2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Maximum-a-posteriori estimation of jump Box-Jenkins models

Valentina Breschi, Dario Piga, Alberto Bemporad

Abstract— **Complex dynamical systems and time series can often be described by jump models, namely finite collections of local models where each sub-model is associated to a different operating condition of the system or segment of the time series. Learning jump models from data thus requires both the identification of the local models and the reconstruction of the sequence of active modes. This paper focuses on *maximum-a-posteriori* identification of jump Box-Jenkins models, under the assumption that the transitions between different modes are driven by a stochastic Markov chain. The problem is addressed by embedding prediction error methods (tailored to Box-Jenkins models with switching coefficients) within a coordinate ascent algorithm, that iteratively alternates between the identification of the local Box-Jenkins models and the reconstruction of the mode sequence.**

I. Introduction

In the last decades the problem of learning jump models, which are characterized by both discrete and continuous states, has attracted the attention of researchers from the *system identification* and *machine learning* communities. By exploiting multiple yet simple sub-models, this class of models can be effectively used to describe the behavior of nonlinear and complex systems, even when they are characterized by sudden changes in their operating condition (*e.g.,* power electronic circuits, robot grasping objects [10], just to cite a few). Indeed, the discrete state of the model indicates the operating condition of the underlying system, while the local models characterize its behavior at the different modes. Furthermore, jump models can be used to address problems on time-series segmentation and clustering, like segmenting human motion and action [12], as well as speech recognition [11]. In this case, the discrete state indicates the cluster which each segment belongs to, and the local models describe the time-series within each temporal segment.

Depending on the considered class of jump models, the evolution of the discrete state might be governed by *deterministic* [15] or *stochastic* rules [6]. In particular, for Markov jump models, if the continuous dynamics is assumed to be known or it is neglected, the problem of learning jump models reduces to *hidden Markov Model* [2]. However, local models are often needed to characterize the behavior of the true system at the different operating conditions. Furthermore, local models might enhance the performance of segmentation approaches. Different methods have been proposed in the literature to learn both the discrete and continuous dynamics of a jump model. Examples range from the identification of *piecewise affine autoregressive models with exogenous inputs* (PWARX) [5], [9] to the estimation of more general *switching autoregressive models with exogenous inputs* (SARX) [1], [3], [7].

Although these problems are tackled both in deterministic [1], [3] and Bayesian [7], [9] settings, most of existing methods to learn jump models rely on the assumption of *autoregressive* (AR) local models. Therefore, they usually exploit the conditional independence of the current output on past modes, given the current mode and the past noisy measurements of the output. Only few contributions relax this assumption and consider local *output-error* (OE) sub-models [8], [14] or *autoregressive moving average models with exogenous inputs* (ARMAX) [4]. However, most approaches for the identification of local OE models rely on the hypothesis that switches are dictated by a polyhedral partition of the state+input space, thus neglecting any stochastic information on the discrete state. Analogously, although the method recently proposed by the authors in [4] is applicable to more general switching systems, it does not exploit the stochastic information on the discrete state, which can be retrieved by formulating the identification problem within a Bayesian framework.

In this work, we tackle the problem of learning jump *Box-Jenkins* (BJ) models from data, under the assumption that switches between different discrete states are driven by a first-order Markov Chain. We thus consider quite a general class of jump models, since both ARX, ARMAX and OE models are particular instances of BJ models. Nonetheless, by learning these local models, we exploit the generality and flexibility of the *Box-Jenkins* structure, at the price of missing independence of the outputs given past data. Indeed, the output observations are not conditionally independent given the current mode and the regressor containing past measurements, thus increasing the complexity of the problem at hand.

The method proposed in this paper relies on the maximization of the posterior distribution of all the unknowns characterizing the jump BJ model. Specifically, the proposed approach allows us to estimate: the parameters Θ of the local models; the transition matrix M that governs the Markov chain driving mode switches; the variance σ_e^2 of the noise affecting the output measurements and the (hidden) sequence of active modes $\mathcal{S}^T$. The resulting algorithm alternates be-

This work was partially supported by the H2020-723248 project *DAEDALUS - Distributed control and simulation platform to support an ecosystem of digital automation developers* and by the Lombardia region and the Cariplo foundation, under the project *Learning to Control (L2C)*, no. 2017-1520.

V. Breschi is with Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy. valentina.breschi@polimi.it

D. Piga is with IDSIA Dalle Molle Institute for Artificial Intelligence, SUPSI-USI, Manno, Switzerland. dario.piga@supsi.ch

A. Bemporad is with IMT School for Advanced Studies Lucca, Lucca, Italy alberto.bemporad@imtlucca.it

978-1-7281-1397-5/19/$31.00 ©2019 IEEE

tween two main steps: (i) the estimation of the parameters Θ, σ_e^2, M, and (ii) the inference of the active mode sequence $\mathcal{S}^T$. The parameters Θ are identified through an instance of the *prediction error method* (PEM), tailored to BJ models with switching coefficients, while the discrete state sequence $\mathcal{S}^T$is estimated via a sub-optimal moving-horizon approach.

The paper is organized as follows. The class of jump *Box-Jenkins* systems is described in Section II. The addressed estimation problem is presented in Section III, by introducing the posterior distribution of the model parameters along with our assumptions on priors distributions of the unknown variables. The method proposed to learn jump *Box-Jenkins* model is then described in Section IV, and its effectiveness is shown through simulation results in Section V.

A. Notation

Let $\mathbb{R}^+$ be the set of positive real numbers and $\mathbb{R}^{n\times m}$ be the set of real matrices of dimension $n \times m$. Given a matrix M, $M_{i,:}$ indicates its i-th row, $M_{i,j}$ denotes its entry in position (i, j) and $vec(M) \in \mathbb{R}^{nm}$ is the column vector obtained by stacking the columns of M. Given a random matrix $M \in \mathbb{R}^{n\times m}$, we indicate the probability distribution of $vec(M)$ as $p(M)$. Given the natural numbers $a, b \in \mathbb{N}$, $\mathbb{I}_{[a=b]}$ is the indicator function of the event $\{a = b\}$, *i.e.,*

$$\mathbb{I}_{[a=b]} = \begin{cases} 1 \text{ if } a = b, \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

For $\xi \in \mathbb{R}^+$, $\Gamma(\xi)$ denotes the *Gamma* function, *i.e.,*

$$\Gamma(\xi) = \int_0^{+\infty} x^{\xi-1} e^{-x}\, dx. \tag{2}$$

II. SETTING AND GOAL

Consider a time sequence $\mathcal{U}^T = \{u_t\}_{t=1}^T$, constituted by the inputs $u_t \in \mathbb{R}$ exciting a single-input single-output jump linear dynamical system with $K \in \mathbb{N}$ *Box-Jenkins* local models. The system returns a sequence of outputs $\mathcal{Y}^T = \{y_t\}_{t=1}^T$, with $y_t \in \mathbb{R}$ generated as

$$y_t = y_t^{\text{o}} + v_t, \tag{3a}$$

where y_t^{o} is the noise-free output at time t, described by

$$y_t^{\text{o}} : \quad A(q^{-1}, \theta^{s_t}) y_t^{\text{o}} = B(q^{-1}, \theta^{s_t}) u_t, \tag{3b}$$

while v_t is a coloured noise affecting the output, given by

$$v_t : \quad D(q^{-1}, \theta^{s_t}) v_t = C(q^{-1}, \theta^{s_t}) e_t. \tag{3c}$$

The variable $s_t \in \mathcal{K} = \{1, \ldots, K\}$ in (3b)-(3c) denotes the (hidden) discrete state at time t, and e_t in (3c) is a zero-mean Gaussian random variable, generated by a white noise process with variance $\sigma_e^2 \in \mathbb{R}^+$. For a fixed mode $k \in \mathcal{K}$, $A(q^{-1}, \theta^k)$, $B(q^{-1}, \theta^k)$, $C(q^{-1}, \theta^k)$ and $D(q^{-1}, \theta^k)$ in (3b)-(3c) are polynomials in the shift operator q^{-1} (*i.e.,* $q^{-d}u_t = u_{t-d}$, for $d \in \mathbb{Z}$), defined as

$$A(q^{-1}, \theta^k) = 1 + \sum_{i=1}^{n_a} a_i^k q^{-i}, \tag{4a}$$

$$B(q^{-1}, \theta^k) = \sum_{i=1}^{n_b} b_i^k q^{-i}, \tag{4b}$$

$$C(q^{-1}, \theta^k) = 1 + \sum_{i=1}^{n_c} c_i^k q^{-i}, \tag{4c}$$

$$D(q^{-1}, \theta^k) = 1 + \sum_{i=1}^{n_d} d_i^k q^{-i}, \tag{4d}$$

where n_a, n_b, n_c and n_d indicate the dynamical order of the local BJ models, and θ^k is the collection of parameters characterizing the k-th sub-model, *i.e.,*

$$\theta^k = [a_1^k \ \cdots \ a_{n_a}^k \ b_1^k \ \cdots \ b_{n_b}^k \ c_1^k \ \cdots \ c_{n_c}^k \ d_1^k \ \cdots \ d_{n_d}^k]'. \tag{5}$$

Switches of the (hidden) discrete state s_t are supposed to be driven by a Markov chain with state transition matrix M. Therefore, for every $t \in \{1, \ldots, T\}$ the conditional probability of state s_t given the sequence of past modes $\mathcal{S}^{t-1} = \{s_\tau\}_{\tau=0}^{t-1}$ is defined as

$$p(s_t|\mathcal{S}^{t-1}) = p(s_t|s_{t-1}) = [M]_{s_{t-1},s_t}, \tag{6}$$

with

$$M_{i,j} \geq 0, \quad \sum_{j=1}^{K} M_{i,j} = 1, \quad i = 1, \ldots, K, \tag{7}$$

and $s_0 \in \mathcal{K}$ being the (unknown) initial discrete state.

Given the sets of inputs $\mathcal{U}^T$ and outputs $\mathcal{Y}^T$, in this paper we aim at estimating: the parameters $\Theta = [\,(\theta^1)' \cdots (\theta^K)'\,]' \in \mathbb{R}^{n_\Theta}$ of the BJ sub-models; the variance $\sigma_e^2 \in \mathbb{R}^+$ of the white noise e_t in (3c); the state transition matrix $M \in \mathbb{R}^{K\times K}$ and the discrete state sequence $\mathcal{S}^T = \{s_t\}_{t=0}^T$. This problem is addressed by assuming that the number and the order of the local models are known. Nonetheless, this hypothesis can be relaxed, *e.g.* by selecting the number K of sub-models via cross-validation.

III. PROBLEM FORMULATION

The problem of learning jump *Box-Jenkins* models is addressed by maximizing the joint posterior distribution $p(\Theta, \sigma_e^2, M, \mathcal{S}^T|\mathcal{Y}^T, \mathcal{U}^T)$ of the unknown parameters, given the datasets $\mathcal{Y}^T$ and $\mathcal{U}^T$. By using Bayes' rule, it is straightforward to see that the posterior distribution can be factorized, up to a proportional term, as

$$\begin{aligned} &p(\Theta, \sigma_e^2, M, \mathcal{S}^T|\mathcal{Y}^T, \mathcal{U}^T) \propto \\ &\propto p(\mathcal{Y}^T|\Theta, \sigma_e^2, M, \mathcal{S}^T, \mathcal{U}^T) p(\Theta, \sigma_e^2, M, \mathcal{S}^T|\mathcal{U}^T). \end{aligned} \tag{8}$$

The posterior, the joint distribution $p(\Theta, \sigma_e^2, M, \mathcal{S}^T|\mathcal{U}^T)$ and the likelihood $p(\mathcal{Y}^T|\Theta, \sigma_e^2, M, \mathcal{S}^T, \mathcal{U}^T)$ are defined as follows. For the mathematical details on the derivation of the posterior distribution, the reader is referred to [13, Section 3.3].

A. Priors

The joint probability distribution $p(\Theta, \sigma_e^2, M, \mathcal{S}^T | \mathcal{U}^T)$ is obtained according to the following priors.

1. The unknown variables $\Theta, \sigma_e^2, M, \mathcal{S}^T$ are *statistically independent* of the inputs $\mathcal{U}^T$, *i.e.,*
$$p(\Theta, \sigma_e^2, M, \mathcal{S}^T | \mathcal{U}^T) = p(\Theta, \sigma_e^2, M, \mathcal{S}^T). \tag{9}$$
2. The joint prior $p(\Theta, \sigma_e^2, M, \mathcal{S}^T)$ factorizes as
$$p(\Theta, \sigma_e^2, M, \mathcal{S}^T) = p(\Theta, \sigma_e^2) p(M, \mathcal{S}^T). \tag{10}$$
3. The joint probability distribution $p(\Theta, \sigma_e^2)$ is a *Gaussian Inverse-Gamma* with parameters $\lambda, \alpha_0, \beta_0 > 0$, so that
$$p(\Theta | \sigma_e^2) = \mathcal{N}(0, \sigma_e^2 \lambda^2 I_{n_\Theta}), \tag{11a}$$
$$p(\sigma_e^2) = \Gamma^{-1}(\alpha_0, \beta_0), \tag{11b}$$
where $\Gamma^{-1}(\alpha_0, \beta_0)$ denotes the *Inverse-Gamma* distribution with parameters α_0 and β_0, with probability density function
$$p(\sigma_e^2; \alpha_0, \beta_0) = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha)} (\sigma_e^2)^{-\alpha_0 - 1} e^{-\frac{\beta_0}{\sigma_e^2}}. \tag{11c}$$
4. The joint probability distribution $p(\mathcal{S}^T, M)$ factorizes as
$$p(\mathcal{S}^T, M) = p(\mathcal{S}^T | M) p(M), \tag{12}$$
with the components of the i-th row of M following a Dirichlet distribution with parameters $\alpha_1, \ldots, \alpha_K$ and probability density function
$$p(M_{i,1}, \ldots, M_{i,K}) = \frac{\Gamma(\alpha_1 + \ldots + \alpha_K)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \prod_{j=1}^{K} M_{i,j}^{\alpha_j - 1}, \tag{13}$$
where $p(M_{i,1}, \ldots, M_{i,K})$ is defined over the simplex defined by (7). Furthermore, the rows of the transitions matrix M are assumed to be statistically independent with each others, *i.e.*, $p(M) = \prod_{i=1}^{K} p(M_{i,1}, \ldots, M_{i,K})$, so that
$$p(M) = \left(\frac{\Gamma(\alpha_1 + \ldots + \alpha_K)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \right)^K \prod_{i,j=1}^{K} M_{i,j}^{\alpha_j - 1}. \tag{14}$$
Based on the modelling assumptions in (6), the probability distribution $p(\mathcal{S}^T | M)$ is thus equal to
$$\begin{aligned} p(\mathcal{S}^T | M) =& p(s_0) \prod_{t=1}^{T} p(s_t | s_{t-1}) = p(s_0) \prod_{t=1}^{T} M_{s_{t-1}, s_t} \\ =& p(s_0) \prod_{i,j}^{K} \prod_{t=1}^{T} M_{i,j}^{\mathbb{I}(s_{t-1} = i \text{ and } s_t = j)} \\ =& p(s_0) \prod_{i,j}^{K} M_{i,j}^{\#(s_{t-1} = i, s_t = j)}, \end{aligned} \tag{15}$$
where $\#$ counts the number of times the joint event $s_{t-1} = i$ and $s_t = j$ occurs in the sequence $\mathcal{S}^T$, *i.e.*,
$$\#(s_{t-1} = i, s_t = j) = \sum_{t=1}^{T} \mathbb{I}(s_{t-1} = i \text{ and } s_t = j),$$
and $p(s_0)$ is the probability of the initial state s_0. In order not to complicate the notation, $p(s_0)$ is assumed to be uniform, *i.e.*, $p(s_0) = \frac{1}{K}$ for all $s_0 = 1, \ldots, K$.

B. Likelihood

The *likelihood* $p(\mathcal{Y}^T | \Theta, \sigma_e^2, M, \mathcal{S}^T, \mathcal{U}^T)$ is computed by exploiting ideas taken from prediction-error methods for LTI systems. Accordingly, the output observation y_t can be factorized into the combination of a predictor $\hat{y}_{t|t-1}$ and a prediction error ε_t, *i.e.,*
$$y_t = \hat{y}_{t|t-1} + \varepsilon_t. \tag{16}$$

The prediction error ε_t is selected as
$$\varepsilon_t = e_t = H^{-1}(q^{-1}, \theta^{s_t}) \left(y_t - G(q^{-1}, \theta^{s_t}) u_t \right), \tag{17a}$$
and $G(q^{-1}, \theta^{s_t})$ and $H(q^{-1}, \theta^{s_t})$ are the following rational functions in q^{-1}:
$$G(q^{-1}, \theta^{s_t}) = \frac{B(q^{-1}, \theta^{s_t})}{A(q^{-1}, \theta^{s_t})}, \tag{17b}$$
$$H(q^{-1}, \theta^{s_t}) = \frac{C(q^{-1}, \theta^{s_t})}{D(q^{-1}, \theta^{s_t})}. \tag{17c}$$

Based on the definition of the polynomials in (4), the predictor $\hat{y}_{t|t-1}$ depends on past inputs/outputs only, while it is independent of e_t. Therefore, the *likelihood* is given by
$$\begin{aligned} &p(\mathcal{Y}^T | \Theta, \sigma_e^2, M, \mathcal{S}^T, \mathcal{U}^T) = \\ &= \frac{1}{(2\pi\sigma_e^2)^{T/2}} e^{-\frac{1}{2\sigma_e^2} \sum_{t=1}^{T} (H^{-1}(\theta^{s_t})(y_t - G(\theta^{s_t}) u_t))^2}. \end{aligned} \tag{18}$$

C. Posterior

The posterior in (8) thus becomes[1]
$$p(\Theta, \sigma_e^2, M, \mathcal{S}^T | \mathcal{Y}^T, \mathcal{U}^T) \propto \tag{19a}$$
$$\propto e^{-\frac{1}{2\sigma_e^2} \sum_{t=1}^{T} (H^{-1}(\theta^{s_t})(y_t - G(\theta^{s_t}) u_t))^2} \tag{19b}$$
$$\times \frac{1}{(\sigma_e^2)^{T/2 + n_\Theta/2 + \alpha_0 + 1}} e^{-\frac{1}{2\lambda^2\sigma_e^2} \Theta' \Theta} e^{-\frac{\beta_0}{\sigma_e^2}} \tag{19c}$$
$$\times \prod_{i,j=1}^{K} M_{i,j}^{\#(s_{t-1} = i, s_t = j) + \alpha_j + 1}. \tag{19d}$$

The maximum-a-posteriori estimation is thus given by the solution of problem
$$\max_{\Theta, \sigma_e^2, M, \mathcal{S}^T} p(\Theta, \sigma_e^2, M, \mathcal{S}^T | \mathcal{Y}^T, \mathcal{U}^T), \tag{20}$$
with $p(\Theta, \sigma_e^2, M, \mathcal{S}^T | \mathcal{Y}^T, \mathcal{U}^T)$ in (19).

[1] We have dropped the dependency of $G(q^{-1}, \theta^{s_t})$ and $H(q^{-1}, \theta^{s_t})$ on q^{-1} to reduce the notational burden.

Algorithm 1 *Maximum-a-posteriori* estimation for *Jump Box-Jenkins* models.

```
Input: Training sets $\mathcal{U}^T$ and $\mathcal{Y}^T$; initial guesses on the mode sequence $S^{T(0)} = (s_0^{(0)}, \dots, s_T^{(0)})$; tolerance $\epsilon_J > 0$; maximum number $h_{\max}$ of iterations.

1. iterate for $h = 1, \dots$
  1.1. Compute $\Theta^{(h)}, \sigma_e^{2(h)}, M^{(h)}$ as the solution of
         $\arg\max_{\Theta,\sigma_e^2,M} p(\Theta, \sigma_e^2, M, \mathcal{S}^{T(h-1)} | \mathcal{Y}^T, \mathcal{U}^T)$;
  1.2. Compute $\mathcal{S}^{T(h)}$ as the solution of
         $\arg\max_{\mathcal{S}^T} p(\Theta^{(h)}, \sigma_e^{2(h)}, M^{(h)}, \mathcal{S}^T | \mathcal{Y}^T, \mathcal{U}^T)$;
2. until $h = h_{\max}$ or
     $\Big| p(\Theta^{(h)}, \sigma_e^{2(h)}, M^{(h)}, \mathcal{S}^{T(h)} | \mathcal{Y}^T, \mathcal{U}^T)$
     $- p(\Theta^{(h-1)}, \sigma_e^{2(h-1)}, M^{(h-1)}, \mathcal{S}^{T(h-1)} | \mathcal{Y}^T, \mathcal{U}^T) \Big| \leq \epsilon_J$

Output: Estimated parameters $\Theta^\star = \Theta^{(h)}$, $\sigma_e^{2\star} = \sigma_e^{2(h)}$, $M^\star = M^{(h)}$ and mode sequence $\mathcal{S}^{T\star} = \mathcal{S}^{T(h)}$.
```

IV. Learning jump BJ models

Problem (20) is solved as outlined in Algorithm 1. At each iteration h, the method alternates between: (i) maximization of the posterior (19) over Θ, σ_e^2, M, for fixed mode sequence $\mathcal{S}^{T(h-1)}$ computed at the previous iteration (Step 1.1); (ii) maximization with respect to $\mathcal{S}^T$ (Step 1.2), for fixed parameters $\Theta^{(h)}, \sigma_e^{2(h)}, M^{(h)}$ obtained at Step 1.1.

Remark 1: The parameters $\Theta^\star, \sigma_e^{2\star}, M^\star$ and the mode sequence $\mathcal{S}^{T\star}$ obtained with Algorithm 1 depend on the initial guess $\mathcal{S}^{T(0)}$ of the mode sequence. Therefore, to improve the quality of the solution, Algorithm 1 can be run starting from N different initial sequences, selecting the outcome corresponding to the maximum posterior. ■

The procedure to solve the maximization problems in Steps 1.1 and 1.2 is described in the following paragraphs.

A. Step 1.1: optimization over Θ, σ_e^2 and M

For a fixed mode sequence $\mathcal{S}^{T(h-1)}$, the posterior distribution $p(\mathcal{Y}^T | \Theta, \sigma_e^2, M, \mathcal{S}^T, \mathcal{U}^T)$ in (19) can be separately optimized with respect to the unknowns Θ, σ_e^2 and M. Step 1.1 thus requires the solution of three problems, as described in the following sections[2].

1) Estimation of Θ : Based on the definition of the posterior distribution $p(\Theta, \sigma_e^2, M, \mathcal{S}^T | \mathcal{Y}^T, \mathcal{U}^T)$ in (19), its maximization over the parameters Θ (for fixed mode sequence $\mathcal{S}^{T(h)}$) can be performed by solving the equivalent optimization problem

$$\min_{\Theta} J(\Theta) \triangleq \sum_{t=1}^{T} \varepsilon_t(\Theta, \mathcal{S}^T)^2 + \frac{1}{\lambda^2}\Theta'\Theta. \tag{21}$$

According to (17), the time evolution of $\varepsilon_t(\Theta, \mathcal{S}^T)$ is given by

$$C(\theta^{s_t})\varepsilon_t(\Theta, \mathcal{S}^T) = D(\theta^{s_t})\left(y_t - y_t^{\mathrm{o}}\right), \tag{22}$$

where y_t^{o} is computed, for fixed parameters θ^{s_t}, as in (3b).

Problem (21) can be solved by means of any unconstrained nonlinear optimization method, such as Gauss-Newton. The gradient $\nabla_\Theta J(\Theta)$ of the cost is computed as

$$\nabla_\Theta J(\Theta) = 2\sum_{t=1}^{T} \varepsilon_t(\Theta, \mathcal{S}^T)\frac{\partial \varepsilon_t(\Theta, \mathcal{S}^T)}{\partial \Theta} + 2\lambda^{-2}\Theta, \tag{23a}$$

and its Hessian $\nabla_\Theta^2 J(\Theta)$ approximated as

$$\nabla_\Theta^2 J(\Theta) \approx 2\sum_{t=1}^{T} \frac{\partial \varepsilon_t(\Theta, \mathcal{S}^T)}{\partial \Theta}\left(\frac{\partial \varepsilon_t(\Theta, \mathcal{S}^T)}{\partial \Theta}\right)' + 2\lambda^{-2}I. \tag{23b}$$

Based on the time evolution of the prediction error $\varepsilon_t(\Theta, \mathcal{S}^T)$ in (22), its partial derivatives with respect to the parameters c_j^i (with $j = 1, \dots, n_c$) and d_j^i (with $j = 1, \dots, n_d$), for $i = 1, \dots, K$, can be computed by simulating the difference equations:

$$C(\theta^{s_t})\frac{\partial \varepsilon_t(\Theta, \mathcal{S}^T)}{\partial c_j^i} = -\varepsilon_{t-j}(\Theta, \mathcal{S}^T)\mathbb{I}_{[s_t = i]}, \tag{24a}$$

$$C(\theta^{s_t})\frac{\partial \varepsilon_t \Theta, \mathcal{S}^T)}{\partial d_j^i} = \left(y_{t-j} - y_{t-j}^{\mathrm{o}}\right)\mathbb{I}_{[s_t = i]}, \tag{24b}$$

with y_t^{o} simulated according to (3b).

The remaining derivatives, namely $\frac{\partial \varepsilon_t(\Theta)}{\partial a_j^i}$ (with $j = 1, \dots, n_a$) and $\frac{\partial \varepsilon_t(\Theta)}{\partial b_j^i}$ (with $j = 1, \dots, n_b$) are obtained by simulating the difference equations

$$C(\theta^{s_t})\frac{\partial \varepsilon_t(\Theta, \mathcal{S}^T)}{\partial a_j^i} = -D(\theta^{s_t})\frac{\partial y_t^{\mathrm{o}}}{\partial a_j^i}, \tag{25a}$$

$$C(\theta^{s_t})\frac{\partial \varepsilon_t(\Theta, \mathcal{S}^T)}{\partial b_j^i} = -D(\theta^{s_t})\frac{\partial y_t^{\mathrm{o}}}{\partial b_j^i}, \tag{25b}$$

for $i = 1, \dots, K$, with $\frac{\partial y_t^{\mathrm{o}}}{\partial a_j^i}$ and $\frac{\partial y_t^{\mathrm{o}}}{\partial b_j^i}$ given by the difference equations

$$A(\theta^{s_t})\frac{\partial y_t^{\mathrm{o}}}{\partial a_j^i} = y_{t-j}^{\mathrm{o}}\mathbb{I}_{[s_t = i]}, \quad A(\theta^{s_t})\frac{\partial y_t^{\mathrm{o}}}{\partial b_j^i} = u_{t-j}\mathbb{I}_{[s_t = i]}. \tag{26}$$

The reader is referred to [13, Section 4.1.1] for details on the derivation of eqs. (24), (25) and (26).

2) Estimation of the variance σ_e^2: The parameter σ_e^2 maximizing the posterior $p(\Theta, \sigma_e^2, M, \mathcal{S}^T | \mathcal{Y}^T, \mathcal{U}^T)$ can be computed analytically. Indeed, given $\Theta^{(h)}$, the value of σ_e^2 maximizing $p(\Theta, \sigma_e^2, M, \mathcal{S}^T | \mathcal{Y}^T, \mathcal{U}^T)$ is given by:

$$\sigma_e^{2(h)} = \frac{\beta_0 + \frac{1}{2}\lambda^{-2}\Theta^{(h)'}\Theta^{(h)} + \frac{1}{2}\sum\limits_{t=1}^{T}\left(\varepsilon_t(\Theta^{(h)}, \mathcal{S}^T)\right)^2}{\frac{T + n_\Theta}{2} + \alpha_0 + 1}. \tag{27}$$

3) Estimation of the transition probabilities: The posterior in (19) can be separately maximized with respect to each row of the transition matrix M. This problem is solved by optimizing the *log* of (19) with the method of *Lagrange multipliers*, in order to take into account the equality constraints in (7). As shown in detail in [13, Section 4.1.2], the following closed-form expression for the transition matrix M maximizing the posterior is obtained:

[2] To simplify the notation, we have dropped the dependence of $\mathcal{S}^{T(h-1)}$ and$s_t^{(h-1)}$ on the algorithm's iteration $h-1$.

$$M_{i,j} = \frac{\#(s_{t-1}=i, s_t=j)+\alpha_j-1}{\sum_{j=1}^{K}\#(s_{t-1}=i, s_t=j)+\alpha_j-1} = \\ = \frac{\#(s_{t-1}=i, s_t=j)+\alpha_j-1}{\sum_{j=1}^{K}\#(s_{t-1}=i, s_t=j)+\sum_{j=1}^{K}(\alpha_j-1)}. \quad (28)$$

The maximum-a-posterior estimate of the transition probabilities $M_{i,j}$ is thus given by the sample transition probability (*i.e.,* the ratio between the number of switches from mode i to mode j and the number of times mode i is active), up to the additive term $\alpha_j - 1$ introduced due to the chosen priors on $M_{i,j}$ (eq. (13)).

B. Step 1.2: optimization over $\mathcal{S}^T$

Once the parameters $\Theta^{(h)}, \sigma_e^{2(h)}, M^{(h)}$ are updated, the posterior distribution $p(\Theta, \sigma_e^2, M, \mathcal{S}^T|\mathcal{Y}^T, \mathcal{U}^T)$ in (19) is maximized with respect to the mode sequence $\mathcal{S}^T$ by considering the log of the posterior and neglecting all the terms that are independent of $\mathcal{S}^T$. This leads to the maximization of the following objective

$$Q_T\left(\mathcal{S}^T\right) = \sum_{t=1}^{T}\mathcal{L}_t(\mathcal{S}^t) + \sum_{t=1}^{T}\mathcal{L}^{trans}(s_{t-1}, s_t), \quad (29a)$$

with

$$\mathcal{L}_t(\mathcal{S}^t) = -\frac{1}{2\sigma_e^{2(h)}}\varepsilon_t(\Theta^{(h)}, \mathcal{S}^t)^2, \quad (29b)$$

$$\mathcal{L}^{trans}(s_{t-1}, s_t) = \sum_{i,j=1}^{K}\mathbb{I}_{[s_{t-1}=i, s_t=j]}\log M_{i,j}^{(h)}. \quad (29c)$$

The objective (29a) is maximized w.r.t. the whole sequence $\mathcal{S}^T$ through a suboptimal moving-horizon approach, that allows us to optimize $Q_T\left(\mathcal{S}^T\right)$ without the need to consider all possible past modal histories. By considering a moving window of T_c time instants, and defining the truncated objective function

$$Q_t\left(\mathcal{S}^t\right) = \sum_{\tau=1}^{t}\mathcal{L}_\tau(\mathcal{S}^\tau) + \sum_{\tau=1}^{t}\mathcal{L}^{trans}(s_{\tau-1}, s_\tau), \quad (30)$$

the proposed suboptimal method consists in the following steps.

Initially, the optimal initial mode $\tilde{s}_0(\mathcal{S}_1^{T_c})$ for every possible T_c-length sequence $\mathcal{S}_1^{T_c}$ is computed, *i.e.,*

$$\tilde{s}_0(\mathcal{S}_1^{T_c}) = \arg\,\max\nolimits_{s_0} Q_{T_c}(\mathcal{S}_0^{T_c}), \quad (31)$$

with $\mathcal{S}_0^{T_c}$ denoting the sequence $\{s_t\}_{t=0}^{T_c}$.

Then, for $t = 1, \ldots, T - T_c - 1$, the optimal mode $\tilde{s}_t$ associated to each possible sequence $\mathcal{S}_{t+1}^{T_c+t}$ is computed as

$$\tilde{s}_t(\mathcal{S}_{t+1}^{T_c+t}) = \arg\,\max\nolimits_{s_t} Q_{Tc+t}(\tilde{\mathcal{S}}_0^{T_c+t}(\mathcal{S}_t^{T_c+t})), \quad (32)$$

where the modes $s_0, \ldots, s_{t-1}$ are fixed to the optimal modes computed at previous steps, *i.e.,*

$$\tilde{\mathcal{S}}_0^{T_c+t}(\mathcal{S}_t^{T_c+t}) = \{\tilde{s}_0(\tilde{\mathcal{S}}_1^{T_c}), \tilde{s}_1(\tilde{\mathcal{S}}_2^{T_c+1}), \ldots, \\ \tilde{s}_{t-1}(\tilde{\mathcal{S}}_t^{T_c+t-1}), s_t, s_{t+1}, \ldots, s_{t+T_c}\}. \quad (33)$$

Then, at time $t = T - T_c$, the optimal sequence $(\mathcal{S}_{T-T_c}^T)^\star$ is computed as

$$(\mathcal{S}_{T-T_c}^T)^\star = \arg\,\max\nolimits_{\mathcal{S}_{T-T_c}^T} Q_T\left(\tilde{\mathcal{S}}_0^{T-T_c}(\mathcal{S}_{T-T_c}^T)\right). \quad (34)$$

TABLE I

TRUE VS ESTIMATED PARAMETERS

		True	JBJ	JARX
a_1^j	$j=1$	0.60	0.59	0.41
	$j=2$	-0.50	-0.53	-0.48
a_2^j	$j=1$	0.10	0.09	0.06
	$j=2$	-0.10	-0.09	-0.19
b_1^j	$j=1$	0.80	0.80	0.80
	$j=2$	-0.20	-0.19	-0.20
b_2^j	$j=1$	-0.80	-0.82	-0.94
	$j=2$	0.50	0.50	0.49
c_1^j	$j=1$	0.60	0.63	-
	$j=2$	0.20	0.15	-
c_2^j	$j=1$	0.20	0.24	-
	$j=2$	0.20	0.19	-
d_1^j	$j=1$	0.10	0.11	-
	$j=2$	-0.25	-0.30	-
d_2^j	$j=1$	0.70	0.72	-
	$j=2$	-0.25	-0.24	-

Finally, starting from $(\mathcal{S}_{T-T_c}^T)^\star$, the rest of the mode sequence is estimated as

$$(\mathcal{S}_0^{T-T_c-1})^\star = \tilde{\mathcal{S}}_0^{T-T_c}((\mathcal{S}_{T-T_c}^T)^\star). \quad (35)$$

Additional details on the proposed suboptimal moving approach can be found in [13, Section 4.2]. The length of the horizon T_c is a tuning parameter, that allows us to trade off between the complexity of the problem to be solved and the sub-optimality of the estimated mode sequence.

V. SIMULATION EXAMPLE

The proposed approach is tested on a simple simulation example, where the data are generated by a jump Box-Jenkins system described as in (3), with $K = 2$ modes. For a more exhaustive assessment of the performance of the proposed approach, the reader is referred to [13, Section 5]. The Box-Jenkins local models are defined by second-order linear time-invariant systems (*i.e.,* $n_a = n_b = n_c = n_d = 2$), with coefficients reported in Table I. The data available for testing are obtained by exciting the system with a sequence of $T = 10,000$ randomly generated inputs, uniformly distributed in the interval $[-1,\ 1]$. The mode switches every 100 samples starting from $s_0 = 2$. The output is corrupted by a zero mean Gaussian noise e_t with standard deviation $\sigma_e = 0.3$, which yields the *Signal-to-Noise-Ratio* (SNR)

$$\text{SNR} = 10\log\frac{\sum_{t=1}^{\tilde{T}}\left(y_t - e_t\right)^2}{\sum_{t=1}^{\tilde{T}} e_t^2} = 8.6\text{ dB}. \quad (36)$$

Algorithm 1 is run by setting the parameters of the *Gaussian Inverse-Gamma* prior distribution in (11) to $\lambda = 10^{-8}$ and $\alpha_0 = \beta_0 = 1$, while the parameters α_k of the Dirichlet prior distribution (13) are set to $\alpha_k = 1$, for all $k = 1, \ldots, K$. The algorithm is executed $N = 5$ times using different randomly generated initial guesses $\mathcal{S}^{T(0)}$ and, for each initial guess, it is iterated until either the maximum number of iterations $h_{max} = 20$ is reached or the termination condition $|V^{(h)} - V^{(h-1)}| \leq \epsilon_V$ is verified, with $V^{(h)}$ being the log of the posterior distribution (19) and $\epsilon_V = 10^{-8}$. In reconstructing the mode sequence with the approach described in Section IV-B, we consider $T_c = \max\left(n_a, n_b, n_c, n_d\right) = 2$.

The estimated standard deviation $\sigma_e^\star$ is equal to 0.37, while the estimated parameters are reported in Table I, along with

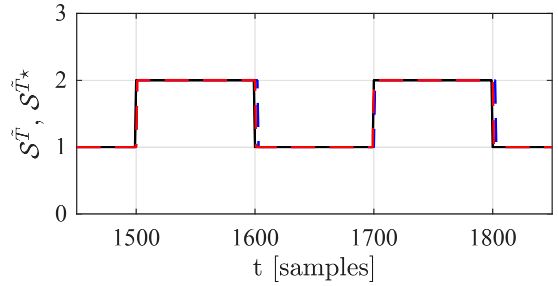

Fig. 1. True $\mathcal{S}^{\tilde{T}}$ (black) vs estimated $\mathcal{S}^{T,\star}$ mode sequence. Estimate with: jump BJ model (red), ARX model (blue). Black and red lines are overlapped.

the true coefficients Θ, showing that both the characteristics of the noise and the local models are accurately reconstructed. The computational time required to run Algorithm 1 for $N=5$ is around 30 minutes. The parameters of a jump ARX (JARX) model, which are identified using the approach proposed in the paper for $C(q^{-1},\theta^i)=1$ and $A(q^{-1},\theta^i)=D(q^{-1},\theta^i)$, for $i=1,2$, are also reported in Table I. These estimates clearly show a bias due to the inconsistent noise model structure. The quality of the estimated sequence $\mathcal{S}^{T\star}$ is measured in terms of the accuracy index

$$L_T^{true} = \frac{100}{T}\sum_{t=1}^{T} \mathbb{I}_{[s_t^\star = s_t]}, \tag{37}$$

by using the true mode sequence $\mathcal{S}^T$ as ground-truth. On the training set, we achieve $L_T^{true}=99.7\%$ and $L_T^{true}=99.5\%$ for the jump BJ and the jump ARX model, respectively. Therefore, in the considered example, a proper choice of the sub-models' structure mainly affects the quality of the estimated local models. The identified jump models (JBJ and JARX) are further used to reconstruct the mode sequence as described in Section IV-B for a validation set of length $\tilde{T}=5,000$, comprising data generated by a starting mode $s_0=1$, with input and noise sequences different from the ones used for training. The attained label accuracy indexes are $L_{\tilde{T}}^{true}=99.0\%$ and $L_{\tilde{T}}^{true}=98.7\%$ with the jump BJ and jump ARX models, respectively. Even if similar results are achieved, Fig. 1 shows that the use of a jump ARX model causes slight delays in the detection of mode switches.

The robustness of the learning method is assessed by performing a Monte Carlo simulation with 30 random realizations of the initial state $s_0 \in \{1,2\}$, the input and the noise e_t. The mean values and standard deviations of the estimated parameters are reported in Table II, showing that the true parameters lie within the uncertainty intervals defined by the standard deviation.

VI. Conclusions

We have described a novel method for the identification of jump Box-Jenkins models, that relies on the derivation of the posterior distribution of all the unknown parameters defining the jump Box-Jenkins model. The posterior distribution is maximized by combining an extension of the prediction error method tailored to BJ models with switching coefficients and a suboptimal moving-horizon approach, used to retrieve the mode sequence with limited computational complexity.

TABLE II

Monte Carlo simulation: True vs estimated parameters (mean ± standard deviation)

		True	Estimated (JBJ)
a_1^j	$j=1$	0.60	0.60 ± 0.01
	$j=2$	-0.50	-0.50 ± 0.01
a_2^j	$j=1$	0.10	0.10 ± 0.01
	$j=2$	-0.10	-0.09± 0.01
b_1^j	$j=1$	0.80	0.80± 0.01
	$j=2$	-0.20	-0.20± 0.01
b_2^j	$j=1$	-0.80	-0.80± 0.02
	$j=2$	0.50	0.50± 0.01
c_1^j	$j=1$	0.60	0.60 ± 0.02
	$j=2$	0.20	0.21 ± 0.05
c_2^j	$j=1$	0.20	0.20 ± 0.02
	$j=2$	0.20	0.20 ± 0.03
d_1^j	$j=1$	0.10	0.10 ± 0.02
	$j=2$	-0.25	-0.24 ± 0.05
d_2^j	$j=1$	0.70	0.70 ± 0.01
	$j=2$	-0.25	-0.25 ± 0.04

Extensions of this work include the analysis of the statistical properties of the estimated models, derivation of the uncertainty intervals of the estimated parameters, and autotuning of the number of possible operating modes.

References

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

[2] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.

[3] A. Bemporad, V. Breschi, D. Piga, and S. Boyd. Fitting jump models. *Automatica*, 96:11 – 21, 2018.

[4] V. Breschi, A. Bemporad, D. Piga, and S. Boyd. Prediction error methods in learning jump armax models. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2247–2252, Dec 2018.

[5] V. Breschi, D. Piga, and A. Bemporad. Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica*, 73:155–162, 2016.

[6] O.L.V. Costa, M.D. Fragoso, and R.P. Marques. *Discrete-Time Markov Jump Linear Systems*. Probability and Its Applications. Springer London, 2006.

[7] E. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4):1569–1585, 2011.

[8] A. Goudjil, M. Pouliquen, E. Pigeon, and O. Gehan. Identification algorithm for mimo switched output error model in presence of bounded noise. In *IEEE 56th Annual Conference on Decision and Control*, pages 5286–5291, 2017.

[9] A. Juloski, W.P.M.H. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, and J.H.G. Niessen. Comparison of four procedures for the identification of hybrid systems. In *HSCC*, volume 3414 of *Lecture Notes in Computer Science*, pages 354–369. Springer, 2005.

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

[11] M. Ostendorf, V. V. Digalakis, and O. A. Kimball. From HMM's to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Transactions on speech and audio processing*, 4(5):360–378, 1996.

[12] N. Ozay, C. Lagoa, and M. Sznaier. Set membership identification of switched linear systems with known number of subsystems. *Automatica*, 51:180–191, 2015.

[13] D. Piga, V. Breschi, and A. Bemporad. Maximum-a-posteriori estimation of jump box-jenkins models. Technical report, 2019. `https://www.dariopiga.com/TR/TRIDSIA1903.pdf`.

[14] F. Rosenqvist and A. Karlström. Realisation and estimation of piecewise-linear output-error models. *Automatica*, 41(3):545–551, 2005.

[15] F.D. Torrisi and A. Bemporad. Hysdel-a tool for generating computational hybrid models for analysis and synthesis problems. *IEEE Transactions on Control Systems Technology*, 12(2):235–249, March 2004.