\lambda)$, $\forall \lambda \neq \lambda^{\star}$. Then, it follows that the mean performance of $\mathbb{P}_{\Psi}$ is maximized by a limit probability distribution $\mathbb{P}^{\star}_{\Psi}$ with probability mass concentrated at $\lambda^{\star}$:

$$\mathbb{P}^{\star}_{\Psi}(\lambda) = \begin{cases} 1 & \text{if } \lambda = \lambda^{\star} \\ 0 & \text{Otherwise.} \end{cases} \tag{10}$$

Furthermore, as discussed in Appendix B, when employing a parameterized version of $\mathbb{P}_{\Psi}$, there is a one-to-one correspondence between the optimal solutions of problems (8) and (9), which implies the equivalence of the two optimization problems.

We employ a randomized algorithm to address this optimization problem, whereby samples extracted from $\mathbb{P}_{\Psi}$ are used to gather information regarding convenient choices for the discrete variables in λ. This information is in turn used to update $\mathbb{P}_{\Psi}$ in order to increase the probability of extracting promising model structures. The resulting iterative *sampling-and-evaluation* approach is outlined in Algorithm 1. Each extraction λ^{p} identifies a SNARX system, and can be evaluated by means of (7). Then the distribution $\mathbb{P}_{\Psi}$ is updated taking collectively into account in the update term $g(\cdot)$ the performances of the N_p extracted model structures. The latter is designed to increase the expected performance of $\mathbb{P}_{\Psi}$, *i.e.* to increase the probability that successful model structures are extracted. Upon successful convergence to a limit distribution the extraction probability is concentrated on a single model structure.

Algorithm 1 Outline of the randomized scheme

```
Input: A data set $\{(y_t, u_t)\}_{t=1}^{N}$, a performance evaluation function $\mathcal{J}(\cdot)$, an update strategy $g(\cdot)$, and an initial distribution $\mathbb{P}_{\Psi}$.
while $\mathbb{P}_{\Psi}(\lambda)$ is not a limit distribution do
    Sampling: Extract $N_p$ samples $\{\lambda^{p}\}_{p=1}^{N_p}$ from $\mathbb{P}_{\Psi}$;
    Evaluation: Calculate $\mathcal{J}(\lambda^{p})$, $p = 1, \ldots, N_p$;
    Update: $\mathbb{P}_{\Psi} \leftarrow \mathbb{P}_{\Psi} + g\left(\{\lambda^{p}, \mathcal{J}(\lambda^{p})\}_{p=1}^{N_p}\right)$;
end while
Return $\lambda^{\star} = \psi \sim \mathbb{P}_{\Psi}$.
```

3.1. Representation of $\mathbb{P}_{\Psi}$

The key to an effective application of Algorithm 1 is a convenient parametric representation of the distribution $\mathbb{P}_{\Psi}$ that facilitates the sampling, evaluation and update tasks. We stress here that $\mathbb{P}_{\Psi}$ does not in any way represent a property of the underlying system, but is only instrumental to the extraction of models, and represents in fact the likelihood of each possible model to be the true one. In the absence of any *a priori* information on the SNARX model structure, no correlation among the elements of λ can be exploited, and the latter are accordingly assumed independent. For this reason, without loss of generality, one can assume that $\mathbb{P}_{\Psi}$ is separable as follows:

$$\mathbb{P}_{\Psi}(\lambda) = \mathbb{P}_{\Psi}(\boldsymbol{\mathcal{T}}, \boldsymbol{\kappa}, \boldsymbol{S}) = \mathbb{P}_{\gamma}(\boldsymbol{\mathcal{T}}) \cdot \mathbb{P}_{\xi}(\boldsymbol{\kappa}) \cdot \mathbb{P}_{\rho}(\boldsymbol{S}), \tag{11}$$

where $\mathbb{P}_{\gamma}(\boldsymbol{\mathcal{T}})$, $\mathbb{P}_{\xi}(\boldsymbol{\kappa})$, and $\mathbb{P}_{\rho}(\boldsymbol{S})$ denote the probability to pick a certain set of switching locations, the probability of a certain mode switching sequence, and the probability to pick a certain structure for the local models, respectively. This reasoning carries over to $\mathbb{P}_{\gamma}(\boldsymbol{\mathcal{T}})$, $\mathbb{P}_{\xi}(\boldsymbol{\kappa})$, and $\mathbb{P}_{\rho}(\boldsymbol{S})$, as well, in that we make no *a priori* assumption that the elements of $\boldsymbol{\mathcal{T}}$, $\boldsymbol{\kappa}$, and $\boldsymbol{S}$ are correlated.

Remark 1 (*Independence Assumption*)**.** This *independence* assumption greatly simplifies the extraction and the update phases, while still ensuring that all feasible elements of Λ are considered. In principle, if additional information were available on the system (*e.g.*, it is known that a certain mode transition can never take place, or that certain regressors work better together), it could be incorporated in $\mathbb{P}_{\Psi}(\lambda)$ to provide more focused extractions (along the lines of [24]), but this would come at an additional cost in both the extraction and update phases, which in the end would defeat the purpose. ■

As introduced in [21–23], we employ a Bernoulli distribution to characterize the probability that a certain regressor belongs to a given mode (regressor inclusion probability, RIP), and a Categorical distribution to represent the probability that a certain mode is assigned to a given subperiod of data (mode extraction probability, MEP). These distributions are briefly recalled in the sequel.

The distribution $\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{\mathcal{T}})$ is much less obvious to design. Indeed, the fact that a switching can occur at any time in the observation horizon would suggest adopting a Bernoulli distribution for each time sample $t \in \{1, \dots, N\}$ defining the probability that a switching occurs at t. However, this choice is impracticable, due to the high combinatorial complexity of the representation, which requires N parameters. A more compact representation can be obtained by assuming a discrete Gaussian distribution for the location of each switching (switching location probability, SLP), as discussed in the following.

3.1.1. Parametrization of $\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{\mathcal{T}})$

Let $\boldsymbol{\gamma}$ be a random variable vector with N_s elements each associated to a candidate switching location, with $\gamma_i \in \mathbb{Z}$. We employ a discrete Gaussian distribution [25] centered at $T_i \in \{1, \dots, N\}$ to represent the probability that a switching is located in the proximity of T_i. The confidence level is defined by the variance ω_i^2. Then, the probability distribution of γ_i is given by

$$\mathbb{P}_{\gamma_i}(\mathcal{T}_i) = \frac{1}{K'} \exp\left(-\frac{(\mathcal{T}_i - T_i)^2}{2\omega_i^2}\right), \tag{12}$$

where $\mathcal{T}_i \in \{1, \dots, N\}$ and $K' = \sum_{x=1}^{N} \exp\left(-\frac{(x-T_i)^2}{2\omega_i^2}\right)$ is a normalization factor. As explained previously, we assume that all random variables γ_i, $i = 1, \dots, N_s$ are independent, and therefore the probability distribution of γ is given by

$$\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{\mathcal{T}}) = \prod_{i=1}^{N_s} \mathbb{P}_{\gamma_i}(\mathcal{T}_i). \tag{13}$$

3.1.2. Parametrization of $\mathbb{P}_{\boldsymbol{\xi}}(\boldsymbol{\kappa})$ [21]

To represent the association of the ith sub-interval I_i to a mode, we employ a random variable ξ_i that takes values in $\{1, \dots, K\}$ according to a Categorical distribution with parameters $\boldsymbol{\eta}^i = [\eta_1^i, \dots, \eta_K^i]^\top$, where η_k^i indicates the probability of associating the kth mode to I_i. Note that $\sum_{k=1}^{K} \eta_k^i = 1$.

In view of the independence assumption discussed above, we assume that ξ_i and ξ_j are independent for $i \neq j$. Then, the random vector $\boldsymbol{\xi} = [\xi_1, \dots, \xi_{N_s+1}]$ is distributed according to the aggregate Categorical distribution

$$\mathbb{P}_{\boldsymbol{\xi}}(\boldsymbol{\kappa}) = \prod_{i=1}^{N_s+1} \eta_{\kappa_i}^i. \tag{14}$$

The parameters of this distribution are conveniently collected in a matrix $\boldsymbol{\eta} = [\boldsymbol{\eta}^1, \dots, \boldsymbol{\eta}^{N_s+1}]$.

3.1.3. Parametrization of $\mathbb{P}_{\boldsymbol{\rho}}(\boldsymbol{S})$ [23]

The presence of a regressor in the model structure of one mode is modeled by a random variable $\rho_{j,k}$ that takes values in $\{0, 1\}$ according to a Bernoulli distribution, $\rho_{j,k} \sim \mathrm{Be}(\mu_{j,k})$, where $j = 1, \dots, n$ and $k = 1, \dots, K$. If $\rho_{j,k} = 1$ the jth regressor is active in the kth mode, and absent otherwise. Clearly, if $\mu_{j,k}$ is the probability that $\rho_{j,k} = 1$, the probability of rejecting the jth regressor is $1 - \mu_{j,k}$.

A vector $\boldsymbol{\rho}^k = [\rho_1^k, \dots, \rho_n^k]$, with n independent and identically distributed random variables, encodes the structure of the kth local model with the probability parameters $\boldsymbol{\mu}^k = [\mu_{1,k}, \dots, \mu_{n,k}]^\top$. To encompass all modes, a multivariate Bernoulli distribution is employed for the matrix of random variables $\boldsymbol{\rho} = [\boldsymbol{\rho}^1, \dots, \boldsymbol{\rho}^K] \in \{1, 0\}^{n \times K}$. Its parameters are also conveniently collected in a matrix, namely $\boldsymbol{\mu} = [\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^K]$. In view of the usual independence assumption, the probability distribution of the random matrix $\boldsymbol{\rho}$ is given by

$$\mathbb{P}_{\boldsymbol{\rho}}(\boldsymbol{S}) = \prod_{k=1}^{K} \left(\prod_{j:S_{j,k}=1} \mu_{j,k} \prod_{j:S_{j,k}=0} \left(1 - \mu_{j,k}\right) \right). \tag{15}$$

The notation associated to the probabilistic reformulation of the problem is summarized in Table 1 for the reader's convenience.

4. Randomized algorithm for the identification of switched NARX system

As shown in Fig. 1, the randomized algorithm designed for the identification of SNARX systems iterates a 3-step sequence, involving three phases: the sampling stage, the evaluation stage and the update stage.

Table 1
Notation employed in the probabilistic reformulation.

Variable explanation	Deterministic variable	Stochastic variable	Distribution parameters
Switching locations	$\boldsymbol{\mathcal{T}}$	$\boldsymbol{\gamma} \sim \mathbb{P}_{\boldsymbol{\gamma}}$	$\boldsymbol{T}, \boldsymbol{\omega}$
Switching sequence	$\boldsymbol{\kappa}$	$\boldsymbol{\xi} \sim \mathbb{P}_{\boldsymbol{\xi}}$	$\boldsymbol{\eta}$
Local model structures	$\boldsymbol{S}$	$\boldsymbol{\rho} \sim \mathbb{P}_{\boldsymbol{\rho}}$	$\boldsymbol{\mu}$

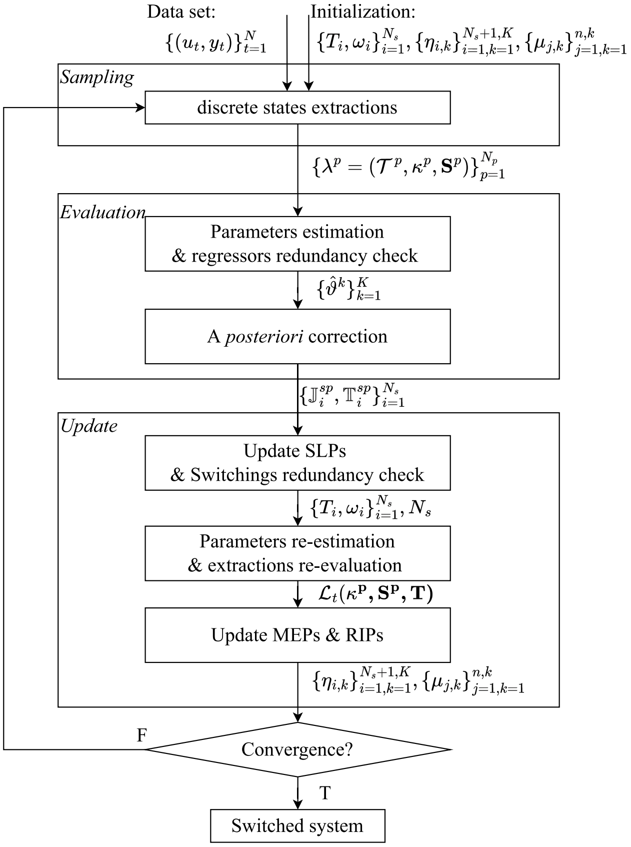

Data set:
$\{(u_t, y_t)\}_{t=1}^{N}$
Initialization:
$\{T_i, \omega_i\}_{i=1}^{N_s}, \{\eta_{i,k}\}_{i=1,k=1}^{N_s+1,K}, \{\mu_{j,k}\}_{j=1,k=1}^{n,k}$
Sampling
discrete states extractions
$\{\lambda^p = (\mathcal{T}^p, \kappa^p, \mathbf{S}^p)\}_{p=1}^{N_p}$
Evaluation
Parameters estimation
& regressors redundancy check
$\{\hat{\vartheta}^k\}_{k=1}^{K}$
A posteriori correction
$\{\mathbb{J}_i^{sp}, \mathbb{T}_i^{sp}\}_{i=1}^{N_s}$
Update
Update SLPs
& Switchings redundancy check
$\{T_i, \omega_i\}_{i=1}^{N_s}, N_s$
Parameters re-estimation
& extractions re-evaluation
$\mathcal{L}_t(\kappa^\mathbf{p}, \mathbf{S}^\mathbf{p}, \mathbf{T})$
Update MEPs & RIPs
$\{\eta_{i,k}\}_{i=1,k=1}^{N_s+1,K}, \{\mu_{j,k}\}_{j=1,k=1}^{n,k}$
F
Convergence?
T
Switched system

Fig. 1. Flowchart of the proposed algorithm.

4.1. Sampling stage

Extracting a sample λ from $\mathbb{P}_{\boldsymbol{\Psi}}$ implies extracting separately:

- a set of switching locations $\boldsymbol{\mathcal{T}}^p$ from $\mathbb{P}_{\boldsymbol{\gamma}}(\boldsymbol{\mathcal{T}})$ in (13);
- a mode switching sequence $\boldsymbol{\kappa}^p$ from $\mathbb{P}_{\boldsymbol{\xi}}(\boldsymbol{\kappa})$ in (14);
- a set of local model structures $\boldsymbol{S}^p$ from $\mathbb{P}_{\boldsymbol{\rho}}(\boldsymbol{S})$ in (15).

The sampling procedure is repeated N_p times, yielding the samples $\{\lambda^p\}_{p=1}^{N_p}$, with $\lambda^p = (\boldsymbol{\mathcal{T}}^p, \boldsymbol{\kappa}^p, \boldsymbol{S}^p)$. Samples not satisfying the following conditions are rejected (and substituted):

(i) $\boldsymbol{\kappa}^p$ must include all modes;
(ii) the switching locations must be ordered, *i.e.* $\mathcal{T}_i^p < \mathcal{T}_{i+1}^p$.

4.2. Evaluation stage

4.2.1. Parameter estimation

The optimal parameters of the local models defined by λ^p are estimated with ordinary Least Squares, yielding:

$$\hat{\boldsymbol{\vartheta}}^k = \left(\boldsymbol{\Phi}_k^\top \boldsymbol{\Phi}_k\right)^{-1} \boldsymbol{\Phi}_k^\top \boldsymbol{Y}, \qquad (16)$$

with $k = 1, \dots, K$. In expression (16) $\boldsymbol{\Phi}_k \in \mathbb{R}^{N_k \times \hat{n}_k}$ is the regressor matrix stacking on its rows the $\hat{n}_k$ regressors $\boldsymbol{\varphi}(\boldsymbol{x}_t)$ extracted for the kth mode in the pth sample, and $\boldsymbol{Y} \in \mathbb{R}^{N_k \times 1}$ is the time-ordered vector collecting the outputs of the kth mode. Both $\boldsymbol{\Phi}_k$ and $\boldsymbol{Y}$ are limited to data in $\mathcal{D}_k = \{(y_t, \boldsymbol{x}_t) \,|\sigma_t^p = k\}$, *i.e.* the subset of samples associated to the kth mode according to the switching signal resulting from λ^p, $N_k = \#\{\mathcal{D}_k\}$ being the corresponding number of samples.

Remark 2. To avoid overfitting issues [23], the statistical significance of the regressors in the extracted structure $\boldsymbol{s}^k$ is validated *a posteriori* by a Student's t-test. The model parameters are re-estimated after rejecting the regressors that are judged not statistically significant by the test. □

$\mathcal{L}(\lambda)$ is determined as the minimum of (5), corresponding to the optimal parameterizations $\hat{\boldsymbol{\vartheta}}^k$, $k = 1, \dots, K$, and the associated exponential index $\mathcal{J}(\lambda)$ is also calculated.[1]

4.2.2. A posteriori *correction of the switching locations*

The location of the switchings (and, therefore, the corresponding segmentation of the observation window) can be significantly improved by taking into account the estimated mode dynamics. To this end, suppose that $\kappa_i^p \neq \kappa_{i+1}^p$. Then, we apply a local search method around $\mathcal{T}_i^p$ to find the switching position that minimizes the loss function, given the current estimated mode dynamics associated to the κ_i^p and κ_{i+1}^p modes. The search window is defined as $[\min_p\{\mathcal{T}_i^p\}, \max_p\{\mathcal{T}_i^p\}]$. Obviously the same scheme cannot be applied if $\kappa_i^p = \kappa_{i+1}^p$ (*i.e.* $\mathcal{T}_i^p$ is not an actual switching location in the pth extraction), and thus $\mathcal{T}_i^p$ will not be corrected in that case. This refinement procedure is repeated for all elements in $\boldsymbol{\mathcal{T}}^p$. The corrected SLP centers and the corresponding local performances are grouped per switching pattern $sp = (\kappa_i^p, \kappa_{i+1}^p)$, in the sets $\mathbb{T}_i^{sp}$ and $\mathbb{J}_i^{sp}$, respectively, for reasons that will be clear in the following. A pseudo-code version of the proposed *a posteriori* correction is provided in Algorithm 2.

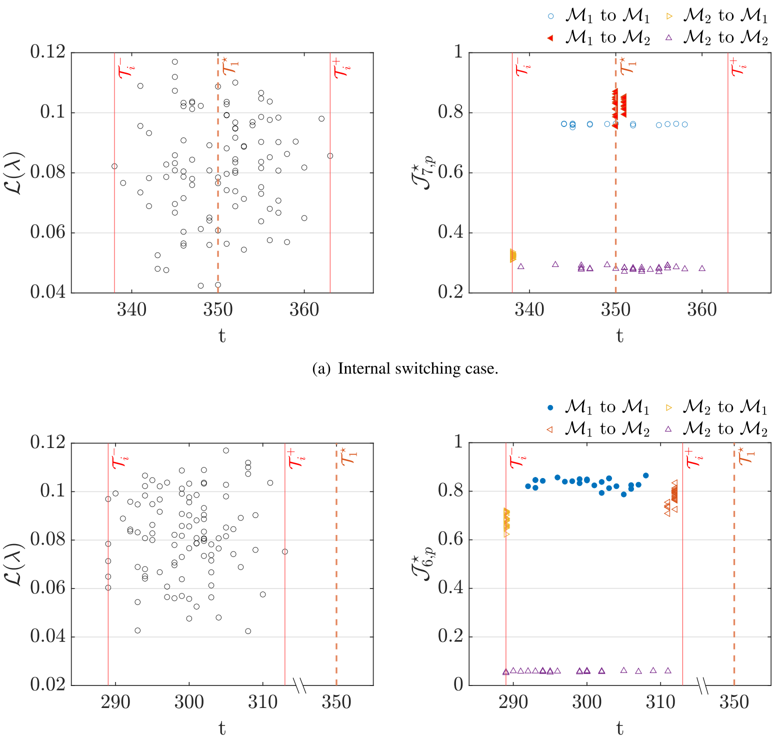
Algorithm 2 *A posteriori* correction

```
Input: $\{\mathcal{T}_i^p\}_{i=1,p=1}^{N_s,N_p}$, $\{\kappa_i^p\}_{i=1,p=1}^{N_s+1,N_p}$, $\{\hat{\boldsymbol{\vartheta}}^k\}_{k=1}^K$, $\mathcal{K}$.
Output: $\{\mathbb{T}_i^{sp}, \mathbb{J}_i^{sp}\}_{i=1}^{N_s}$, where $sp = (m, n)$, $m, n = 1, \dots, K$
for $i = 1$ to $N_s$ do
    $\mathbb{T}_i^{sp} \leftarrow [\,]$, $\mathbb{J}_i^{sp} \leftarrow [\,]$, $\forall sp$;
    for $p = 1$ to $N_p$ do
        if $\kappa_i^p \neq \kappa_{i+1}^p$ then
            $\mathcal{T}_i^- \leftarrow \min_p\{\mathcal{T}_i^p\}$;  $\mathcal{T}_i^+ \leftarrow \max_p\{\mathcal{T}_i^p\}$;
            $t^\star \leftarrow \mathcal{T}_i^-$;  $L^\star \leftarrow \infty$;
            for $\hat{t} \leftarrow \mathcal{T}_i^-$ to $\mathcal{T}_i^+$ do
                $L \leftarrow \sum_{t=\mathcal{T}_i^-}^{\hat{t}-1} \left(y_t - \boldsymbol{\varphi}(\boldsymbol{x}_t)^\top \hat{\boldsymbol{\vartheta}}^{\kappa_i^p}\right)^2 + \sum_{t=\hat{t}}^{\mathcal{T}_i^+} \left(y_t - \boldsymbol{\varphi}(\boldsymbol{x}_t)^\top \hat{\boldsymbol{\vartheta}}^{\kappa_{i+1}^p}\right)^2$;
                if $L < L^\star$ then $L^\star \leftarrow L$;  $t^\star \leftarrow \hat{t}$;
                end if
            end for
        else
            $t^\star \leftarrow \mathcal{T}_i^p$;
            $L^\star \leftarrow \sum_{t=\mathcal{T}_i^-}^{\mathcal{T}_i^+} \left(y_t - \boldsymbol{\varphi}(\boldsymbol{x}_t)^\top \hat{\boldsymbol{\vartheta}}^{\kappa_i^p}\right)^2$;
        end if
        $\mathcal{J}^\star \leftarrow \exp(-\mathcal{K}L^\star)$;
        $sp = (\kappa_i^p, \kappa_{i+1}^p)$;
        $\mathbb{T}_i^{sp} \leftarrow [\mathbb{T}_i^{sp}, t^\star]$;
        $\mathbb{J}_i^{sp} \leftarrow [\mathbb{J}_i^{sp}, \mathcal{J}^\star]$;
    end for
end for
```

The next example illustrates the role of the *a posteriori* correction in the estimation of the switching locations (the simulations are taken from Example 1, see Section 5.1, at the first iteration of the algorithm).

Fig. 2 shows the distribution of the extractions of two SLPs before and after the *a posteriori* correction. In the latter case, the extractions are clearly clustered according to the switching patterns, the cluster associated to the best performance

[1] The scaling parameter is set to $\mathcal{K} = 10^{-(\min(\lfloor \log_{10}(\mathcal{L}(\lambda)) \rfloor)+1)}$ at the first iteration, to guarantee an adequate discrimination ability on the model performance in the [0, 1] range.



(a) Internal switching case.

(b) External switching case.

Fig. 2. Example 1: Distribution of the extracted switching locations before and after the *a posteriori* correction (extracted switching locations (left), corrected results (right)). The true switching location $\mathcal{T}_i^\star$ is marked by a dashed orange line. The corrected switching locations are grouped based on the associated switching pattern. The group with the maximum mean performance index is marked by filled symbols.

indicating the most likely switching pattern. Restricting attention to this switching pattern, denoted $sp^\star$, if it corresponds to an actual switching (*i.e.*, $\kappa_i^p \neq \kappa_{i+1}^p$), one also gets a very precise indication regarding the location of the switching. For example, in case (a) $sp^\star = (\mathcal{M}_1, \mathcal{M}_2)$, which is clustered in the proximity of the true switching (see Fig. 2(a), right sub-figure).

Conversely, Fig. 2(b) refers to an *external switching* situation, in that there is no real switching in the search window. Accordingly, here $sp^\star = (\mathcal{M}_1, \mathcal{M}_1)$, indicating that this whole region should probably be assigned to mode 1. Notice also that both switching patterns $(\mathcal{M}_1, \mathcal{M}_2)$ and $(\mathcal{M}_2, \mathcal{M}_1)$ are clustered near the borders of the search window, which confirms that it is unlikely that a switching can occur inside.

4.3. Update stage

4.3.1. Update of the SLPs

Let $sp^\star = (m, n) = \arg\max_{sp} \bar{\mathbb{J}}_i^{sp}$, where $\bar{\mathbb{J}}_i^{sp}$ denotes the average performance index associated to the switching pattern sp at the ith SLP. Let also $\mathbb{J}_i^\star = \mathbb{J}_i^{sp^\star}$ and $\mathbb{T}_i^\star = \mathbb{T}_i^{sp^\star}$.

In the external switching case ($m = n$), there is not a clear indication of where to move the corresponding SLP in order to find a switching. Therefore, its center T_i is not modified ($T_i^{new} = T_i$), but its variance is increased, so as to extend the exploration region: $\omega_i^{new} = \beta\omega_i$, where $\beta > 1$ is a hyper-parameter.[2]

In the internal switching case ($m \neq n$), the SLP center is updated to the weighted position:

$$T_i^{new} = \left[\frac{\sum_{q=1}^{\#\{\mathbb{J}_i^\star\}} \mathbb{J}_{i,q}^\star \cdot \mathbb{T}_{i,q}^\star}{\sum_{q=1}^{\#\{\mathbb{J}_i^\star\}} \mathbb{J}_{i,q}^\star} \right], \tag{17}$$

where $\#\{\mathbb{J}_i^\star\}$ is the length of vector $\mathbb{J}_i^\star$, and $[\cdot]$ is the round operator. The update to the weighted position according to expression (17) can be nicely interpreted as a local gradient ascent rule as discussed in Appendix A.

[2] In all experiments, we set $\beta = 1.1$.

The reliability of this indication is related to the dispersion of the results associated to $sp^\star$. For example, at the early stages of the algorithm, it is not unusual that these results should be widely spread due to coarse modeling and insufficient accuracy. To account for this, in the internal switching case we generally decrease the variance of the SLP unless the dispersion is large, according to the following update law:

$$\omega_i^{new} = a\omega_i + (1-a)\omega_{i,0}\mathcal{P}_i, \tag{18}$$

where $0 < a < 1$ (*e.g.*, $a = 0.9$), $\omega_{i,0}$ is the initial variance of the SLP, and $0 \leq \mathcal{P}_i \leq 1$ is a dispersion indicator for the ith SLP, defined as follows:

$$\mathcal{P}_i = \frac{\frac{1}{\#\{\mathbb{T}_i^\star\}}\sum_{q=1}^{\#\{\mathbb{T}_i^\star\}} |T_i - \mathbb{T}_{i,q}^\star|}{\max(\mathcal{T}_i^+ - T_i,\ T_i - \mathcal{T}_i^-)}. \tag{19}$$

If $\mathcal{P}_i = 1$, the variance will tend to revert to the initial value $\omega_{i,0}$, whereas if the dispersion is sufficiently small, the variance will decrease to 0.

4.3.2. SLP redundancy check

SLPs with significant overlap, *i.e.* such that $T_{i+1} - T_i < 2(\omega_{i+1} + \omega_i)$, are merged into one, to avoid redundancy. The center of the resulting SLP is set between the two original centers according to the weighted location:

$$T_i^{new} = \frac{\omega_{i+1}T_i + \omega_i T_{i+1}}{\omega_i + \omega_{i+1}}, \tag{20}$$

which places it nearer to the center of the original SLP with smaller variance. The standard deviation of the merged SLP is calculated so as to cover 99% of the confidence interval of the SLP with a smaller variance:

$$\omega_i^{new} = \begin{cases} \dfrac{T_{i+1} - T_i^{new} + 3\omega_{i+1}}{3} & \text{if} \quad \omega_i > \omega_{i+1} \\ \dfrac{T_i^{new} - T_i + 3\omega_i}{3} & \text{otherwise} \end{cases} \tag{21}$$

With a similar rationale, SLPs close to the borders (*i.e.* such that either $T_i + 3\omega_i$ touches the right boundary or $T_i - 3\omega_i$ touches the left one) are updated by increasing the variance according to $\omega_i^{new} = \beta\omega_i$, and moving the center far from the border by $3(\omega_i^{new} - \omega_i)$. In this way, the SLPs close to the borders are not lost, but can eventually detect a switching location.

Remark 3. Notice that the mechanism explained previously to eliminate redundant SLPs can be indirectly exploited to estimate the number of switchings – which is generally unknown – during the identification process. Indeed, one can start the identification procedure assuming a number of switchings equal to an estimated upper bound, and then the rules described above can get rid of redundant SLPs. See also the example in Section 5.1.

4.3.3. Update of the MEPs and RIPs

After the refinement of the switching locations, the parameters of the local models are re-estimated, this time assuming the switchings fixed at the locations $\boldsymbol{\mathcal{T}} = [T_1, \dots, T_{N_s}]$ for all extractions, *i.e.* using the modified extracted structures $\hat{\lambda}^p = (\boldsymbol{\mathcal{T}}, \boldsymbol{\kappa}^p, \boldsymbol{S}^p)$. Then, the model performance is evaluated and the point-wise loss $\mathcal{L}_t(\hat{\lambda}^p)$ calculated for $t = 1, \dots, N$.

The update of the MEPs and the RIPs is carried out along the same lines of [20–23]. For the MEPs, the general principle is that the probability of assigning I_i to the kth mode (associated to the parameter η_i^k in a MEP) should be increased if the mean performance of the extractions such that $\kappa_i^p = k$ is higher than other choices, $\kappa_i^p \neq k$, and *vice versa*. Accordingly, the MEPs are updated as follows:

$$\eta_{i,k}^{new} = \eta_{i,k} + \chi_i^M \delta_{i,k} \tag{22}$$

for $i = 1, \dots, N_s$ and $k = 1, \dots, K$, where $\chi_i^M > 0$ is a step size, and the update factor $\delta_{i,k}$ is defined as

$$\delta_{i,k} = \frac{1}{\#\{p|\kappa_i^p = k\}} \sum_{p|\kappa_i^p = k} J_i^M(\hat{\lambda}^p) - \frac{1}{\#\{p|\kappa_i^p \neq k\}} \sum_{p|\kappa_i^p \neq k} J_i^M(\hat{\lambda}^p), \tag{23}$$

where $J_i^M(\hat{\lambda}^p) = \exp\left(-\mathcal{K}_i^M L_i^M(\hat{\lambda}^p)\right)$ is the local performance index, $L_i^M(\hat{\lambda}^p)$ being the mean loss in the time interval I_i.

Similarly, the RIPs are updated according to the law:

$$\mu_{j,k}^{new} = \mu_{j,k} + \chi_k^R l_{j,k}, \tag{24}$$

for $j = 1, \dots, n$ and $k = 1, \dots, K$, where $\chi_k^R > 0$ is the step size, and the update term $l_{j,k}$ is defined as

$$l_{i,k} = \frac{1}{\#\{p|S_{j,k}^p = 1\}} \sum_{p|S_{j,k}^p = 1} J_k^R(\hat{\lambda}^p) - \frac{1}{\#\{p|S_{j,k}^p = 0\}} \sum_{p|S_{j,k}^p = 0} J_k^R(\hat{\lambda}^p), \tag{25}$$

where $J_k^R(\hat{\lambda}^p) = \exp\left(-\mathcal{K}_k^R L_k^R(\hat{\lambda}^p)\right)$, the loss $L_k^R(\hat{\lambda}^p)$ being calculated only on the sup-periods associated to the kth local model in the pth extraction.

Since the update terms in the previous expressions are based on performance averages computed on the extracted samples, the confidence that can be placed on them is limited, especially at early iterations, which justifies the inertia term in the update equations.

The step sizes χ_i^M and χ_k^R are set adaptively as follows, to take into account the dispersion of the extracted samples [21]:

$$\begin{aligned} \chi_i^M &= 1/\left(10(\widetilde{J}_i^M(\hat{\lambda}^p) - \bar{J}_i^M(\hat{\lambda}^p)) + 0.1\right); \\ \chi_k^R &= 1/\left(10(\widetilde{J}_k^R(\hat{\lambda}^p) - \bar{J}_k^R(\hat{\lambda}^p)) + 0.1\right) \end{aligned} \tag{26}$$

where $\widetilde{J}_i^M(\hat{\lambda}^p)(\widetilde{J}_k^R(\hat{\lambda}^p))$ and $\bar{J}_i^M(\hat{\lambda}^p)(\bar{J}_k^R(\hat{\lambda}^p))$ are the maximum and mean performance indices over all extractions. Briefly, if the extracted samples exhibit a wide range of performance index values the update term is not fully reliable (this occurs often at the early iterations), and accordingly the step size is set to a small value. The opposite applies when the dispersion is small.

Finally, suitable saturation thresholds are introduced to keep $\eta_{i,k}$ and $\mu_{j,k}$ in proper intervals [21,23]. Actually, these probabilities are never allowed to go to 0 or 1, to preserve the exploration capabilities of the algorithm.

4.4. Stopping criterion

The identification procedure operates by iteratively updating the distribution $\mathbb{P}_{\Psi}$ to increase its mean performance. The mean performance is approximated by its sampled counterpart obtained over the extractions $\hat{\lambda}^p$, $1, \ldots, N_p$, obtained at a given iteration:

$$\bar{\mathcal{J}} = \frac{1}{N_p}\frac{1}{N}\sum_{p=1}^{N_p}\sum_{t=1}^{N}\mathcal{J}_t(\hat{\lambda}^p). \tag{27}$$

The algorithm terminates when $\bar{\mathcal{J}}$ does not increase anymore. In practice, for increased robustness, we check that $\bar{\mathcal{J}}$ does not vary more than a given tolerance ϵ for a certain number (*e.g.*, 5) of consecutive iterations. Upon termination the probability distribution is saturated to the nearest limit distribution.

5. Simulation examples

Four examples are discussed in this section. All tests are performed in a MATLAB R2021a environment [26] on an Intel(R) Core(TM) i7-6700K CPU @4.0 GHz with 32G of RAM. In all examples, the confidence level in the statistical test is set to 0.001. The initial RIPs and MEPs are set to 0.1 and $1/K$,[3] respectively, and the lower bounds are set to $\eta_{min} = 0.01$ and $\mu_{min} = 0.001$.

We will rate the performance of the estimated models with the following two indices:

- The classification error rate, evaluated as $\mathrm{CER} = \#\{t|\sigma_t \neq \hat{\sigma}_t\}$
- The fitting performance index, defined as $\mathrm{FIT} = 100\left(1 - \frac{\|\hat{\boldsymbol{y}}-\boldsymbol{y}\|_2^2}{\|\boldsymbol{y}-\bar{y}\|_2^2}\right)$, where $\boldsymbol{y}$ is the output in the training set, $\bar{y}$ is the mean value, and $\hat{\boldsymbol{y}}$ is the one-step-ahead prediction output of $\boldsymbol{y}$.

5.1. Example 1: A 2-mode SNARX case

Consider the following SNARX system [27], with 2 modes:

$$\begin{aligned} \mathcal{M}_1: &\quad y_t = -0.905y_{t-1} + 0.9u_{t-1} + e_t, \\ \mathcal{M}_2: &\quad y_t = -0.4y_{t-1}^2 + 0.5u_{t-1} + e_t, \end{aligned} \tag{28}$$

where u_t is the input signal distributed in [0, 1] uniformly and e_t is a zero mean white noise with variance 0.012 (SNR=18). Starting at $\mathcal{M}_1$, the system switches between the two modes at $\mathcal{T}^\star = [350\ 1450\ 1600\ 1750]$ in an observation window of $N = 2000$ samples. We initialize the SLPs associated to the candidate switchings centered at $\boldsymbol{T}^{(0)} = [50, 100, 150, \ldots, 1950]$, with initial standard deviations set to 5. The orders of the local models are set to $n_u = 2$, $n_y = 2$ and $n_d = 2$. Consequently, there are a total of 15 candidate regressors in each mode. $N_p = 100$ extractions are sampled and evaluated at every iteration.

Initially, as many as 39 SLPs are present, which sets to 2^{40} the number of possible switching sequences. Fortunately, even though the number of SLPs is largely overestimated, many redundant ones are quickly removed during the

[3] Parameter $\eta_{1,1}$ is initialized to 1 to reduce the multiplicity of equivalent solutions.

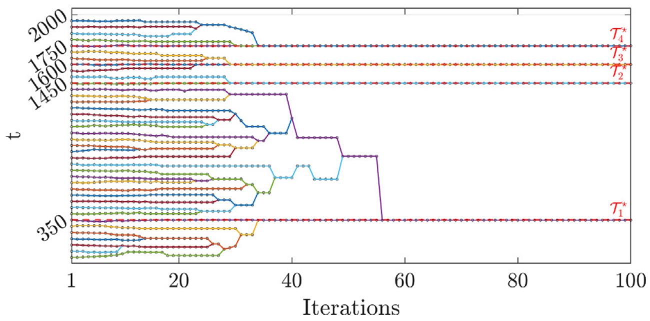


Fig. 3. Example 1: Refinement and reduction of the switching locations over iterations.

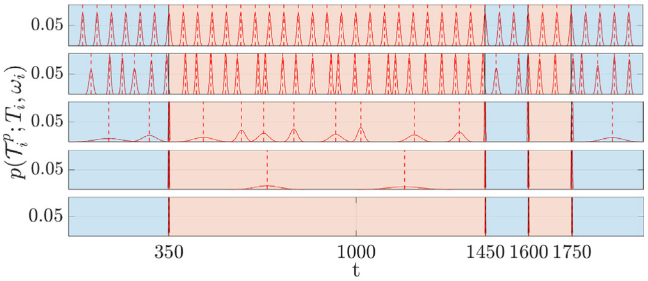


Fig. 4. Example 1: Evolution of the SLPs over iterations. From top to bottom: initialization, 10th, 30th, 40th, and 60th iteration. Different background colors indicate the true mode assignment over time.

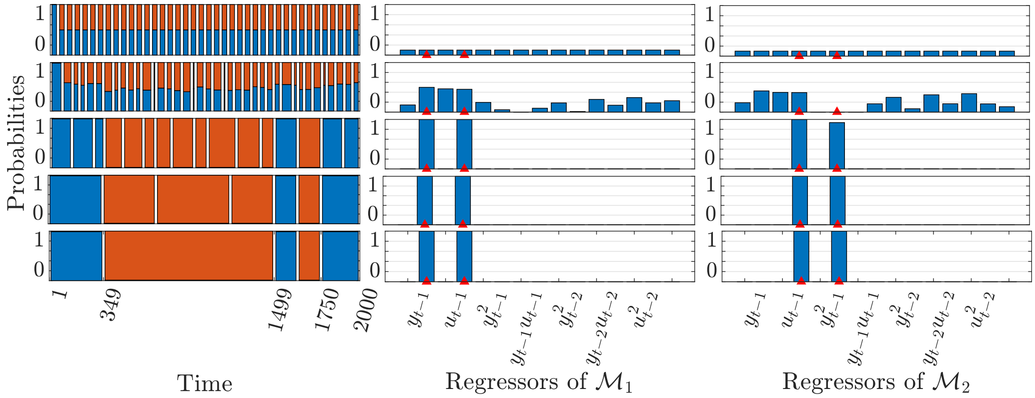


Fig. 5. Example 1: Evolution of the MEPs and RIPs over iterations. From left to right: MEPs ($\mathcal{M}_1$ in blue, $\mathcal{M}_2$ in red), RIPs of $\mathcal{M}_1$, RIPs of $\mathcal{M}_2$ (true regressors are marked by red triangles). From top to bottom: initialization, 10th, 30th, 40th, and 60th iteration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

identification (compare *e.g.* the results at iteration 30 with the initial settings in Fig. 3), thus progressively reducing the combinatorial complexity of the problem. More precisely, the SLPs nearest to the true switchings quickly converge to the correct locations and reduce their variance, as can be appreciated in Fig. 4. Conversely, the variances of the SLPs that do not "cover" any true switching are enlarged to search for actual switchings, triggering various SLP merging operations. Fig. 5 shows that around the 30th iteration, although there are still some spurious switching locations, each interval is unequivocally assigned to one mode, with the correct switching pattern, and the local model structures have already converged to the true ones.

Table 2 gives some aggregate results relative to 100 MC runs. The proposed algorithm shows high efficiency and precision in the estimation of the switching signal. In particular, the CER is as low as 0.17%, corresponding to 3.4 samples over 2000, and the FIT is 94.45%. Notice also the extremely low variance of the two indices, indicating a remarkable consistency of the results over the MC runs. Furthermore, the algorithm was able to estimate the switching sequence by exploring and testing a relatively small number of switching sequences.

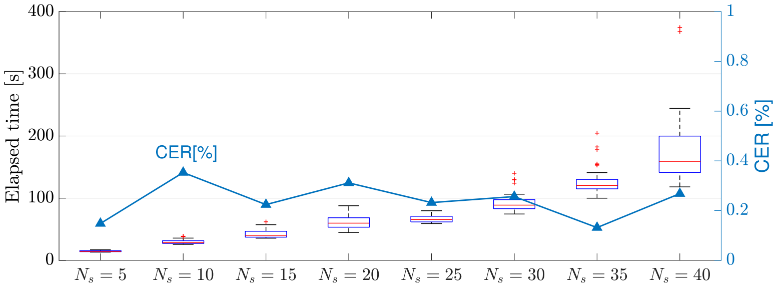


Fig. 6. Example 1: MC analysis for increasing N_s: elapsed time (boxplots, left axis) and average CER (blue line, right axis). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Example 1: MC simulation results.

Average elapsed time [s]	17.68
Percentage of correct selection of $\boldsymbol{\kappa}$ [%]	100
Average # of explored switching sequences	2326.3
# of allowed switching sequences	1.1E12
CER [%] (mean (std.))	0.17 (0.04)
FIT [%] (mean (std.))	94.45 (1.1E−3)
Percentage of correct selection of $\boldsymbol{s}_1$ [%]	100
Average # of explored models for $\mathcal{M}_1$	997
# of possible model structures for $\mathcal{M}_1$	32768
Percentage of correct selection of $\boldsymbol{s}_2$ [%]	100
Average # of explored models for $\mathcal{M}_2$	1159
# of possible model structures for $\mathcal{M}_2$	32768

A further test was carried out, to verify the ability of the proposed method to deal with a large number of switchings. For this purpose, we allow the system to switch between the two modes for N_s times, the switching locations being set randomly according to $\mathcal{T}_i^\star \sim \mathcal{N}(500i, 5)$, $i = 1, \ldots, N_s$. A 50-run MC is carried out for each value of $N_s \in \{5, 10, \ldots, 40\}$. Fig. 6 (right axis) indicates that the CER is under 0.5% in all realizations. Even when dealing with a large data set (*i.e.*, $N = 20500$, $N_s = 40$), no more than 0.27% of the samples are misclassified on average.

5.2. Example 2: A 3-mode SNARX case

Consider the following SNARX system presented in [21]:

$$\begin{aligned} \mathcal{M}_1 : y_t &= 0.5y_{t-1} + 0.8u_{t-2} + u_{t-1}^2 - 0.3y_{t-2}^2 + e_t, \\ \mathcal{M}_2 : y_t &= 0.2y_{t-1}^3 - 0.5y_{t-2} - 0.7y_{t-2}u_{t-2}^2 + 0.6u_{t-2}^2 + e_t, \\ \mathcal{M}_3 : y_t &= 0.4y_{t-1}^3 + 0.5y_{t-2} - 0.7y_{t-2}u_{t-2}^2 + 0.6u_{t-2}^2 + e_t, \end{aligned} \tag{29}$$

where u_t is the input signal uniformly distributed in $[-1,\ 1]$ and e_t is a Gaussian white noise with zero mean and variance 0.01 (SNR = 23). The SNARX system switches 5 times at $\mathcal{T}^\star = [500, 1030, 2115, 2740, 3000]$ in the observation window with $N = 3400$ samples, and the true switching sequence is $\boldsymbol{\kappa}^\star = [1, 2, 1, 3, 2, 3]$. The pre-defined model orders are $n_u = 2$, $n_y = 2$, $n_d = 3$, for a total of 35 candidate model structures for each mode. The SLPs are initialized with centers at $\boldsymbol{T}^{(0)} = [300, 600, \ldots, 3300]$ (*i.e.*, every 300 samples) and standard deviation $\omega_i^{(0)} = 15$ for all i. The algorithm samples $N_p = 200$ extractions at each iteration.

Besides the additional mode, this example poses another challenge with respect to the previous one, in that $\mathcal{M}_2$ and $\mathcal{M}_3$ share the same structure. Nonetheless, the algorithm is again effective in locating the switching locations, quickly removing the redundant SLPs, and driving the other ones to the correct locations (see the update of the SLPs in Fig. 7).

It is interesting to compare the proposed method with the two-stage solutions developed in [21,22]. Fig. 8 shows the average loss over time for all three methods. Both two-stage methods ultimately achieve the same performance as the proposed one, but require various restarts (after the refinement stage) and a significantly larger computational time. Regarding the proposed method, it is important to remark that the evaluation stage costs an average of 0.4022 s per iteration, while the *a posteriori* correction, as a part of the evaluation stage, takes just 0.0544s per iteration on average (see Fig. 9).

Table 3 reports the results for a 100 run MC simulation. Apparently, the proposed algorithm never misses any switching, after testing only 5909 possible sequences on average among 177 147 admissible ones. The model structures of modes $\mathcal{M}_2$ and $\mathcal{M}_3$ are occasionally slightly different from the true ones, but this does not significantly affect the model accuracy,

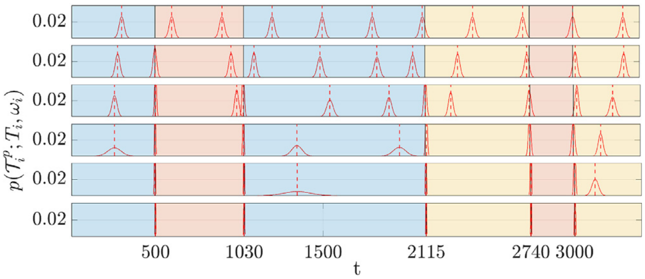


Fig. 7. Example 2: Evolution of the SLPs over iterations. From top to bottom: initialization, 10th, 20th, 30th, 40th, and 50th iteration. Different background colors indicate the true mode assignment over time.

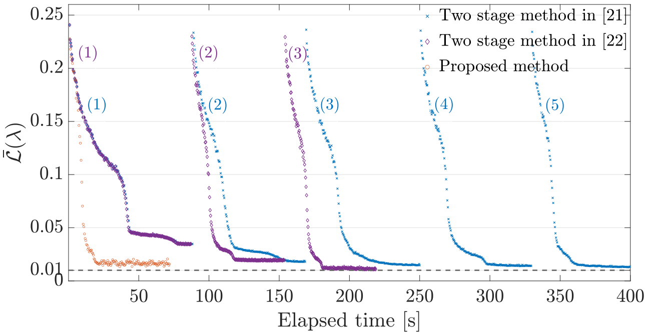


Fig. 8. Example 2: Comparison between the proposed method and previous randomized-based approaches. The number in parenthesis counts the algorithm re-starts in the two-stage schemes.

Table 3
Example 2: MC simulation results.

Elapsed time [s]	73.6
CER [%] mean (std.)	0.09 (0.02)
FIT [%] mean (std.)	95.55 (0.05)
Percentage of correct selection of $\boldsymbol{\kappa}$ [%]	100
Aver. # of explored switching sequences	5909
# of allowed switching sequences	177147
Percentage of correct selection of $\boldsymbol{s}_1$ [%]	100
Average # of explored models for $\mathcal{M}_1$	5612
Percentage of correct selection of $\boldsymbol{s}_2$ [%]	98
Average # of explored models for $\mathcal{M}_2$	7018
Percentage of correct selection of $\boldsymbol{s}_3$ [%]	88
Average # of explored models for $\mathcal{M}_3$	6230
# of possible structures (for all modes)	3.4E10

as can be appreciated with the CER and FIT indices. As evident from Table 3, all 100 runs yield models with comparable CER and FIT values.

Tables 4 and 5 give an overview of the obtained results in terms of the average elapsed time, the CER, and the percentage of correct identifications of each switching location (for the purpose of this analysis, a switching location is assumed to be estimated correctly if the error is within 3 samples, *i.e.*, $|\mathcal{T}_i^\star - T_i| \leq 3$). In this respect, notice that a large SLP variance and a sparse set of SLPs hampers the detection of close switchings (*i.e.*, such that $T_{i+1} - T_i \leq 2(\omega_i + \omega_{i+1})$). In the example, such a problem occasionally occurs with $\mathcal{T}_3^\star$ (at sample 2740) and $\mathcal{T}_4^\star$ (at sample 3000). On the other hand, notice that the combinatorial complexity increases exponentially with the number of initial SLPs, which may explain the failed MC run when using an SLP every 100 samples.[4] Overall, there is a relatively large range of values of ω_i and initial settings of the centers of the SLPs for which the behavior of the algorithm is more than acceptable.

[4] In the failed realization only two local models are identified, which conflicts with the *a priori* information on the number of modes, which would prompt the user to re-identify the system anyway.

Table 4
Example 2: Analysis on the number of initial SLPs (100 MC runs for each set)

$T_i^{(0)}$	$100i$	$200i$	$300i$	$400i$	$500i$
$\omega_i^{(0)}$	15				
Elapsed time [s]	80.6	74.2	73.6	68.6	65.4
CER [%]	0.16	0.09	0.09	0.09	0.09
% of correct identification of $\mathcal{T}_1^\star$	100	100	100	100	100
% of correct identification of $\mathcal{T}_2^\star$	100	100	100	100	100
% of correct identification of $\mathcal{T}_3^\star$	99	100	100	100	100
% of correct identification of $\mathcal{T}_4^\star$	99	100	100	100	100

Table 5
Example 2: Analysis on the initial variances of the SLPs (100 MC runs for each set)

$T_i^{(0)}$	$300i$			
$\omega_i^{(0)}$	5	15	25	35
Elapsed time [s]	74.2	73.6	72.2	71.0
CER [%]	0.10	0.09	0.09	0.79
% of correct identification of $\mathcal{T}_1^\star$	100	100	100	100
% of correct identification of $\mathcal{T}_2^\star$	100	100	100	100
% of correct identification of $\mathcal{T}_3^\star$	100	100	100	91
% of correct identification of $\mathcal{T}_4^\star$	100	100	100	91

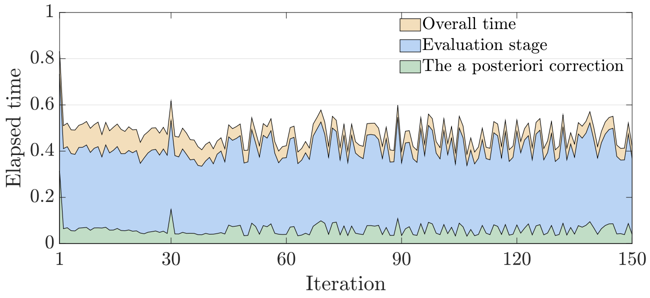


Fig. 9. Example 2: Iteration time of the proposed algorithm.

5.3. Example 3: An LPV system

We next address the identification of a linear parameter varying (LPV) system used in [28]:

$$y_t = \vartheta_1^i y_{t-1} - 0.7y_{t-2} + u_{t-1} - 0.5u_{t-2} + e_t, \tag{30}$$

where u_t, y_t and e_t are the input, output and noise signal, respectively, u_t being defined as a Pseudo Random Binary sequence and e_t as a Gaussian white noise with zero mean and standard deviation 0.5 (SNR = 38). Parameter ϑ_1^i is the varying parameter scheduled by a switching signal and has four possible values, *i.e.* −1.5, −1, −0.5, and 0.5, which determine four different modes. Switching occurs at $\boldsymbol{\mathcal{T}}^\star = [400, 810, 1270, 1500, 1830, 2150]$ over an observation horizon of $N = 2500$ time-ordered samples, and the switching sequence is $\boldsymbol{\kappa}^\star = [1, 2, 3, 2, 3, 4, 1]$.

The identification of this system is particularly challenging, as the model structure is the same for all modes, with only one parameter changing. The proposed algorithm is here initialized assuming a switching every 100 samples, *i.e.* setting 24 SLPs with centers at $T_i = 100i$, and standard deviations $\omega_i = 10$, $k = 1, \dots, 24$. At each iteration, $N_p = 100$ extractions are sampled.

The proposed algorithm is first compared with the SON-EM algorithm proposed in [28]. To make a fair comparison, we skip the model structure selection part, and assume from the start that the true regressors ($[y_{t-1}, y_{t-2}, u_{t-1}, u_{t-2}]$) are known. The two algorithms show a comparable accuracy on the estimated parameters of the models (see Fig. 10), but the SON-EM algorithm is not as efficient in the estimation of the switching signal, with several outliers. The results obtained with the proposed approach are also consistent with those of the two-stage randomized algorithm (see Fig. 9 in [21]).

Over a batch of 100 MC runs, the proposed algorithm never lost any switching location. The accuracy of the parameter estimation can be appreciated in Table 6. Conversely, 9 failed runs out of 100 were reported by running the algorithm in [21]. Even considering only the successful runs in the comparison, the distribution of the detected switchings (see the blue barplots in Fig. 11) is much sharper with the proposed algorithm, as also witnessed by the extremely low CER value (0.17%) achieved on average. The average FIT equals 97.27%, with a standard deviation of 0.03.

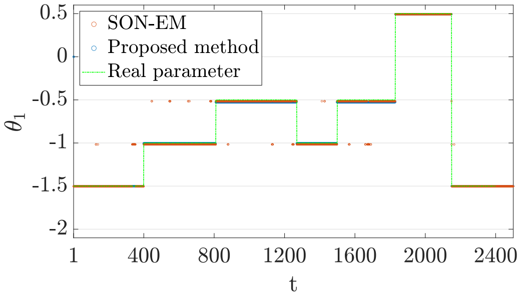


Fig. 10. Example 3: Estimation of θ_1 with the proposed method and the SON-EM algorithm.

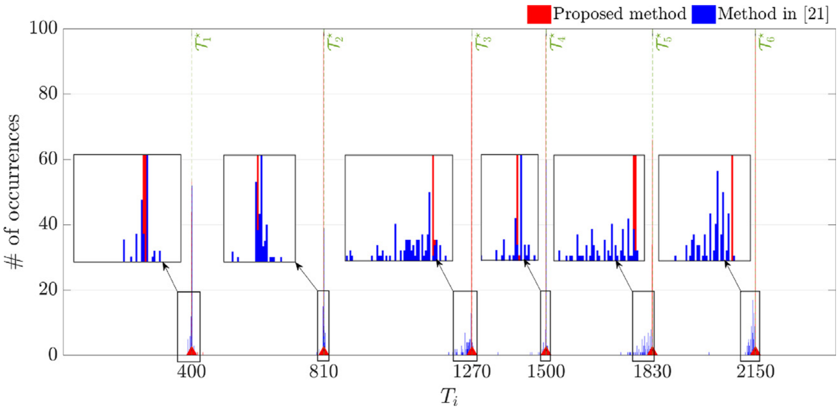


Fig. 11. Example 3: Distribution of the detected switching locations. A comparison between the proposed method (red bars) and the two-stage randomized algorithm in [21] (blue bars). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Example 3: Parameter estimation performance over 50 MC runs.

Parameter	Mode 1 (mean (std.))	Mode 2 (mean (std.))
ϑ_1	−1.4999 (6.53E−5)	−1.0029 (1.60E−3)
ϑ_2	−0.7007 (6.01E−4)	−0.7071 (1.40E−3)
ϑ_3	1.0198 (1.57E−3)	1.0161 (2.56E−3)
ϑ_4	−0.5032 (1.97E−3)	−0.5233 (1.14E−3)
Parameter	Mode 3 (mean (std.))	Mode 4 (mean (std.))
ϑ_1	−0.5258 (1.74E−4)	0.4907 (4.54E−3)
ϑ_2	−0.7065 (4.76E−5)	−0.7093 (1.61E−3)
ϑ_3	0.9960 (3.07E−4)	1.0004 (4.03E−3)
ϑ_4	−0.4618 (1.93E−4)	−0.5189 (1.15E−4)

Table 7
Example 3: Robustness w.r.t to noise. A comparison between the proposed algorithm and the method in [9] (50 MC runs for each set).

std(e) (SNR)	0.1 (755)		0.3 (85)		0.5 (36)	
Algorithm	Prop.	Method [9]	Prop.	Method [9]	Prop.	Method [9]
mean FIT	99.84	92.38	98.78	88.83	97.14	88.18
max FIT	99.87	98.05	98.83	94.51	97.21	92.20
min FIT	99.78	81.82	98.72	82.38	97.04	83.63

Finally, the proposed algorithm is compared with the method introduced in [9] to analyze its robustness with respect to noise. In the experiments, the output is corrupted by white noise with standard deviation std($\boldsymbol{e}$) = 0.1, 0.3 and 0.5.

For each noise level 50 MC runs were carried out with each algorithm. The resulting FIT values are reported in Table 7. The proposed algorithm consistently shows higher (and less variable) performances for all tested levels of noise.

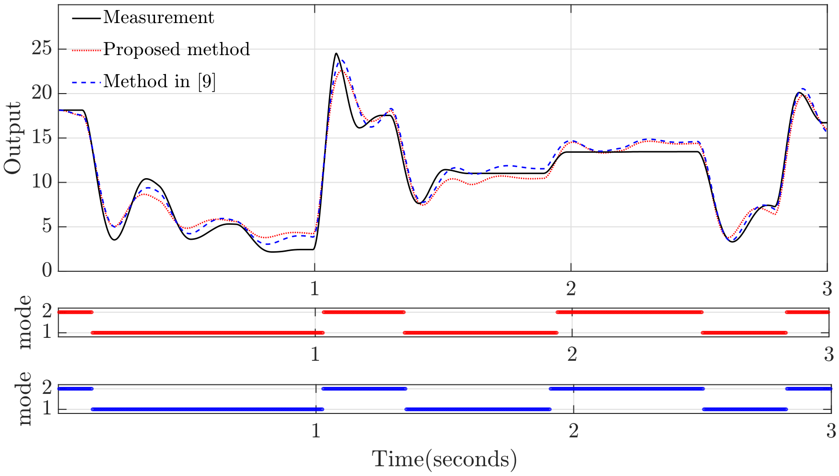


Fig. 12. Example 4: Simulated vs. measured output on the validation data set (top); switching signal estimated using our method (middle), and using the method proposed in [9] (bottom).

5.4. Example 4: A pick-and-place machine

We consider the identification of the hybrid dynamics of a pick-and-place process introduced in [29]. An electronic component is attracted by the mounting head and then placed on the printed circuit board (PCB), which is characterized by two main operation modes: (1) the *free mode* (the mounting head carries the electronic component without touching the circuit board), and (2) the *impact mode* (the mounting head is in contact with the PCB). This process has been used as a benchmark example in several works [4,9,30].

A data record has been collected over an interval of 15s, with a sampling frequency of 400 Hz. This includes the voltage applied to the motor (the input signal, u) and the vertical position of the mounting head (the output signal, y). The data are split into three disjoint subsets: (1) the training set that includes $N = 4400$ samples gathered in the first 11 s, (2) the evaluation set including 400 samples in 1 s after the training set which is used to tune the initial SLPs, and (3) the validation set with the remaining $N_v = 1200$ samples.

We compare our method with the algorithm proposed in [9] that is specifically studied for fitting jump models, and which is reported to have a better performance compared to the clustering approach of [31] on this example. A SNARX model with $K = 2$ modes is employed to model the placement process. To make a fair comparison, we use linear local models such that $\boldsymbol{\varphi}(\boldsymbol{x}_t) = [1, y_{t-1}, y_{t-2}, u_{t-1}, u_{t-2}]$ and ignore the model structure selection part. The proposed algorithm has been applied starting with initial SLPs set to $T_i = 40i$ and $\omega_i = 8$ for $i = 1, \ldots, 109$. We set $N_p = 200$ in this example. After the identification, a Voronoi diagram is employed to define a piece-wise map in the regressors space, that drives the mode-switching mechanism.

Fig. 12 compares the output measurements with the corresponding open-loop simulations on the validation set, and shows the predicted switching signal. Mode 1 and Mode 2 can be associated to the *free mode* and the *impact mode*, respectively, according to the physical knowledge of the process. The resulting FIT of the proposed algorithm is equal to 95.7%, which is slightly lower than in [9] (96.8%). Still, the performance of our method is considerably better than the clustering approach of [31] which yields a FIT of 93.8%. We here emphasize the advantages of the proposed method in terms of increased robustness compared to [9] (see Table 7) and the additional ability to deal with the model structure selection of nonlinear local models.

6. Conclusions

A randomized algorithm for the identification of switched nonlinear ARX systems has been proposed, with special focus on the estimation of the location of mode switchings. We use Gaussian distributions to provide a probabilistic characterization of the switching locations, which smoothly fits in a previously developed framework of a randomized identification method, that already included a probabilistic representation of the mode switching sequence and the local model structures. These distributions are tuned by way of a sample-and-evaluate strategy. Compared with previous versions of the randomized method, all discrete variables can now be estimated in a single run, and switching refinements and algorithm re-starts are avoided. While showing a higher identification efficiency, the proposed algorithm also achieves higher precision in the estimation of the switching locations. Two key elements that enable this improvement are the *a posteriori* correction of the switching locations and the variance adaptation mechanism, that facilitate and accelerate the tuning of the switching locations.

The proposed algorithm is potentially able to estimate the correct number of switching locations by removing the redundant ones (subtractive strategy). Further work will focus also on additive strategies allowing the inclusion of additional switching locations and modes as needed, to reduce the combinatorial complexity and the *a priori* information required by the algorithm, especially regarding the number of modes.

CRediT authorship contribution statement

Miao Yu: Conceptualization, Methodology, Software, Writing – original draft, Formal analysis. **Federico Bianchi:** Conceptualization, Methodology, Writing – review & editing. **Luigi Piroddi:** Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request

Acknowledgments

Miao Yu gratefully acknowledges the financial support for his doctoral studies provided by the China Scholarship Council (CSC).

Appendix A. Update to the weighted position

The performance of $\mathbb{P}_{\boldsymbol{\Psi}}$ can be calculated as $\mathbb{E}_{\mathbb{P}_{\boldsymbol{\Psi}}(\lambda)}\left[\mathcal{J}(\lambda)\right]$. Let us calculate the gradient of this expression with respect to the center T_i of the ith SLP[5]:

$$\nabla_{T_i}\mathbb{E}_{\mathbb{P}_{\boldsymbol{\Psi}}(\lambda)}\left[\mathcal{J}(\lambda)\right] = \mathbb{E}_{\mathbb{P}_{\boldsymbol{\Psi}}(\lambda)}\left[\mathcal{J}(\lambda)\ \nabla_{T_i}\log\mathbb{P}_{\boldsymbol{\Psi}}(\lambda)\right]$$

where we have used the well known *log-derivative trick*. Estimating this expected value via Monte Carlo sampling one gets:

$$\nabla_{T_i}\mathbb{E}_{\mathbb{P}_{\boldsymbol{\Psi}}(\lambda)}\left[\mathcal{J}(\lambda)\right] \cong \frac{1}{N_p}\sum_{p=1}^{N_p}\mathcal{J}(\lambda^p)\ \nabla_{T_i}\log\mathbb{P}_{\boldsymbol{\Psi}}(\lambda^p)$$

Now, in view of the independence assumption ($\mathbb{P}_{\boldsymbol{\Psi}}(\lambda)$ is the product of the individual probabilities associated to the elements of λ), of the properties of the logarithm (the logarithm of the product is the sum of logarithms), and considering that only $\mathbb{P}_{\gamma_i}(\mathcal{T}; T_i)$ depends on parameter T_i, the previous expression can be simplified to:

$$\nabla_{T_i}\mathbb{E}_{\mathbb{P}_{\boldsymbol{\Psi}}(\lambda)}\left[\mathcal{J}(\lambda)\right] \cong \frac{1}{N_p}\sum_{p=1}^{N_p}\mathcal{J}(\lambda^p)\ \nabla_{T_i}\log\mathbb{P}_{\gamma_i}(\mathcal{T}_i^p; T_i)$$

Now, since

$$\begin{aligned}\nabla_{T_i}\log\mathbb{P}_{\gamma_i}(\mathcal{T}_i^p; T_i) &= \nabla_{T_i}\log\left(\frac{1}{\sqrt{2\pi\omega_i^2}}\exp\left(-\frac{(\mathcal{T}_i^p - T_i)^2}{2\omega_i^2}\right)\right)\\ &= \frac{\mathcal{T}_i^p - T_i}{\omega_i^2},\end{aligned}$$

one obtains

$$\nabla_{T_i}\mathbb{E}_{\mathbb{P}_{\boldsymbol{\Psi}}(\lambda)}\left[\mathcal{J}(\lambda)\right] \cong \frac{1}{\omega_i^2}\frac{1}{N_p}\sum_{p=1}^{N_p}\mathcal{J}(\lambda^p)\,(\mathcal{T}_i^p - T_i).$$

[5] For the purpose of this calculation we will assume that SLPs are modeled as *continuous* Gaussians.

Using this gradient to update T_i according to the gradient ascent method, the following update rule would be obtained:

$$T_i^{new} = T_i + \chi \frac{1}{\omega_i^2} \frac{1}{N_p} \sum_{p=1}^{N_p} \mathcal{J}(\lambda^p)(\mathcal{T}_i^p - T_i),$$

which, by setting the adaptive step size $\chi = \frac{\omega_i^2 N_p}{\sum_{p=1}^{N_p} \mathcal{J}(\lambda^p)}$, becomes:

$$T_i^{new} = \frac{\sum_{p=1}^{N_p} \mathcal{J}(\lambda^p)\mathcal{T}_i^p}{\sum_{p=1}^{N_p} \mathcal{J}(\lambda^p)}.$$

Update Eq. (17) is inspired by the previous expression, but incorporates two modifications that accelerate convergence and provide reasonable results even when using few samples in the Monte Carlo estimation of the expected value. First of all, a local performance index measured on the search window is used in the update equation, as opposed to a global index, as this is more directly informative on the performance of the SLP. On a similar line, not all samples are taken into account, but only those associated to the best performing switching pattern ($sp^\star$).

Appendix B. Equivalence of the reformulated optimization problem

There is a one to one correspondence between the optimal solutions of the optimization problem (8) and its reformulated version (9), consisting in the maximization of the mean performance of $\mathbb{P}_{\Psi}$. As already discussed, if the first problem admits an optimal solution $\lambda^\star$ then the reformulated problem admits a corresponding optimal solution $\mathbb{P}^\star_{\Psi}$, *i.e.* the limit probability distribution with probability mass concentrated at $\lambda^\star$.

If $\lambda^\star$ is not unique, however, the reformulated problem has other optimal solutions as well. Without loss of generality, suppose that there are two equivalent optimal solutions $\lambda^\star$ and $\lambda^{\star\star}$, *i.e.* $\mathcal{J}(\lambda^\star) = \mathcal{J}(\lambda^{\star\star}) > \mathcal{J}(\lambda)$, $\forall\lambda \neq \lambda^\star, \lambda^{\star\star}$. Besides the 2 limit distributions corresponding to $\lambda^\star$ and $\lambda^{\star\star}$, all "mixed" distributions, such that $\mathbb{P}(\lambda^\star) + \mathbb{P}(\lambda^{\star\star}) = 1$, with both $\mathbb{P}(\lambda^\star) > 0$ and $\mathbb{P}(\lambda^{\star\star}) > 0$ (and, clearly, $\mathbb{P}(\lambda) = 0$, $\forall\lambda \neq \lambda^\star, \lambda^{\star\star}$) also maximize the mean performance.

Fortunately, since in practice we employ a parameterized version of the probability distribution (*e.g.* $\mathbb{P}(\lambda) = \mathbb{P}(\lambda_1)\mathbb{P}(\lambda_2)\ldots\mathbb{P}(\lambda_{N_\lambda})$, where λ_i, $i = 1, \ldots, N_\lambda$, are the elements of vector λ), such "mixed" distributions cease to be optimal. Indeed, $\mathbb{P}(\lambda^\star) > 0$ implies that the probabilities of all individual elements of $\lambda^\star$ are positive, and a similar condition applies for the elements of $\lambda^{\star\star}$, given that $\mathbb{P}(\lambda^{\star\star}) > 0$. This implies that the models extracted from this distribution can be $\lambda^\star$ and $\lambda^{\star\star}$, but also mixed models containing elements of both solutions. Since the latter are not optimal, the mean performance of this distribution is also suboptimal.

Therefore, maximizing the mean performance of $\mathbb{P}_{\Psi}$ will yield the same optimal solutions as the original problem.

References

[1] R. Vidal, S. Soatto, A. Chiuso, Applications of hybrid system identification in computer vision, in: 2007 European Control Conference, ECC, IEEE, 2007, pp. 4853–4860.

[2] I. Maruta, T. Sugie, Identification of PWA models via data compression based on l 1 optimization, in: 2011 50th IEEE Conference on Decision and Control and European Control Conference, IEEE, 2011, pp. 2800–2805.

[3] R. Zimmerschied, R. Isermann, Nonlinear system identification of block-oriented systems using local affine models, IFAC Proc. Vol. 42 (10) (2009) 658–663.

[4] D. Piga, A. Bemporad, A. Benavoli, Rao-blackwellized sampling for batch and recursive Bayesian inference of piecewise affine models, Automatica 117 (2020) 109002.

[5] A. Garulli, S. Paoletti, A. Vicino, A survey on switched and piecewise affine system identification, IFAC Proc. Vol. 45 (16) (2012) 344–355.

[6] F. Lauer, G. Bloch, Hybrid System Identification: Theory and Algorithms for Learning Switching Models, vol. 478, Springer, Cham, Switzerland, 2018.

[7] A. Svensson, T.B. Schön, F. Lindsten, Identification of jump Markov linear models using particle filters, in: 53rd IEEE Conference on Decision and Control, IEEE, 2014, pp. 6504–6509.

[8] E. Özkan, F. Lindsten, C. Fritsche, F. Gustafsson, Recursive maximum likelihood identification of jump Markov nonlinear systems, IEEE Trans. Signal Process. 63 (3) (2014) 754–765.

[9] A. Bemporad, V. Breschi, D. Piga, S.P. Boyd, Fitting jump models, Automatica 96 (2018) 11–21.

[10] A. Andriën, D.J. Antunes, Near-optimal recursive identification for Markov switched systems, in: 2021 60th IEEE Conference on Decision and Control, CDC, IEEE, 2021, pp. 132–138.

[11] D. Piga, V. Breschi, A. Bemporad, Estimation of jump Box–Jenkins models, Automatica 120 (2020) 109126.

[12] I.J. Leontaritis, S.A. Billings, Input-output parametric models for non-linear systems part I: Deterministic non-linear systems, Internat. J. Control 41 (2) (1985) 303–328.

[13] I.J. Leontaritis, S.A. Billings, Input-output parametric models for non-linear systems part II: Stochastic non-linear systems, Internat. J. Control 41 (2) (1985) 329–344.

[14] F. Bianchi, V. Breschi, D. Piga, L. Piroddi, Model structure selection for switched NARX system identification: A randomized approach, Automatica 125 (2021) 109415.

[15] G. Pillonetto, A new kernel-based approach to hybrid system identification, Automatica 70 (2016) 21–31.

[16] A. Scampicchio, G. Pillonetto, A new model selection approach to hybrid kernel-based estimation, in: 2018 IEEE Conference on Decision and Control, CDC, IEEE, 2018, pp. 3068–3073.

[17] L. Bako, K. Boukharouba, E. Duviella, S. Lecoeuche, A recursive identification algorithm for switched linear/affine models, Nonlinear Anal. Hybrid Syst. 5 (2) (2011) 242–253.

[18] Z. Wang, H. An, X. Luo, Switch detection and robust parameter estimation for slowly switched Hammerstein systems, Nonlinear Anal. Hybrid Syst. 32 (2019) 202–213.

[19] Y. Tian, T. Floquet, L. Belkoura, W. Perruquetti, Algebraic switching time identification for a class of linear hybrid systems, Nonlinear Anal. Hybrid Syst. 5 (2) (2011) 233–241.

[20] F. Bianchi, M. Prandini, L. Piroddi, A randomized approach to switched nonlinear systems identification, IFAC-PapersOnLine 51 (15) (2018) 281–286.

[21] F. Bianchi, M. Prandini, L. Piroddi, A randomized two-stage iterative method for switched nonlinear systems identification, Nonlinear Anal. Hybrid Syst. 35 (2020) 100818.

[22] M. Yu, F. Bianchi, L. Piroddi, A switched nonlinear system identification method with switching location refinement, in: 2021 60th IEEE Conference on Decision and Control, CDC, IEEE, 2021, pp. 858–863.

[23] A. Falsone, L. Piroddi, M. Prandini, A randomized algorithm for nonlinear model structure selection, Automatica 60 (2015) 227–238.

[24] F. Bianchi, A. Falsone, M. Prandini, L. Piroddi, A randomised approach for NARX model identification based on a multivariate Bernoulli distribution, Internat. J. Systems Sci. 48 (6) (2017) 1203–1216.

[25] A. Karmakar, S.S. Roy, O. Reparaz, F. Vercauteren, I. Verbauwhede, Constant-time discrete Gaussian sampling, IEEE Trans. Comput. 67 (11) (2018) 1561–1571.

[26] MATLAB, Version 9.10.0.1602886 (R2021a), The MathWorks Inc., Natick, Massachusetts, 2021.

[27] F. Lauer, G. Bloch, Switched and piecewise nonlinear hybrid system identification, in: International Workshop on Hybrid Systems: Computation and Control, Springer, 2008, pp. 330–343.

[28] A. Hartmann, J.M. Lemos, R.S. Costa, J. Xavier, S. Vinga, Identification of switched ARX models via convex optimization and expectation maximization, J. Process Control 28 (2015) 9–16.

[29] A.L. Juloski, W.M.H. Heemels, G. Ferrari-Trecate, Data-based hybrid modelling of the component placement process in pick-and-place machines, Control Eng. Pract. 12 (10) (2004) 1241–1252.

[30] K. Boukharouba, L. Bako, S. Lecoeuche, Identification of piecewise affine systems based on Dempster-Shafer theory, IFAC Proc. Vol. 42 (10) (2009) 1662–1667.

[31] G. Ferrari-Trecate, M. Muselli, D. Liberati, M. Morari, A clustering technique for the identification of piecewise affine systems, Automatica 39 (2) (2003) 205–217.