

## Neural network-based identification of state-space switching nonlinear systems

Questa è la versione preprint della seguente opera:

*Original*

Neural network-based identification of state-space switching nonlinear systems / Zhang, Yanxin; Yu, Chengpu; Ferrer, Gonzalo; Fabiani, Filippo. - In: IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - ISSN 0018-9286. - (2026). [10.1109/TAC.2026.3669446]

*Availability:*

This version is available at: 20.500.11771/38658

*Publisher:*

IEEE

*Published*

DOI:10.1109/TAC.2026.3669446

*Terms of use:*

This publication is made accessible in accordance with the terms for deposit in the institutional repository, as defined by the IMT School for Advanced Studies Lucca's Open Access Policy. ([https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib\\_0.pdf](https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib_0.pdf)).

Si prega di consultare le pagine informative dell'editore relative alle politiche di autoarchiviazione.

(Article begins on next page)

# Neural network-based identification of state-space switching nonlinear systems

Yanxin Zhang, Chengpu Yu and Filippo Fabiani

**Abstract**—We design specific neural networks (NNs) for the identification of switching nonlinear systems in the state-space form, which explicitly model the switching behavior and address the inherent coupling between system parameters and switching modes. This coupling is specifically addressed by leveraging the expectation-maximization (EM) framework. In particular, our technique will combine a moving window approach in the E-step to efficiently estimate the switching sequence, together with an extended Kalman filter (EKF) in the M-step to train the NNs with a quadratic convergence rate. Extensive numerical simulations, involving both academic examples and a battery charge management system case study, illustrate that our technique outperforms available ones in terms of parameter estimation accuracy, model fitting, and switching sequence identification.

**Index Terms**—Switching systems, Neural network, Expectation maximization, Extend Kalman filter, nonlinear system identification.

## I. INTRODUCTION

In several engineering applications, such as speech recognition [33], financial [14], [35], [36], and robotic systems [8], [32], the occurrence of mode switching is a pervasive feature characterizing dynamical systems. To effectively analyze and control these systems, it is then vital to investigate their internal structures and develop models that accurately represent their behavior. The process of modeling such systems, however, often entails the challenging task of parameter identification across various subsystems, as well as the estimation of the system's operational modes at any given time instant [7], [9].

Remarkably, the active modes of a switching system are often coupled with the parameters of its subsystems, a fact that complicates the identification of the active subsystem at each time instant and the estimation of the switching behaviors. Current methodologies treat the switching mode sequence as implicit discrete states, clustering the temporal segments to obtain a collection of temporal partitions belonging to a subsystem [2], [11]. Subsequently, the parameters of the subsystems are identified using observed input-output data [22], [24].

This work was supported by the National Natural Science Foundation of China (Grant No. 62088101, 62473046), Chongqing Natural Science Foundation CSTB2023NSCQ-JQX0018, and Beijing Natural Science Foundation L221005. (Corresponding author: Chengpu Yu).

Yanxin Zhang is with School of Automation, Beijing Institute of Technology, Beijing 100081, PR China (e-mail: zhangyanxin@bit.edu.cn).

Chengpu Yu is with School of Automation, Beijing Institute of Technology, Beijing 100081, PR China (e-mail: yuchengpu@bit.edu.cn).

Filippo Fabiani is with IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100, Lucca, Italy (e-mail: filippo.fabiani@imtlucca.it).

## A. Related work

It is well-known that the identification of switching systems amounts to an NP-hard problem in general [21], [30]. Nevertheless, several methods have been proposed to address the typical identification issues for switching systems [27], [29], [38], [39]. Specifically, in [12] the authors survey the methodologies for identifying switching systems, also in the form of piecewise-affine (PWA) models, that have been developed during the last decade. More recently, some alternating identification methods have been proposed for jump models [6], [25], [38]. These approaches leverage the Bayesian theory to maximize the posterior probability function, thereby obtaining estimates of the switching sequence and the parameters of the subsystems. Available works on linear parameter varying (LPV) systems identification [13], [20], [26] employ a data-driven approach to model the LPV system with the influence of noise. All the abovementioned works, however, focus on switching linear or PWA systems, while to the best of our knowledge the literature on switching nonlinear systems identification, especially in state-space form, is relatively poor.

Neural networks (NNs) serve as an effective means for modeling nonlinear systems and have been widely applied across various domains. In [4], a multi-layer NN was applied to model PWA systems, while in [10] it has been shown that the same class of models can be identified by combining an OptNet layer [1] in cascade with a buffer one. In both approaches, the resulting NN could then be trained using back-propagation algorithms. For systems with (unknown) hidden states, some useful methods are introduced for learning the nonlinear state-space models in [23], [28]. However, these methods rely on gradient-based algorithms, which are renowned for their slow convergence rate. Inspired by the fresh look of extended Kalman filter (EKF) in [18], a NN training algorithm based on the EKF is proposed in [5]. However, these methods model the system using a NN as a black box, which can result in the loss of crucial internal information. For instance, when the system is composed of multiple subsystems, and only one of them is active at each time instant, the aforementioned approaches fail to capture the switching behavior of the system. Consequently, it is key to develop an algorithmic framework capable of modeling switching nonlinear systems without losing the information on their switching behavior.

## B. Summary of contribution and paper organization

We devise a NN-based method for the identification of state-space switching nonlinear systems. Our method is developed

in the expectation-maximization (EM) framework [3], thus consisting of two parts. In the E-step, the switching sequence is estimated by using a moving window approach. In the M-step, an EKF is used to obtain the estimation of the parameters of each subsystem.

Our main contribution can be summarized as follows:

- 1) We design a NN-based model able to represent nonlinear switching systems in the state-space form.
- 2) The Markov property possessed by the system usually makes the switching sequence estimation computationally intensive [3]. We adopt a moving window approach to drastically reduce such a computational burden.
- 3) Given the time-consuming training of standard NNs and their lack of robustness to noise [15], [19], [31], we develop an EKF-based NN training technique featuring a quadratic convergence rate.

The performance of our method is finally tested through extensive numerical experiments on both academic examples and a real-world battery charge management system case study, which also aim at evaluating the efficiency of our EKF-based training scheme.

The rest of the paper is organized as follows: in §II we describe the system and formalize the problem considered. In §III we discuss the identification method based on the EM framework. Specifically, §III-A gives the maximization step of the parameters in each subsystems by using the EKF-based method, while §III-B describes the expectation step based on a moving window approach to obtain the estimation of the switching sequence. The convergence analysis is then given in §III-C. Numerical simulations are discussed in §IV. Finally, the conclusion and future work are given in §V. The proofs of the main technical results derived in the paper are deferred to Appendix A.

### Notation

$\mathbb{N}$ ,  $\mathbb{Z}$ , and  $\mathbb{R}$  denote the set of natural, integer and real numbers, respectively. We indicate the extended real numbers as  $\bar{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$ . Given a matrix  $X$ ,  $\|X\|$  denotes its spectral norm,  $\text{tr}(X)$  the trace, and  $\text{vec}(X)$  the column vectorization of  $X$ . The operator  $\text{diag}(\cdot)$  produces a diagonal matrix with entries as its arguments.  $\mathbb{P}[\cdot]$  and  $\mathbb{E}[\cdot]$  respectively denote a probability distribution and the related expected value.  $\mathbb{P}_\theta[\cdot]$  and  $\mathbb{E}_\theta[\cdot]$  respectively denote the probability and expectation under the parameter  $\theta$ . For a sequence  $\{s_t\}_{t \in \mathbb{N}}$ ,  $s_T = \mathcal{O}(T)$  indicates that  $\limsup_{T \rightarrow \infty} s_T/T < \infty$ .  $\mathbb{I}(\cdot, \cdot)$  denotes the standard indicator function, i.e.,  $\mathbb{I}(s(i), \hat{s}(i)) = 1$  if  $s(i) = \hat{s}(i)$ , 0 otherwise.  $I$  identifies a standard identity matrix,  $N(\mu, \sigma^2)$  denotes the normal distribution of a random variable with mean  $\mu$  and standard deviation  $\sigma$ .

## II. MATHEMATICAL FORMULATION

### A. Dynamical system description

In this paper we will consider switching nonlinear systems in the following state-space representation:

$$x(t+1) = f_{s_t}(x(t), u(t), \theta_{x,s_t}) + \zeta(t), \quad (1)$$

$$y(t) = g_{s_t}(x(t), u(t), \theta_{y,s_t}) + \xi(t), \quad (2)$$

where  $t \in \mathbb{Z}$  is the discrete-time index,  $x \in \mathbb{R}^{n_x}$  represents the vector of state variables,  $u \in \mathbb{R}^{n_u}$  is the control input, and  $y \in \mathbb{R}^{n_y}$  denotes the system output. The scalar variable  $s_t \in \{1, \dots, K\}$  denotes the hidden active mode at time  $t$ , which selects some  $f_{s_t}$  and  $g_{s_t}$  within a collection of  $K$  nonlinear submodels. Specifically, for each  $s_t \in \{1, \dots, K\}$  we have  $f_{s_t} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_{\theta_x}} \rightarrow \mathbb{R}^{n_x}$  and  $g_{s_t} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_{\theta_y}} \rightarrow \mathbb{R}^{n_y}$ , where  $\theta_{x,s_t} \in \mathbb{R}^{n_{\theta_x}}$  and  $\theta_{y,s_t} \in \mathbb{R}^{n_{\theta_y}}$  denote the parameters characterizing  $f_{s_t}$  and  $g_{s_t}$  at time instant  $t$ , respectively. Finally,  $\zeta(t) \in \mathbb{R}^{n_x}$  and  $\xi(t) \in \mathbb{R}^{n_y}$  represent the process and the measurement noises, respectively.

*Standing Assumption 1:* For all  $t \in \mathbb{Z}$ ,  $\zeta(t) \sim N(0, \Sigma_1(t))$  and  $\xi(t) \sim N(0, \Sigma_2(t))$ .

### B. A recurrent neural network model

Recurrent neural networks (RNNs) are usually designed for processing sequential data, and are characterized by their directional cyclic structure that enables to retain and utilize past information, making them appropriate for modeling causal dynamical systems [40].

In the considered framework, we then propose to model the nonlinear, yet unknown, functions in (1) by means of tailored RNNs. Specifically, the state functions  $f(\cdot)$  can be modeled by the following specific RNN:

$$\begin{aligned} h_f^1 &= W_f^1 \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + b_f^1, \\ h_f^2 &= W_f^2 f^1(h_f^1) + b_f^2, \\ &\vdots \\ x(t+1) &= W_f^{l_f} f^{l_f-1}(h_f^{l_f-1}) + b_f^{l_f}. \end{aligned} \quad (3)$$

Similarly, the output function  $g(\cdot)$  turns into:

$$\begin{aligned} h_g^1 &= W_g^1 \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} + b_g^1, \\ h_g^2 &= W_g^2 g^1(h_g^1) + b_g^2, \\ &\vdots \\ h_g^{l_g} &= W_g^{l_g} g^{l_g-1}(h_g^{l_g-1}) + b_g^{l_g}, \\ y(t) &= g^{l_g}(h_g^{l_g}), \end{aligned} \quad (4)$$

where  $l_f, l_g > 0$  represent the number of hidden layers of the state and output functions, respectively,  $h_f^i, i = 1, \dots, l_f$  and  $h_g^i, i = 1, \dots, l_g$  denote the output of the  $i$ -th layer within the RNN, which is then used as input to the  $(i+1)$ -th layer,  $f^i, i = 1, \dots, l_f$  and  $g^i, i = 1, \dots, l_g$  are the corresponding activation functions, such as  $\tanh(\cdot)$  or rectified linear unit (ReLU). Finally,  $W_f^i, i = 1, \dots, l_f$  and  $W_g^i, i = 1, \dots, l_g$  are the weight matrices for the corresponding layer with appropriate dimensions,  $b_f^i, i = 1, \dots, l_f$  and  $b_g^i, i = 1, \dots, l_g$  are the associated vector of bias terms.

*Lemma 1:* (Universal approximation theorem [16], [17]) The standard multilayer feedforward network with as few as a single hidden layer, and arbitrary bounded and nonconstant activation functions is a universal approximator with respect (w.r.t.) to any given continuous function.

*Remark 1:* From Lemma 1, a standard multilayer feed-forward network can universally approximate any function. Given that RNN activation functions are a set of simple functions, any complex continuous nonlinear function  $f(\cdot)$  can be uniformly approximated by polynomial functions. Thus, the RNN with structure in (3) can approximate any continuous function through its activation functions.

Let us then collect the training parameters as  $\theta_f := \{(W_f^i, b_f^i)\}_{i=1}^{l_f}$ , and  $\theta_g := \{(W_g^i, b_g^i)\}_{i=1}^{l_g}$ . According to Lemma 1, the RNNs in (3)–(4) with parameters  $\theta_f$  and  $\theta_g$  can be used to describe the nonlinear functions in (1) and (2). To simplify notation, we let  $\mathcal{N}_x$  and  $\mathcal{N}_y$  denote the RNNs with structure in (3) and (4), respectively.

By making use of the introduced RNNs, we can then rewrite the system in (1) and (2) as follows

$$\begin{aligned} x(t+1) &= \mathcal{N}_{x,s_t}(x(t), u(t), \theta_{f,s_t}) + \zeta(t), \\ y(t) &= \mathcal{N}_{y,s_t}(x(t), u(t), \theta_{g,s_t}) + \xi(t), \end{aligned} \quad (5)$$

where  $\mathcal{N}_{x,s_t}$  and  $\mathcal{N}_{y,s_t}$  are two groups of RNNs, each of them containing  $K$ -RNNs. Accordingly,  $\theta_{f,s_t}$  and  $\theta_{g,s_t}$  are the associated NN parameters.

### C. Problem description

Assume that we are able to collect a dataset consisting of  $T$  samples of the system input and output (not necessarily a  $T$ -long trajectory), stacked together as  $\mathbf{y} = \mathbf{y}_{1:T} := \{y(1), \dots, y(T)\}$  and  $\mathbf{u} = \mathbf{u}_{1:T} := \{u(1), \dots, u(T)\}$ . To establish our main technical results on the identification of the switching nonlinear state-space system in (1)–(2), we make use of the following assumptions:

*Standing Assumption 2:* For any  $T > 0$ , the switching sequence  $\mathbf{S} := \{s_1, \dots, s_T\}$ , the system parameters  $\Theta := \{\theta_{f,1}, \dots, \theta_{f,K}, \theta_{g,1}, \dots, \theta_{g,K}\}$ , and the system inputs  $\mathbf{u}$  are all independent among them, i.e.,

$$\mathbb{P}[\mathbf{S}|\Theta, \mathbf{u}] = \mathbb{P}[\mathbf{S}], \text{ and } \mathbb{P}[\Theta|\mathbf{S}, \mathbf{u}] = \mathbb{P}[\Theta].$$

*Standing Assumption 3:* The switching sequence satisfies the Markov property, i.e., for any  $t \in \{1, \dots, T\}$ ,

$$\mathbb{P}[s_t|s_{t-1}, \dots, s_1] = \mathbb{P}[s_t|s_{t-1}] = \pi_{s_t, s_{t-1}}.$$

Note that the aforementioned assumptions have simplified the complexity of the system, while possibly imposing restrictions. For instance, Standing Assumption 2 implies that there is no correlation between the system parameters and the inputs. Similarly, the Markov property (i.e., Standing Assumption 3) assumes that the switching sequence depends solely on the previous state, without considering the influence of a longer time span. Similar conditions, however, have already been postulated in the cognate literature – see, e.g., [6], [25].

In view of Standing Assumption 3, the switching probability can then be characterized using a mode transition matrix  $\Pi \in \mathbb{R}^{K \times K}$ , which can be defined as follows:

$$\Pi = \begin{bmatrix} \pi_{1,1} & \pi_{1,2} & \cdots & \pi_{1,K} \\ \pi_{2,1} & \pi_{2,2} & \cdots & \pi_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{K,1} & \pi_{K,2} & \cdots & \pi_{K,K} \end{bmatrix},$$

and amounts to a row stochastic matrix since it satisfies

$$\sum_{i=1}^K \pi_{i,j} = 1, \text{ for all } j = 1, \dots, K. \quad (6)$$

By making use of the available data samples  $\mathbf{u}$  and  $\mathbf{y}$  collected from (1)–(2), our goal is thus to determine  $\mathcal{N}_x$  and  $\mathcal{N}_y$  in (5) with a known number of system modes  $K$ .

To this end, given the model parameters  $\Theta \in \mathbb{R}^d$ , a suitable cost function to be minimized for the identification of the switching nonlinear systems consists of three parts: a pure loss function  $\ell : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \times \mathbb{R}^d \times \{1, \dots, K\}^T \rightarrow \mathbb{R}$ , a regularization term for the parameters  $r : \mathbb{R}^d \rightarrow \mathbb{R}$ , and a loss involving the mode sequence  $\mathcal{L} : \{1, \dots, K\}^T \rightarrow \mathbb{R}$ :

$$J(\mathbf{y}, \mathbf{u}, \Theta, \mathbf{S}) = \ell(\mathbf{y}, \mathbf{u}, \Theta, \mathbf{S}) + r(\Theta) + \mathcal{L}(\mathbf{S}) \quad (7)$$

Under Standing Assumption 3, from [6] we note that  $\mathcal{L}$  reads as:  $\mathcal{L}(\mathbf{S}) = \log \pi_{s_0} + \sum_{i=1}^T \log \pi_{s_i, s_{i-1}}$ .

We then aim at determining the weights of the RNNs in (3)–(4) by minimizing  $J(\mathbf{y}, \mathbf{u}, \Theta, \mathbf{S})$  w.r.t.  $\Theta$ , transition matrix  $\Pi$  and switching sequence  $\mathbf{S}$ . Specifically, we will make use of the EM algorithm, an iterative method yielding an estimate of the underlying decision variables at each iteration, as detailed in the following section.

## III. THE EM FRAMEWORK FOR THE IDENTIFICATION OF SWITCHING NONLINEAR SYSTEMS

By making use of the training data  $\mathbf{u}, \mathbf{y}$  and the initial state  $x(0)$ , we aim at determining the parameters  $\Theta$  of the RNNs  $\mathcal{N}_x$  and  $\mathcal{N}_y$  in (5). The EM algorithm can iteratively obtain an estimate of  $\Theta$  when the original system includes unobservable hidden variables, such as the switching sequence  $\mathbf{S}$ .

Let us then denote the parameter estimate at the  $k$ -th iteration as  $\Theta^k$ . The likelihood function associated to the data collected over  $T$  of  $\mathbf{y}, \mathbf{x} = \mathbf{x}_{1:T} := \{x(1), \dots, x(T)\}$ ,  $\mathbf{S}$ , and  $\Theta$ , can be expressed as:

$$\log \mathbb{P}[\mathbf{y}, \mathbf{x}, \mathbf{S}, \Theta] = \log \mathbb{P}[\mathbf{y}] + \log \mathbb{P}[\mathbf{x}, \mathbf{S}, \Theta|\mathbf{y}]. \quad (8)$$

Given some  $\Theta^k$ , the conditional expectation of  $\mathbb{P}_{\Theta^k}[\mathbf{x}, \mathbf{S}|\mathbf{y}]$  is abbreviated as

$$\mathbb{E}_{\Theta^k}[\cdot] = \int \sum_{\mathbf{S}} (\cdot) \mathbb{P}_{\Theta^k}[\mathbf{x}, \mathbf{S}|\mathbf{y}] d(\mathbf{x})$$

Then, take the expectation operator  $\mathbb{E}_{\Theta^k}[\cdot]$  on both sides of (8):

$$\mathbb{E}_{\Theta^k}[\log \mathbb{P}[\mathbf{y}, \mathbf{x}, \mathbf{S}, \Theta]] = Q_1 + Q_2 + Q_3, \quad (9)$$

where

$$\begin{aligned}
Q_1 &= \sum_{i=1}^T \int \sum_{s_i} \log \mathbb{P}[y(i)|x(i), u(i), \theta_{g,s_i}] \\
&\quad \mathbb{P}_{\Theta^k}[x(i), s(i)|\mathbf{y}]d(x(i)) \\
&\quad + \sum_{i=1}^{T-1} \int \int \sum_{s_i} \log \mathbb{P}[x(i+1)|x(i), u(i), \theta_{f,s_i}] \\
&\quad \mathbb{P}_{\Theta^k}[x(i+1), x(i), s(i)|\mathbf{y}]d(x(i+1))d(x(i)), \\
Q_2 &= \sum_{i=1}^K \log \mathbb{P}[\theta_{f,s_i}]\mathbb{P}[\theta_{g,s_i}], \\
Q_3 &= \sum_{j=1}^K \sum_{l=1}^K \mathbb{E}_{\Theta^k} \left[ \sum_{i=1}^T \mathbb{I}(j, s_i)\mathbb{I}(l, s_{i-1}) \right] \log \pi_{j,l} + \log \pi_{s_0}.
\end{aligned}$$

With  $\boldsymbol{\theta}_f := \{\theta_{f,1}, \dots, \theta_{f,K}\}$  and  $\boldsymbol{\theta}_g := \{\theta_{g,1}, \dots, \theta_{g,K}\}$ , we show next that the cost in (7) can be obtained in the maximum likelihood estimation framework

*Proposition 1:* Minimizing the cost function  $J(\mathbf{y}, \mathbf{u}, \Theta, \mathbf{S})$  in (7) w.r.t.  $\Theta$ , where

$$\begin{aligned}
\ell(\mathbf{y}, \mathbf{u}, \Theta, \mathbf{S}) &= - \sum_{t=1}^T \log \mathbb{P}[y(t)|x(t), u(t), \theta_{g,s_t}] \\
&\quad - \sum_{t=1}^{T-1} \log \mathbb{P}[x(t+1)|x(t), u(t), \theta_{f,s_t}], \quad (10a)
\end{aligned}$$

$$r(\Theta) = - \sum_{i=1}^K (\log \mathbb{P}[\theta_{f,i}] + \log \mathbb{P}[\theta_{g,i}]), \quad (10b)$$

$$\mathcal{L}(\mathbf{S}) = - \log \mathbb{P}[\mathbf{S}] = - \sum_{t=1}^T \log \pi_{s_t, s_{t-1}} - \log \pi_{s_0}, \quad (10c)$$

is equivalent to maximizing the joint probability density function  $\mathbb{P}[\mathbf{y}, \mathbf{x}, \mathbf{S}, \Theta]$ .

The proof of the Proposition 1 is shown in Appendix A.

*Remark 2:* Proposition 1 establishes a connection between the cost function and the joint probability density function (pdf). Specifically, it transforms a deterministic training problem into a probabilistic one. We will therefore be able to leverage tools from probability theory and statistics to address our challenging identification problem, as the fact that the pdf contains unknown states  $\mathbf{S}$  will be resolved iteratively by our EM-based algorithm.

It thus follows from Proposition 1 that, rather than minimizing  $J(\mathbf{y}, \mathbf{u}, \Theta, \mathbf{S})$  with ingredients as in (10), we can equivalently maximize the following objective function:

$$Q(\Theta, \Theta^k) = \mathbb{E}_{\Theta^k} [\log \mathbb{P}[\mathbf{y}, \mathbf{x}, \mathbf{S}, \Theta]].$$

Conceptually, the proposed EM-based methodology then consists in the two steps summarized next:

- 1) **Expectation** (E-step): Calculate the optimal posterior estimate values of the implicit state (switching sequence  $\mathbf{S}$ ) using the parameter estimates obtained in the M-step at the previous iteration, and compute the expectation to obtain  $Q(\Theta, \Theta^k)$ ;
- 2) **Maximization** (M-step): The objective function  $Q(\Theta, \Theta^k)$  is maximized w.r.t.  $\Theta$  and  $\Pi$  by using an

EKF-based method. Then, the estimate of the parameter  $\Theta$  is updated to obtain  $\Theta^{k+1}$ .

Next, each one of the steps above will be detailed starting from the M-step in §III-A, which will introduce the ingredients required for the E-step in §III-B.

#### A. The maximization step

Given its expression, maximizing  $Q(\Theta, \Theta^k)$  yields the optimal parameter estimates and transition matrix  $\Pi$  together. By starting with  $\Pi$ , it is clear that a maximizer to  $Q(\Theta, \Theta^k)$  w.r.t.  $\Pi$  can be equivalently found as a maximizer to  $Q_3$  subject to (6). Then, the entries of the transition matrix can be calculated, for all  $l = 1, \dots, K$ , as:

$$\pi_{j,l} = \frac{\mathbb{E}_{\Theta^k} \left[ \sum_{i=1}^T \mathbb{I}(j, s_i)\mathbb{I}(l, s_{i-1}) \right]}{\sum_{j=1}^K \mathbb{E}_{\Theta^k} \left[ \sum_{i=1}^T \mathbb{I}(j, s_i)\mathbb{I}(l, s_{i-1}) \right]} \quad (11)$$

Then, the optimal parameter estimates can be given by:

$$\Theta^{k+1} = \arg \max_{\Theta} Q(\Theta, \Theta^k). \quad (12)$$

To this end, a common approach is the maximum likelihood (ML) technique [37] that requires one to calculate the extreme points of the parameter (correspond to the optimal estimates). Therefore, the gradient descent (GD) method can be applied to solve the maximization problem (12) iteratively with update

$$\Theta^{n+1} = \Theta^n - \alpha_n (\partial Q(\Theta, \Theta^n) / \partial \Theta),$$

where  $n = 1, \dots, N$  is the training epoch index, with associated learning rate  $\alpha_n$ . The gradients of  $Q(\Theta, \Theta^k)$  w.r.t. all parameters in the RNNs (3)–(4) can then be calculated by applying the chain rule. However, the GD method has few disadvantages, such as slow convergence rate when the gradient is close to zero, and over-fitting, especially in case of noise-corrupted data. Although there are currently many well-known algorithms and NN structures proposed to address these problems, such as SGD, Adam, BiGRU; however, the objective function contains unknown state parameters  $\mathbf{x}$  and  $\mathbf{S}$ , which poses several challenges to their differentiation.

Inspired by [5], we propose instead to employ an EKF to recursively update the parameter  $\Theta$  and state trajectory  $\mathbf{x}$ . For computational convenience, the parameter will be vectorized and the dynamics in (5) rewritten as follows:

$$\begin{aligned}
x(t+1) &= \mathcal{N}_{x,s_t}(x(t), u(t), \theta_{f,s_t}) + \zeta(t), \\
y(t) &= \mathcal{N}_{y,s_t}(x(t), u(t), \theta_{g,s_t}) + \xi(t), \\
\vartheta(t+1) &= \vartheta(t) + \eta(t),
\end{aligned} \quad (13)$$

with  $\vartheta := [\vartheta_{f,s_t}^\top \ \vartheta_{g,s_t}^\top]^\top$ , and  $\vartheta_{f,s_t} := \text{vec}(\theta_{f,s_t})$ ,  $\vartheta_{g,s_t} := \text{vec}(\theta_{g,s_t})$ . In (13) we have implicitly assumed that the parameters vector dynamics  $\vartheta$  is affected by Gaussian white noise  $\eta(t)$  with zero mean and variance  $\Sigma_{\vartheta}(t)$

We discuss next the update process of the EKF at time  $t$ . First, the nonlinear functions  $\mathcal{N}_{x,s_t}$  and  $\mathcal{N}_{y,s_t}$  need to be expanded in a Taylor series to perform the first-order

linearization of the system (13). The corresponding Jacobian matrices read as follows:

$$F(t) = \frac{\partial \mathcal{N}_{x,s_t}}{\partial \begin{bmatrix} x \\ \vartheta \end{bmatrix}} \bigg|_{\hat{x}(t), \hat{\vartheta}(t), u(t)} = \begin{bmatrix} \frac{\partial \mathcal{N}_{x,s_t}}{\partial x} & \frac{\partial \mathcal{N}_{x,s_t}}{\partial \vartheta} & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \quad (14)$$

$$H(t) = \frac{\partial \mathcal{N}_{y,s_t}}{\partial \begin{bmatrix} x \\ \vartheta \end{bmatrix}} \bigg|_{\hat{x}(t), \hat{\vartheta}(t), u(t)} = \begin{bmatrix} \frac{\partial \mathcal{N}_{y,s_t}}{\partial x} & 0 & \frac{\partial \mathcal{N}_{y,s_t}}{\partial \vartheta} \end{bmatrix}. \quad (15)$$

Then, the prior estimates of  $\vartheta(t+1)$  and state  $x(t+1)$ , denoted as  $\hat{\vartheta}^-(t+1)$  and  $\hat{x}^-(t+1)$ , can be computed by exploiting the forward propagation process of the RNNs:

$$\begin{bmatrix} \hat{x}^-(t+1) \\ \hat{\vartheta}^-(t+1) \end{bmatrix} = \begin{bmatrix} \mathcal{N}_{x,s_t}(\hat{x}(t), u(t), \hat{\vartheta}_{f,s_t}(t)) \\ \hat{\vartheta}(t) \end{bmatrix}. \quad (16)$$

With this regard, the measurement equation can then be used to update and correct the prior estimates of the parameter  $\hat{\vartheta}^-(t+1)$  and state  $\hat{x}^-(t+1)$ . In general, they are referred to as the posterior estimates of the parameter and state, denoted as  $\hat{\vartheta}(t+1)$  and  $\hat{x}(t+1)$ :

$$\begin{bmatrix} \hat{x}(t+1) \\ \hat{\vartheta}(t+1) \end{bmatrix} = \begin{bmatrix} \hat{x}^-(t+1) \\ \hat{\vartheta}^-(t+1) \end{bmatrix} + \Gamma(t)e(t), \quad (17)$$

where  $\Gamma(t)$  is the Kalman gain,  $e(t)$  is the posterior estimation error that can be calculated by the forward propagation of  $\mathcal{N}_{y,s_t}$  and the true output, i.e.,

$$e(t) = y(t) - \mathcal{N}_{y,s_t}(\hat{x}^-(t), u(t), \hat{\vartheta}_{g,s_t}^-(t)). \quad (18)$$

Thus, the optimal Kalman gain  $\Gamma(t)$  has to be found so as to minimize  $e(t)$ , which is equivalent to minimizing the covariance matrix of the posterior estimation error. Let us then define the prior and posterior error covariance matrices as:

$$P^-(t) := \mathbb{E} \left[ \left( \begin{bmatrix} x(t) \\ \vartheta(t) \end{bmatrix} - \begin{bmatrix} \hat{x}^-(t) \\ \hat{\vartheta}^-(t) \end{bmatrix} \right) \left( \begin{bmatrix} x(t) \\ \vartheta(t) \end{bmatrix} - \begin{bmatrix} \hat{x}^-(t) \\ \hat{\vartheta}^-(t) \end{bmatrix} \right)^\top \right],$$

$$P(t) := \mathbb{E} \left[ \left( \begin{bmatrix} x(t) \\ \vartheta(t) \end{bmatrix} - \begin{bmatrix} \hat{x}(t) \\ \hat{\vartheta}(t) \end{bmatrix} \right) \left( \begin{bmatrix} x(t) \\ \vartheta(t) \end{bmatrix} - \begin{bmatrix} \hat{x}(t) \\ \hat{\vartheta}(t) \end{bmatrix} \right)^\top \right].$$

Then, we have:

$$P^-(t) = \mathbb{E} \left[ \left( F(t) \left( \begin{bmatrix} x(t-1) \\ \vartheta(t-1) \end{bmatrix} - \begin{bmatrix} \hat{x}(t-1) \\ \hat{\vartheta}(t-1) \end{bmatrix} \right) + \begin{bmatrix} \zeta(t) \\ \eta(t) \end{bmatrix} \right) \left( F(t) \left( \begin{bmatrix} x(t-1) \\ \vartheta(t-1) \end{bmatrix} - \begin{bmatrix} \hat{x}(t-1) \\ \hat{\vartheta}(t-1) \end{bmatrix} \right) + \begin{bmatrix} \zeta(t) \\ \eta(t) \end{bmatrix} \right)^\top \right]$$

$$= F(t)P(t-1)F^\top(t) + \begin{bmatrix} \Sigma_1(t) & 0 \\ 0 & \Sigma_\vartheta(t) \end{bmatrix}, \quad (19)$$

and

$$P(t) = \mathbb{E} \left[ ((I - \Gamma(t)H(t))(x(t) - \hat{x}^-(t)) - \Gamma(t)\xi(t)) \right. \\ \left. ((I - \Gamma(t)H(t))(x(t) - \hat{x}^-(t)) - \Gamma(t)\xi(t))^\top \right]$$

$$= P^-(t) - P^-(t)H^\top(t)\Gamma^\top(t) - \Gamma(t)H(t)P^-(t) \\ + \Gamma(t)H(t)P^-(t)H^\top(t)\Gamma^\top(t) + \Gamma(t)\Sigma_2(t)\Gamma^\top(t). \quad (20)$$

Moreover, note that minimizing the covariance matrix  $P(t)$  is equivalent to minimizing its trace. Thus, the optimal  $\Gamma(t)$  assumes the following closed form expression:

$$\Gamma(t) = P^-(t)H^\top(t) [H(t)P^-(t)H^\top(t) + \Sigma_2(t)]^{-1}. \quad (21)$$

Finally, by substituting the obtained Kalman gain into (14), the updated error covariance matrices can be calculated for the subsequent iteration as:

$$P(t) = (I - \Gamma(t)H(t))P^-(t). \quad (22)$$

The instructions in (14)-(22) then summarize the iterative training process based on EKF for each submodel. Before performing (14)-(22), it is however key to cluster the training data by using the results provided in the E-step. We will detail this process in §III-B. We now show that performing the aforementioned submodel EKF-based training steps, for each submodel, is equivalent to maximizing the objective function  $Q(\Theta, \Theta^k)$ :

*Theorem 1:* Performing the steps in (14)-(22), separately for each submodel, is equivalent to maximizing the objective function  $Q(\Theta, \Theta^k)$  w.r.t.  $\Theta$ .

The proof of Theorem 1 is shown in Appendix A.

Theorem 1 proves that using EKF to update parameters and state variables is equivalent to directly maximizing the objective function  $Q(\Theta, \Theta^k)$ , thus simplifying heavily the parameter estimation process. The proposed EKF-based training alleviates the computational difficulties that may arise when directly maximizing the objective function through a recursive procedure.

### B. The expectation step

The maximization step described in §III-A requires an expression for the expectation of the switching sequence. Specifically, this shall be obtained based on the parameter  $\Theta^k$  and the state variables obtained from the maximization step at the previous iteration.

Due to the Markov property of the switching sequence, which means that the hidden state  $s_t$  at the current time is only related to the previous time, we have:

$$\mathbb{P}[s_t = i] = \sum_{j=1}^K \mathbb{P}[s_t = i, s_{t-1} = j]$$

$$= \sum_{j=1}^K \mathbb{P}[s_t = i | s_{t-1} = j] \mathbb{P}[s_{t-1} = j]. \quad (23)$$

Then, the posterior pdf of the switching sequence can be calculated by using the Bayes' rule, i.e.,  $\mathbb{P}[\mathcal{S} | \mathbf{y}, \mathbf{x}, \Theta] = \mathbb{P}[\mathbf{y}, \mathbf{x}, \Theta, \mathcal{S}] / \mathbb{P}[\mathbf{y}, \mathbf{x}, \Theta]$ . By maximizing the pdf of the switching sequence  $\mathcal{S}$ , we have:

$$\hat{\mathcal{S}} = \arg \max_{\mathcal{S}} \log \mathbb{P}[\mathcal{S} | \mathbf{y}, \mathbf{x}, \Theta]$$

$$= \arg \max_{\mathcal{S}} \log \mathbb{P}[\mathbf{y}, \mathbf{x}, \Theta, \mathcal{S}]$$

$$= \arg \max_{\mathcal{S}} \ell(\mathbf{y}, \mathbf{x}, \Theta, \mathcal{S}) + r(\Theta)$$

$$+ \sum_{t=1}^T \sum_{i=1}^K \sum_{j=1}^K \mathbb{I}(s_t, i) \mathbb{I}(s_{t-1}, j) \log \pi_{i,j} \quad (24)$$

Unfortunately, the calculation of (24) requires to try all possibilities of  $\mathcal{S}$ , which is however computationally challenging to obtain. To derive the estimate for the switching sequence, a number of  $K^{T+1}$  calculations are thus necessary. To alleviate the resulting computational burden, we adopt a moving window approach for calculating these posterior probabilities. In words, this method maintains the estimated switching mode from the previous time instant, while calculating the pdf for the switching mode at the current time instant. Successively, the estimation of the current mode is obtained by maximizing the pdf of the switching mode at the current time instant.

We now provide a specific procedure starting from  $t = 0$ . Specifically, we note that the loss of the switching sequence (10c) is only related to the initial mode. In fact, for all possible modes the optimal one can be chosen as:

$$\begin{aligned} \hat{s}_1 &= \arg \max_{s_1} \mathbb{P}[s_1 | y(1), x(1), u(1), \Theta] \\ &= \arg \min_{s_1} \ell(y(1), u(1), \theta_{f,s_1}, \theta_{g,s_1}) + r(\Theta) + \mathcal{L}(s_1). \end{aligned} \quad (25)$$

Note that, when the initial mode is determined, the step-forward can then be computed for all possible switching sequences  $\mathcal{S}_{2:T}$ . Assume the length of the considered time window is  $T_w$ , and denote the time window as  $\mathcal{W}_t = \{t, t+1, \dots, t+T_w-1\}$ . Then, the switching sequence  $\mathcal{S}_{\mathcal{W}_t}$  can be computed by maximizing the local posterior pdf in (24):

$$\begin{aligned} \hat{\mathcal{S}}_{\mathcal{W}_t} &= \arg \max_{\mathcal{S}_{\mathcal{W}_t}} \mathbb{P}[\mathcal{S}_{\mathcal{W}_t} | \mathbf{y}_{\mathcal{W}_t}, \mathbf{x}_{\mathcal{W}_t}, \mathbf{u}_{\mathcal{W}_t}, \Theta] \\ &\quad \text{s.t. } s_{t-1} = \hat{s}_{t-1}, \\ &= \arg \min_{\mathcal{S}_{\mathcal{W}_t}} \ell(\mathbf{y}_{\mathcal{W}_t}, \mathbf{u}_{\mathcal{W}_t}, \Theta) + r(\Theta) + \sum_{t \in \mathcal{W}_t} \log \pi_{s_t, s_{t-1}}, \\ &\quad \text{s.t. } s_{t-1} = \hat{s}_{t-1}. \end{aligned} \quad (26)$$

The idea is thus to calculate an optimal solution to (26) for all the possible switching sequences  $\mathcal{S}_{\mathcal{W}_t}$ . Then, the optimal mode at time instant  $t$  can be fixed as the first element of the  $\mathcal{S}_{\mathcal{W}_t}$ , i.e.,  $\hat{s}_t = \hat{\mathcal{S}}_{\mathcal{W}_t}(1)$ . This process is then repeated up to  $t = T - T_w + 1$ , and hence obtaining an estimate of switching sequence  $\hat{\mathcal{S}}$ .

Remarkably, with the proposed approach the posterior pdf is calculated  $K^{T_w+1}$  times for each time instant  $t = 2, \dots, T - T_w + 1$ . Thus, the computational complexity of the moving window approach is only  $O((T - T_w)K^{T_w+2})$ .

*Remark 3:* The time window length  $T_w \in \{1, \dots, T - 1\}$  offers a trade-off between the accuracy and computational complexity. When  $T_w = T - 1$ , the moving window approach degenerates into considering all possible of switching sequences, and the computational complexity increases up to  $O(K^{T+1})$ .

*Remark 4:* The moving window method calculates the posterior probability at each time instant by only considering the possible modes at the current time and connecting them with the estimated mode from the previous time. This approach alleviates the combinatorial nature of enumerating all possible sequences, reducing the time complexity from exponential to linear order, and significantly improving the running speed of the algorithm, making it suitable for processing large-scale dataset.

---

**Algorithm 1** EM-based identification of switching nonlinear system

---

**Initialization:** Collect data  $\mathbf{y}_{1:T}, \mathbf{u}_{1:T}$ , set  $\Theta^0$ , retrieve number of the modes  $K$

**Iteration**  $k \in \mathbb{Z}$ :

- 1) **E-step:** Compute  $Q(\Theta, \Theta^k)$  using (9), (24), (25), (26)
  - 2) **M-step:** Maximize  $Q(\Theta, \Theta^k)$  w.r.t.  $\Theta$  and  $\Pi$  by using (11) and the EKF-based procedure (14)-(22)
- 

### C. Convergence analysis

The main steps of our method are summarized in Algorithm 1. We characterize next its convergence properties.

*Proposition 2:* Given a set of data  $\mathbf{y}, \mathbf{u}$ , and a number of modes  $K$ , let  $\{\Theta^k\}_{k \in \mathbb{Z}}$  be the sequence generated by Algorithm 1. Then, the log likelihood function  $\log \mathbb{P}_{\Theta}[\mathbf{y}]$  is non-decreasing, i.e.,  $\log \mathbb{P}_{\Theta^{k+1}}[\mathbf{y}] \geq \log \mathbb{P}_{\Theta^k}[\mathbf{y}]$ .

*Proof.* According to Theorem 1, performing the EKF steps (14)-(22) is equivalent to maximizing the objective function  $Q(\Theta, \Theta^k)$ , which yields at each iteration

$$Q(\Theta, \Theta^{k+1}) \geq Q(\Theta, \Theta^k). \quad (27)$$

Then, we have:

$$\begin{aligned} \log \mathbb{P}_{\Theta}[\mathbf{y}] - \log \mathbb{P}_{\Theta^k}[\mathbf{y}] &= Q(\Theta, \Theta^k) - Q(\Theta^k, \Theta^k) \\ &\quad + \int \log \mathbb{P}_{\Theta^k}[\mathbf{x}, \mathcal{S} | \mathbf{y}] \log \mathbb{P}_{\Theta^k}[\mathbf{x}, \mathcal{S} | \mathbf{y}] d(\mathbf{x}) \\ &\quad - \int \log \mathbb{P}_{\Theta}[\mathbf{x}, \mathcal{S} | \mathbf{y}] \log \mathbb{P}_{\Theta^k}[\mathbf{x}, \mathcal{S} | \mathbf{y}] d(\mathbf{x}) \\ &= Q(\Theta, \Theta^k) - Q(\Theta^k, \Theta^k) \\ &\quad + \int \frac{\log \mathbb{P}_{\Theta^k}[\mathbf{x}, \mathcal{S} | \mathbf{y}]}{\log \mathbb{P}_{\Theta}[\mathbf{x}, \mathcal{S} | \mathbf{y}]} \log \mathbb{P}_{\Theta^k}[\mathbf{x}, \mathcal{S} | \mathbf{y}] d(\mathbf{x}) \\ &= Q(\Theta, \Theta^k) - Q(\Theta^k, \Theta^k) \\ &\quad + DL(\log \mathbb{P}_{\Theta^k}[\mathbf{x}, \mathcal{S} | \mathbf{y}] \| \log \mathbb{P}_{\Theta}[\mathbf{x}, \mathcal{S} | \mathbf{y}]). \end{aligned}$$

where  $DL(P \| Q)$  is the Kullback-Leibler divergence, which is guaranteed to be nonnegative [34]. Therefore, applying (27) directly lead to a non-decreasing sequence for  $\log \mathbb{P}_{\Theta}[\mathbf{y}]$ , i.e.,  $\log \mathbb{P}_{\Theta^{k+1}}[\mathbf{y}] \geq \log \mathbb{P}_{\Theta^k}[\mathbf{y}]$ , completing the proof.  $\square$

Proposition 2 is a well-known result for EM method [3] which says that the parameter estimates generated at each iteration of Algorithm 1 are approximating the optimal value of the maximum likelihood estimate. By denoting with  $\Theta^*$  as an optimal solution to the minimization problem of (7), we can claim the following result:

*Theorem 2:* Denote the Hessian matrix of  $Q(\Theta, \Theta^k)$  as  $\mathcal{H}(\Theta, \Theta^k)$ , assume its inverse being bounded, and  $\Theta \rightarrow \mathcal{H}(\Theta, \Theta^k)$  Lipschitz continuous. Then, the sequence  $\{\Theta^k\}_{k \in \mathbb{Z}}$  generated by Algorithm 1 converges quadratically to some optimal solution  $\Theta^*$ .

*Proof.* According to the EKF-based training process for the parameters update, and (33)–(34), we have:

$$\begin{aligned}
\vartheta(t+1) &= \vartheta(t) + P^-(t)H^\top(t) [H(t)P^-(t)H^\top(t) + \Sigma_2(t)]^{-1} e(t) \\
&= \vartheta(t) + (P^-(t)^{-1} + H^\top(t)\Sigma_2^\top(t)H(t))^{-1} H^\top(t)\Sigma_2^{-1}(t)e(t) \\
&= \vartheta(t) + P(t)H^\top(t)\Sigma_2^{-1}(t)(y(t) - \mathcal{N}_{y,s_t}(\vartheta(t))), \quad (28)
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial Q(\vartheta(t), \Theta^k)}{\partial \vartheta(t)} &= -H^\top(t)\Sigma_2^{-1}(t)e(t), \\
\mathcal{H}(\vartheta(t), \Theta^k) &= \frac{\partial^2 Q(\vartheta(t), \Theta^k)}{\partial \vartheta(t)^2} \\
&= P^-(t)^{-1} + H^\top(t)\Sigma_2^\top(t)H(t).
\end{aligned}$$

Let us define  $\mathcal{F}(a) = H^\top(t)\Sigma_2^{-1}(t)(y(t) - \mathcal{N}_{y,s_t}(\vartheta(t) + a(\Theta^* - \vartheta(t))))$ , and its derivative as  $\mathcal{F}'(a) = \mathcal{H}(\vartheta(t) + a(\Theta^* - \vartheta(t)), \Theta^k)(\Theta^* - \vartheta(t))$ . According to Newton-Leibniz formula, we readily obtain:

$$\begin{aligned}
& - \frac{\partial Q(\vartheta(t), \Theta^k)}{\partial \vartheta(t)} \\
&= H^\top(t)\Sigma_2^{-1}(t)(y(t) - \mathcal{N}_{y,s_t}(\Theta^*)) \\
&\quad - H^\top(t)\Sigma_2^{-1}(t)(y(t) - \mathcal{N}_{y,s_t}(\vartheta(t))) \\
&= \mathcal{F}(1) - \mathcal{F}(0) \\
&= \int_0^1 \mathcal{F}'(a)da \\
&= \int_0^1 \mathcal{H}(\vartheta(t) + a(\Theta^* - \vartheta(t)), \Theta^k)(\Theta^* - \vartheta(t))da.
\end{aligned}$$

By subtracting  $\Theta^*$  from both sides of (28) simultaneously, it can be inferred that:

$$\begin{aligned}
\vartheta(t+1) - \Theta^* &= \vartheta(t) - \Theta^* + \mathcal{H}(\vartheta(t), \Theta^k)^{-1} \\
&\quad \int_0^1 \mathcal{H}(\vartheta(t) + a(\Theta^* - \vartheta(t)), \Theta^k)(\Theta^* - \vartheta(t))da \\
&= \mathcal{H}(\vartheta(t), \Theta^k)^{-1} \int_0^1 (\mathcal{H}(\vartheta(t) + a(\Theta^* - \vartheta(t)), \Theta^k) \\
&\quad - \mathcal{H}(\vartheta(t), \Theta^k))(\Theta^* - \vartheta(t))da.
\end{aligned}$$

Taking the norm on both sides of the relation above yields:

$$\begin{aligned}
& \|\vartheta(t+1) - \Theta^*\| \\
&\leq \|\mathcal{H}(\vartheta(t), \Theta^k)^{-1}\| \cdot \|\Theta^* - \vartheta(t)\| \\
&\cdot \left\| \int_0^1 (\mathcal{H}(\vartheta(t) + a(\Theta^* - \vartheta(t)), \Theta^k) - \mathcal{H}(\vartheta(t), \Theta^k))da \right\|.
\end{aligned}$$

According to the assumptions postulated on the Hessian matrix  $\mathcal{H}$ , we have:

$$\begin{aligned}
\mathcal{H}(\vartheta(t), \Theta^k) &\leq \frac{1}{\mathcal{A}}, \\
\|\mathcal{H}(\vartheta_1, \Theta^k) - \mathcal{H}(\vartheta_2, \Theta^k)\| &\leq \mathcal{B}\|\vartheta_1 - \vartheta_2\|,
\end{aligned}$$

for some nonnegative scalars  $\mathcal{A}$  and  $\mathcal{B}$ . We can therefore derive the following bound on the distance between the current parameter estimate  $\vartheta(t+1)$  and some  $\Theta^*$ :

$$\|\vartheta(t+1) - \Theta^*\| \leq \frac{2\mathcal{B}}{\mathcal{A}} \|\vartheta(t) - \Theta^*\|^2.$$

For a finite training dataset of length  $T$  we have hence proved that the sequence  $\{\Theta^k\}_{k \in \mathbb{Z}}$  generated by Algorithm 1 enjoys quadratic convergence to some  $\Theta^*$ .  $\square$

By assuming that the Hessian matrix is bounded and Lipschitz continuous, Theorem 2 says that Algorithm 1 produces a sequence of parameter estimates that quickly converges to an optimal set of parameters  $\Theta^*$ . Generally speaking, if a function is strongly convex, then its Hessian matrix is usually bounded and Lipschitz continuous. These properties are key for establishing convergence to the optimal solution at a certain rate.

#### IV. NUMERICAL EXPERIMENTS

The proposed EM-based technique is now tested on three numerical examples. Algorithm 1 is then run  $N$  times with  $K$  fixed modes and the window length  $T_w$ . To evaluate the effectiveness of our methodology, we will make use of the mean square error (MSE):

$$\text{MSE} = \frac{1}{T} \sum_{t=1}^T (y(t) - \hat{y}(t))^2,$$

along with the best fit rate (BFR):

$$\text{BFR} = 100 \left( 1 - \sqrt{\frac{\sum_{t=1}^T \|y(t) - \hat{y}(t)\|^2}{\sum_{t=1}^T \|y(t) - \bar{y}(t)\|^2}} \right) \%.$$

*A. An academic example: a switching system composed of a linear and a nonlinear part*

Consider the 1000 input/output pairs generated by the system in [31], which consists of two submodels ( $K = 2$ ): one linear and another nonlinear. The value of inputs follows a uniform distribution and are hence randomly generated in  $[0, 1]$ . The noise term follows a Gaussian distribution with zero mean and finite variance. The proposed NN structure with four RNN submodels is used to model the above switching system. Each of the RNN submodel has  $l_f = l_g = 2$  layers and four neurons. The activation function of these layers is  $\arctan(\cdot)$ . We then use the Algorithm 1 to train the above switching system.

In Fig. 1 we show the true and estimated outputs, the resulting MSE, and the true switching sequence with the estimated one over the time window  $[200, 300]$ . The three subgraphs presented in Fig. 1 collectively illustrate the superior performance of the proposed method. Then, our technique is compared with the kernel-based approach proposed in [31]. Table I reports both indices for the two methods under the different noise level, highlighting that our RNN-based technique exhibits better performance in most of the cases, especially in situations with large noise level.

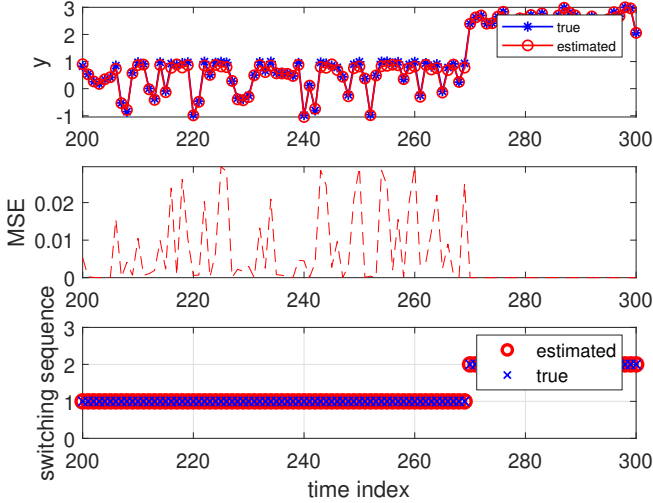


Fig. 1: Top: The true output (solid blue line with asterisks) and the estimated one by our method (solid red line with circles) over the time window  $t \in [200, 300]$ . Middle: The resulting MSE. Bottom: The true switching sequence (blue crosses) and the estimated one (red circles).

TABLE I: The MSE and BFR for the proposed method and the kernel-based approach in [31], under different noise level.

	noise level	RNNs	[31]
MSE	0.001	0.0054	<b>0.0043</b>
	0.01	<b>0.0056</b>	0.0684
	0.1	<b>0.0155</b>	0.0855
	0.2	<b>0.0451</b>	0.1061
BFR	0.001	93.50%	<b>96.49%</b>
	0.01	<b>93.39%</b>	71.86%
	0.1	<b>89.01%</b>	69.22%
	0.2	<b>81.23%</b>	67.22%

### B. An academic example: a switching system composed of two nonlinear parts

Consider the following switching nonlinear system:

$$\begin{aligned} x(t+1) &= A_{s_t} \tanh(x(t)) + B_{s_t} u(t) + \zeta(t), \\ y(t) &= C_{s_t} \sin(x(t)) + D_{s_t} u(t) - 2 + \xi(t), \end{aligned} \quad (29)$$

with  $n_u = 1$ ,  $n_x = 3$ ,  $n_y = 1$ . We take  $x(0) = 0$  and the values of the inputs  $u(t)$  follows a uniform distribution and are hence randomly generated in  $[0, 1]$ . The finite variance characterizing the noise terms  $\zeta(t)$  and  $\xi(t)$  is  $Q_x = Q_y = 10^{-3}$ . We consider  $K = 2$  modes with the following system

matrices/vectors:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0.8 & 0.2 & -0.1 \\ 0 & 0.9 & 0.1 \\ 0.1 & -0.1 & 0.7 \end{bmatrix}, & A_2 &= \begin{bmatrix} 0.5 & -0.2 & -0.1 \\ 0 & 0.9 & 0.1 \\ -0.1 & -0.3 & 0.8 \end{bmatrix}, \\ B_1 &= [-1 \ 0.5 \ 1]^\top, & B_2 &= [-0.5 \ 0.1 \ 0.5]^\top, \\ C_1 &= [-1 \ 1.5 \ 0.5], & C_2 &= [-0.1 \ -0.5 \ 0.8], \\ D_1 &= 0.1, & D_2 &= -0.1, \\ \Pi &= \begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix}. \end{aligned}$$

The data are generated by (29) for  $T = 1000$  samples. Then, we use the RNNs in (5) to model the dynamics in (29), where each RNN submodel  $\mathcal{N}_{x,1}$ ,  $\mathcal{N}_{x,2}$ ,  $\mathcal{N}_{y,1}$ , and  $\mathcal{N}_{y,2}$  has  $l_f = l_g = 2$  layers with six neurons. The activation function for the first layer  $f_1$ ,  $g_1$  is  $\arctan(\cdot)$ , and the output layer  $g_2$  is instead a linear mapping. Thus, the parameters  $\theta_{f,1}$ ,  $\theta_{f,2}$ ,  $\theta_{g,1}$ ,  $\theta_{g,2}$ , and the switching sequence  $S$  fully describe the system dynamics in (5).

We run Algorithm 1 for  $N = 10$  times and the EKF process for 10 epochs, for each RNN submodel. In this example, our technique is compared with the Bayesian ensemble learning algorithm proposed in [15]. Fig. 2 shows the true and estimated outputs obtained by the proposed method, demonstrating that our EM-based algorithm can achieve a great performance in the parameter estimation, with a remarkable 100% match rate in estimating the switching sequence.

To verify the performance of the proposed method against different noise levels, we use the trained RNNs to predict 100 trajectories with the different initial states and noise conditions. In Figs. 3 and 4 we report both mean and variance of the two indices considered, comparing the results obtained with our RNN-based method, the kernel-based method [31] and Bayesian ensemble learning [15]. It can indeed be observed that our technique features better accuracy and lower error, since our method has excellent performance in both MSE and BFR evaluation metrics, regardless of the level of noise.

### C. Battery state of charge estimation

We now test our approach to estimate the state of charge (SOC) in a battery management system. Specifically, we use a battery platform to generate the dataset of  $T = 1047$  samples, which consists of a computer, a battery tester (NEWARE CT-4008-5V12A-TB) and a group of lithium batteries (NCR18650PF). The dataset is composed of two parts: the control input consisting of both current and voltage in the circuit, and the output including the SOC at each time instant. In addition, the data contains two operating modes of the circuit overall, i.e., charging and discharging, illustrated in Fig. 5. Therefore, we use  $K = 2$  submodels to characterize different circuit operations. Then, RNNs are used to model the relation between inputs (current and voltage) and outputs (SOC) in each of these operating modes. Specifically, we choose  $l_f = l_g = 2$  layers with six neurons each. The activation functions adopted in the first and the second layer are standard affine and sigmoid functions, respectively. The EKF-based training procedure is run for 20 epochs for each RNN submodel, and Algorithm 1 is run  $N = 10$  times. The SOC

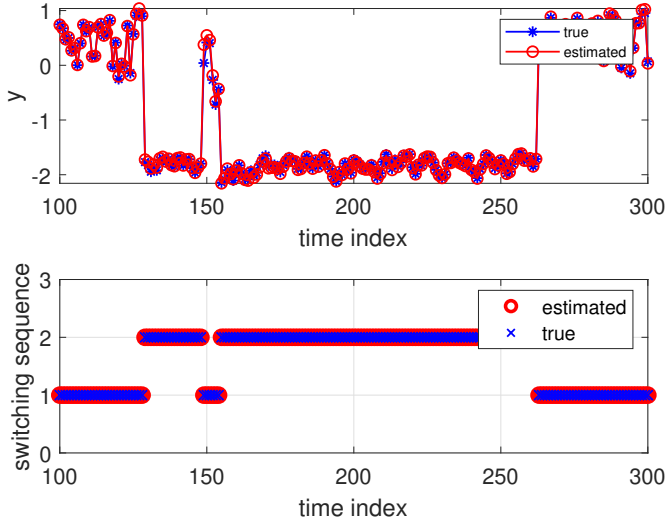


Fig. 2: Top: The true (solid blue line with asterisks) and the estimated output by using our RNN-based method (solid red line with circles) over the time window  $t \in [100, 300]$ . Bottom: The true switching sequence (blue crosses) and the one estimated by our method (red circles).

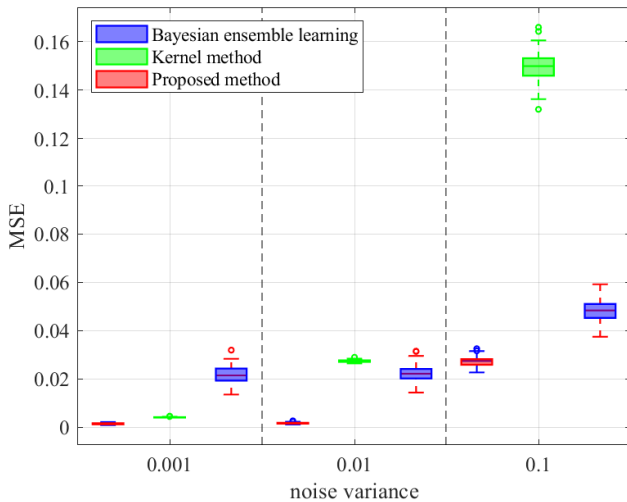


Fig. 3: The MSE obtained by proposed method (red), the kernel-based method (yellow) [31] and the Bayesian ensemble learning (blue) [15] under different noise conditions.

estimates and MSE errors are shown in Fig. 6 which reports the great performance of the proposed method. Furthermore, the match rate of the switching sequence achieves 100%.

We finally compare our technique with a standard SOC learning method [19], which uses the gate recurrent unit (GRU) based momentum algorithm to estimate the SOC. Table II reports both MSE and BFR indices for the two methods, highlighting how our approach results in better performance.

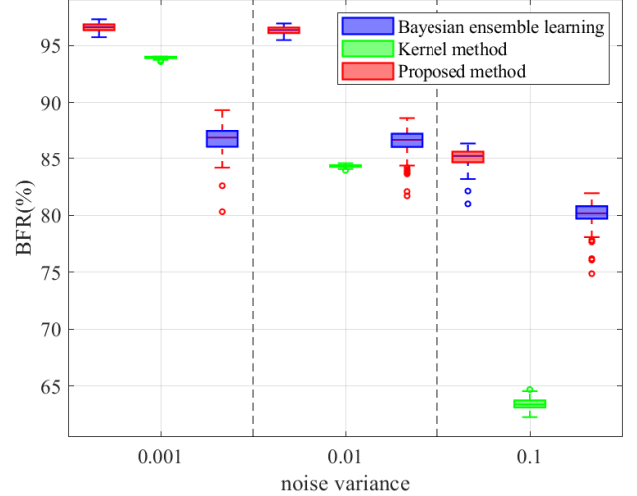


Fig. 4: The BFR obtained by proposed method (red), the kernel-based method (yellow) [31] and the Bayesian ensemble learning (blue) [15] under different noise conditions.

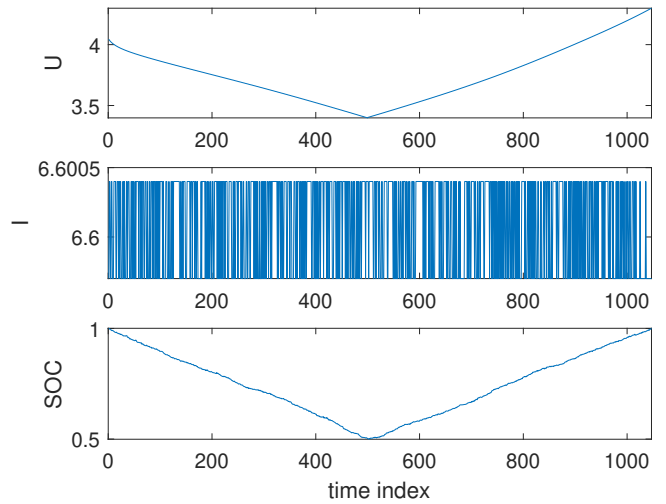


Fig. 5: The training data collected in the battery platform where  $U$  and  $I$  respectively represent the voltage and current in the circuit, and SOC represents the battery state of charge.

TABLE II: Battery platform: The MSE and BFR for the two algorithms

Method	BFR	MSE
GRU [19]	88.25%	0.0020
Proposed method	<b>92.62%</b>	<b>0.0004</b>

## V. CONCLUSION

We have presented a neural network-based scheme for the identification of switching nonlinear systems with unknown inner structures. In the EM framework, we have then devised an iterative procedure alternating an E-step and M-step. While in the former we have used a moving window approach to obtain the posterior estimate of the switching sequence, with

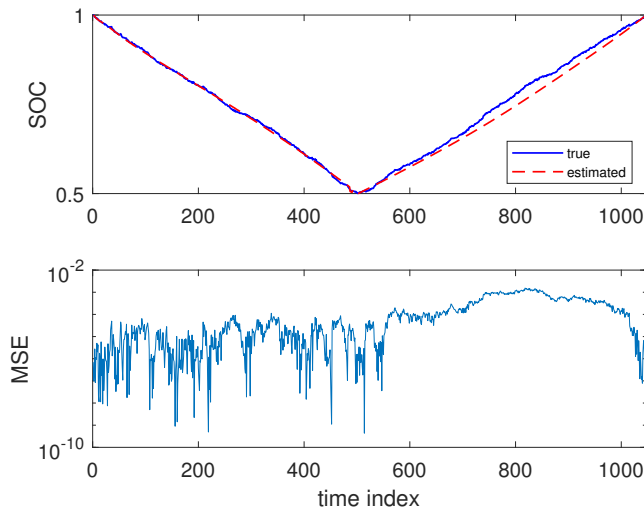


Fig. 6: Estimates and MSE errors of the SOC by using the proposed method.

significant reduction of the overall computational complexity, in the latter an EKF-based training procedure, suitable for switching systems, has been proposed to retrieve the parameter estimates for each subsystem, modeled through an RNN. Numerical experiments have been carried out to demonstrate that the proposed methodology exhibits excellent performance in terms of parameter estimation, model fitting, and switching sequence identification.

Besides RNNs, more expressive neural network architectures, such as long short-term memory networks (LSTMs) or GRUs, could be considered to further enhance the model's generalization ability. In addition, one may also investigate the case in which the number of subsystems is not known, thereby facing further technical challenges. These aspects are left to future work.

## REFERENCES

- [1] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pages 136–145. PMLR, 2017.
- [2] Laurent Bako. Identification of switched linear systems via sparse optimization. *Automatica*, 47(4):668–677, 2011.
- [3] Mark P. Balenzuela, Adrian G. Wills, Renton Christopher, and Ninness Brett. Parameter estimation for jump markov linear systems. *Automatica*, 135:109949, 2022.
- [4] Roy Batruni. A multilayer neural network with piecewise-linear structure and back-propagation learning. *IEEE Transactions on Neural Networks*, 2(3):395–403, 1991.
- [5] Alberto Bemporad. Recurrent neural network training with convex loss and regularization functions by extended Kalman filtering. *IEEE Transactions on Automatic Control*, 68(9):5661–5668, 2022.
- [6] Alberto Bemporad, Valentina Breschi, Dario Piga, and Stephen P Boyd. Fitting jump models. *Automatica*, 96:11–21, 2018.
- [7] Valentina Breschi, Dario Piga, and Alberto Bemporad. Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica*, 73:155–162, 2016.
- [8] Raffaella Carloni, Ricardo G Sanfelice, Andrew R Teel, and Claudio Melchiorri. A hybrid control strategy for robust contact detection and force regulation. In *2007 American Control Conference*, pages 1461–1466. IEEE, 2007.
- [9] Antoni B Chan and Nuno Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):909–926, 2008.
- [10] Filippo Fabiani, Bartolomeo Stellato, Daniele Masti, and Paul J Goulart. A neural network-based approach to hybrid systems identification for control. *Automatica*, 2025. (In press).
- [11] Giancarlo Ferrari-Trecate, Marco Muselli, Diego Liberati, and Manfred Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.
- [12] Andrea Garulli, Simone Paoletti, and Antonio Vicino. A survey on switched and piecewise affine system identification. *IFAC Proceedings Volumes*, 45(16):344–355, 2012.
- [13] Arash Golabi, Nader Meskin, Roland Toth, and Javad Mohammadpour. A bayesian approach for lpv model identification and its application to complex processes. *IEEE Transactions on Control Systems Technology*, 25(6):2160–2167, 2017.
- [14] Massimo Guidolin. Markov switching models in empirical finance. In *Missing data methods: Time-series methods and applications*, pages 1–86. Emerald Group Publishing Limited, 2011.
- [15] Antti Honkela. *Nonlinear switching state-space models*. PhD thesis, Helsinki University of Technology, 2001.
- [16] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.
- [17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [18] Jeffrey Humpherys, Preston Redd, and Jeremy West. A fresh look at the Kalman filter. *SIAM review*, 54(4):801–823, 2012.
- [19] Meng Jiao, Dongqing Wang, and Jianlong Qiu. A GRU-RNN based momentum optimized algorithm for SOC estimation. *Journal of Power Sources*, 459:228051, 2020.
- [20] Kon Johan, Roland Toth, Wijdeven Jeroen, Heertjes Marcel, and Oomen Tom. Guaranteeing stability in structured input-output models: With application to system identification. *IEEE Control Systems Letters*, 8:1565–1570, 2024.
- [21] Fabien Lauer. On the complexity of piecewise affine system identification. *Automatica*, 62:148–153, 2015.
- [22] Fabien Lauer, Gérard Bloch, and René Vidal. A continuous optimization framework for hybrid system identification. *Automatica*, 47(3):608–613, 2011.
- [23] Daniele Masti and Alberto Bemporad. Learning nonlinear state-space models using autoencoders. *Automatica*, 129:109666, 2021.
- [24] Henrik Ohlsson, Lennart Ljung, and Stephen Boyd. Segmentation of ARX-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010.
- [25] Dario Piga, Valentina Breschi, and Alberto Bemporad. Estimation of jump Box–Jenkins models. *Automatica*, 120:109126, 2020.
- [26] Dario Piga, Pepijn Cox, Roland Toth, and Vincent Laurain. Lpv system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53(C):329–338, 2015.
- [27] Riccardo Porreca, Samuel Drulhe, Hidde De Jong, and Giancarlo Ferrari-Trecate. Identification of parameters and structure of piecewise affine models of genetic networks. *IFAC Proceedings Volumes*, 42(10):587–592, 2009.
- [28] Vinay Prasad and B Wayne Bequette. Nonlinear system identification and model reduction using artificial neural networks. *Computers & Chemical Engineering*, 27(12):1741–1754, 2003.
- [29] Giulio Ripaccioli, Alberto Bemporad, Francis Assadian, Clement Dextreit, Stefano Di Cairano, and Ilya V Kolmanovskiy. Hybrid modeling, identification, and predictive control: An application to hybrid electric vehicle energy management. In *Hybrid Systems: Computation and Control: 12th International Conference, HSCC 2009, San Francisco, CA, USA, April 13-15, 2009. Proceedings 12*, pages 321–335. Springer, 2009.
- [30] Jacob Roll, Alberto Bemporad, and Lennart Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, 2004.
- [31] Anna Scampicchio, Alberto Giaretta, and Gianluigi Pillonetto. Nonlinear hybrid systems identification using kernel-based techniques. volume 51, pages 269–274. Elsevier, 2018.
- [32] Thomas Schlegel, Martin Buss, and Günther Schmidt. A hybrid systems approach toward modeling and dynamical simulation of dextrous manipulation. *IEEE/ASME transactions on mechatronics*, 8(3):352–361, 2003.
- [33] Björn Schuller, Martin Wöllmer, Tobias Moosmayr, Günther Ruske, and Gerhard Rigoll. Switching linear dynamic models for noise robust in-car speech recognition. In *Pattern Recognition: 30th DAGM Symposium Munich, Germany, June 10-13, 2008 Proceedings 30*, pages 244–253. Springer, 2008.

- [34] Ji Shuyi, Zhang Zizhao, Ying Shihui, Wang Liejun, Zhao Xibin, and Yue Gao. Kullback–leibler divergence metric learning. *IEEE transactions on cybernetics*, 52(4):2047–2058, 2022.
- [35] Kelvin Tan, William J Parquette, and Meng Tao. A predictive algorithm for maximum power point tracking in solar photovoltaic systems through load management. *Solar Energy*, 265:112127, 2023.
- [36] Allan Timmermann. *Markov Switching Models in Finance*, pages 1–3. John Wiley & Sons, Ltd, 2015.
- [37] J Umenberger, J Wägberg, I.R Manchester, and T.B. Schon. Maximum likelihood identification of stable linear dynamical systems. *Automatica*, 96:280–292, 2018.
- [38] René Vidal, Alessandro Chiuso, and Stefano Soatto. Observability and identifiability of jump linear systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 4, pages 3614–3619. IEEE, 2002.
- [39] Jun Xu, Xiaolin Huang, and Shuning Wang. Adaptive hinging hyperplanes and its applications in dynamic system identification. *Automatica*, 45(10):2325–2332, 2009.
- [40] Pan Zhaojie, Li Chenyu, Plaza Antonio, Chanussot Jocelyn, and Hong Danfeng. Hyperspectral image classification with mamba. *IEEE Transactions on Geoscience and Remote Sensing*, page 1, 2024.

## APPENDIX

*Proof of Proposition 1:* In view of Standing Assumption 3, we have:

$$\begin{aligned}
& \mathbb{P}[\mathbf{y}, \mathbf{x}, \mathbf{S}, \Theta | \mathbf{u}] \\
&= \mathbb{P}[\mathbf{y}, \mathbf{x} | \mathbf{u}, \mathbf{S}, \Theta] \mathbb{P}[\mathbf{S}, \Theta | \mathbf{u}] \\
&= \mathbb{P}[\mathbf{y} | \mathbf{x}, \mathbf{u}, \mathbf{S}, \theta_f^k] \mathbb{P}[\mathbf{x} | \mathbf{u}, \mathbf{S}, \theta_f^k] \\
&\quad \mathbb{P}[\mathbf{S} | \mathbf{u}] \mathbb{P}[\theta_f^k | \mathbf{u}] \mathbb{P}[\theta_g^k | \mathbf{u}] \\
&= \mathbb{P}[\mathbf{y} | \mathbf{x}, \mathbf{u}, \mathbf{S}, \theta_f^k] \mathbb{P}[\mathbf{x} | \mathbf{u}, \mathbf{S}, \theta_f^k] \mathbb{P}[\mathbf{S}] \mathbb{P}[\theta_f^k] \mathbb{P}[\theta_g^k] \\
&= \prod_{t=1}^T \mathbb{P}[y(t) | x(t), u(t), \theta_{g,s_t}] \\
&\quad \prod_{t=1}^T \mathbb{P}[x(t) | x(t-1), u(t-1), \theta_{f,s_t}] \prod_{i=1}^K \mathbb{P}[\theta_{f,i}] \\
&\quad \prod_{i=1}^K \mathbb{P}[\theta_{g,i}] \mathbb{P}[s_0] \prod_{t=1}^T \mathbb{P}[s_t | s_{t-1}].
\end{aligned}$$

By taking the logarithm of the probability density function as mentioned in §III we then obtain:

$$\begin{aligned}
& \log \mathbb{P}[\mathbf{y}, \mathbf{x}, \mathbf{S}, \Theta | \mathbf{u}] \\
&= \sum_{t=1}^T \log \mathbb{P}[y(t) | x(t), u(t), \theta_{g,s_t}] \\
&+ \sum_{t=1}^T \log \mathbb{P}[x(t) | x(t-1), u(t-1), \theta_{f,s_t}] + \sum_{i=1}^K \log \mathbb{P}[\theta_{f,i}] \\
&+ \sum_{i=1}^K \log \mathbb{P}[\theta_{g,i}] + \log \mathbb{P}[s_0] + \sum_{t=1}^T \log \mathbb{P}[s_t | s_{t-1}].
\end{aligned}$$

Then, minimizing  $J(\mathbf{y}, \mathbf{u}, \Theta, \mathbf{S})$  in (7) w.r.t.  $\Theta$ , whose components are defined in (10), it is equivalent to maximizing the logarithm of  $\mathbb{P}[\mathbf{y}, \mathbf{x}, \mathbf{S}, \Theta | \mathbf{u}]$ , which is hence also equivalent to maximizing  $\mathbb{P}[\mathbf{y}, \mathbf{x}, \mathbf{S}, \Theta]$ . ■

*Proof of Theorem 1:* From Proposition 1, we know that minimizing the loss function  $J(\mathbf{y}, \mathbf{u}, \Theta, \mathbf{S})$  defined in (7) with ingredients as in (10) is equivalent to maximizing  $Q(\Theta, \Theta^k)$  w.r.t.  $\Theta$ . Therefore, we only need to prove that applying the EKF process to each submodel minimizes the loss function

$J(\mathbf{y}, \mathbf{u}, \Theta, \mathbf{S})$ . For conciseness, only one iteration process of EKF is considered, while the others can be derived in the same way.

At the  $n$ -th step, the loss function can be written as

$$\begin{aligned}
& J(\mathbf{y}_{1:n}, \mathbf{u}_{1:n}, \Theta, \mathbf{S}_{1:n}) \\
&= \sum_{t=1}^n \ell(y(t), u(t), \theta_{f,s_t}, \theta_{g,s_t}) + \sum_{i=1}^K (r(\theta_{f,i}) + r(\theta_{g,i})) \\
&\quad + \mathcal{L}(\mathbf{S}) \\
&= J(\mathbf{y}_{1:n-1}, \mathbf{u}_{1:n-1}, \Theta^k, \mathbf{S}_{1:n-1}) \\
&\quad + \ell(y(n), u(n), \theta_{f,s_n}, \theta_{g,s_n}) + \mathcal{L}(s_n). \tag{30}
\end{aligned}$$

Let us introduce  $z_n := \{\mathbf{y}_{1:n}, \mathbf{u}_{1:n}, \Theta, \mathbf{S}_{1:n}\}$ . The loss function in (30) can hence be rewritten as

$$J(z_n) = J(z_{n-1}) + \ell(y(n), u(n), \theta_{f,s_n}, \theta_{g,s_n}) + \mathcal{L}(s_n).$$

The EKF is a recursive process consisting in two main parts. The first part consists in predicting the prior estimate of the parameter in (16). Similar to §III-A, we indicate with  $\hat{z}_n^-$  the prior estimate of  $z_n$  so that the prior loss can be rewritten as:

$$\begin{aligned}
& J^-(z_n) = J(z_{n-1}) + \ell(x(n+1), x(n), u(n), \theta_{f,s_n}) \\
&\quad + \frac{1}{2} \mathcal{L}(s_n) \tag{31}
\end{aligned}$$

In the second part, one updates the prior estimate by using the measurement data (17). To this end, let  $\hat{z}_n$  be the posterior estimate of  $z_n$ . Then, the posterior loss reads as:

$$\begin{aligned}
& J(z_n) = J^-(z_n) + \ell(y(n), x(n), u(n), \theta_{g,s_n}) + \frac{1}{2} \mathcal{L}(s_n). \tag{32}
\end{aligned}$$

Let us then consider the prediction step first. To apply the EKF, one has to linearize the nonlinear system (13) as in (14) and (15). Thus, the RNN-based system is locally equivalent to the following linear system:

$$\begin{aligned}
& x(t+1) = F(t)\nu(t) + \tilde{u}_x(t) + \zeta(t), \\
& y(t) = H(t)\nu(t) + \tilde{u}_y(t) + \xi(t), \\
& \nu(t) = \begin{bmatrix} x(t) \\ \vartheta(t) \end{bmatrix},
\end{aligned}$$

where  $\tilde{u}_x(t) = \mathcal{N}_{x,s_t}(x(t), u(t), \theta_{f,s_t}) - F(t)\nu(t)$ , and  $\tilde{u}_y(t) = \mathcal{N}_{y,s_t}(x(t), u(t), \theta_{g,s_t}) - H(t)\nu(t)$ . A standard method for solving linear equations is adopting the weighted least-square, which relies on the loss function  $\ell(x(n+1), x(n), u(n), \theta_{f,s_n}) = \frac{1}{2} \|x(n+1) - \mathcal{N}_{x,s_n}(x(n), u(n), \theta_{f,s_n})\|_{Q_x^{-1}}$ , where  $Q_x$  is a positive definite weight matrix (notice the abuse of notation with the  $\ell(\cdot)$  determining (7)). To alleviate notation, we omit the dependency for  $\mathcal{N}_{x,s_n}$  on its arguments, and define  $\nu_{1:n} := [\nu(1) \dots \nu(n)]^\top$ . Then, deriving the expression in (31) w.r.t.  $\nu_{1:n+1}$  leaves us with

$$\frac{\partial J^-(z_n)}{\partial \nu_{1:n+1}} = \begin{bmatrix} \frac{\partial J^-(z_{n-1})}{\partial \nu_{1:n}} - F(n)^\top Q_x^{-1}(x(n+1) - \mathcal{N}_{x,s_n}) \\ Q_x^{-1}(x(n+1) - \mathcal{N}_{x,s_n}) \end{bmatrix}.$$

Let the gradient be zero (i.e.,  $\nu(n) = \hat{\nu}^-(n)$ ), which yields (16). We derive next the covariance of the estimate  $\hat{\nu}(n+1)$ . Specifically, the Hessian matrix reads as:

$$\frac{\partial^2 J^-(z_n)}{\partial \nu_{1:n+1}^2} = \begin{bmatrix} \frac{\partial^2 J^-(z_n)}{\partial \nu_{1:n}^2} + F(n)Q_x^{-1}F^\top(n) & -F(n)^\top Q_x^{-1} \\ -Q_x^{-1}F(n) & Q_x^{-1} \end{bmatrix},$$

which in view of its positive definiteness, the one-step iteration of the Newton's method can be obtained through the gradient and the Hessian matrix:

$$\nu(n+1) = \nu(n) - \left( \frac{\partial^2 J^-(z_n)}{\partial \nu_{1:n+1}^2} \right)^{-1} \frac{\partial J^-(z_n)}{\partial \nu_{1:n+1}},$$

where the lower-right block of the inverse of the Hessian matrix is the covariance matrix of the prior estimation error. Then, we have  $P^-(n) = F(n)P(n-1)F(n)^\top + Q_x$ , which yields (19) with  $Q_x(t) = \text{diag}(\Sigma_1(t), \Sigma_\vartheta(t))$ .

Successively, we need to prove that the update part of the EKF process also minimizes the loss function in (32). Similar to the prediction part, the gradient and the Hessian matrix of  $J(\hat{z}_n)$  reads as:

$$\begin{aligned} \frac{\partial J(z_n)}{\partial \nu_{1:n}} &= \frac{\partial J^-(z_n)}{\partial \nu_{1:n}} - H(n)^\top Q_y^{-1}(y(n) - \mathcal{N}_{y,s_n}), \\ \frac{\partial^2 J(z_n)}{\partial \nu_{1:n}^2} &= \frac{\partial^2 J^-(z_n)}{\partial \nu_{1:n}^2} + H(n)Q_y^{-1}H^\top(n). \end{aligned}$$

Let  $\nu(n) = \hat{\nu}^-(n)$ . The gradient then takes the form:

$$\frac{\partial J(z_n)}{\partial \nu_{1:n}} = \begin{bmatrix} 0 \\ -H(n)^\top Q_y^{-1}(y(n) - \mathcal{N}_{y,s_n}) \end{bmatrix}. \quad (33)$$

Similarly, the covariance matrix of the posterior estimation error can be expressed as the right-block of the inverse of the Hessian matrix, which reads as follows:

$$P(n) = (P^-(n) + H(n)^\top Q_y H(n))^{-1}. \quad (34)$$

Let us denote the Kalman gain as  $\Gamma(n) = P^-(n)H^\top(n)[H(n)P^-(n)H^\top(n) + Q_y]^{-1}$ . Then, the covariance matrix is  $P(n)$  equivalent to  $(I - \Gamma(n)H(n))P^-(n)$  which yields (22).

Therefore, the EKF process minimizes the one-step loss function (31) and (32), and in turn minimizes the loss function (30) jointly. This concludes the proof. ■

## REFERENCES

- [1] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International conference on machine learning*, pages 136–145. PMLR, 2017.
- [2] Laurent Bako. Identification of switched linear systems via sparse optimization. *Automatica*, 47(4):668–677, 2011.
- [3] Mark P. Balenzuela, Adrian G. Wills, Renton Christopher, and Ninness Brett. Parameter estimation for jump markov linear systems. *Automatica*, 135:109949, 2022.
- [4] Roy Batruni. A multilayer neural network with piecewise-linear structure and back-propagation learning. *IEEE Transactions on Neural Networks*, 2(3):395–403, 1991.
- [5] Alberto Bemporad. Recurrent neural network training with convex loss and regularization functions by extended Kalman filtering. *IEEE Transactions on Automatic Control*, 68(9):5661–5668, 2022.
- [6] Alberto Bemporad, Valentina Breschi, Dario Piga, and Stephen P Boyd. Fitting jump models. *Automatica*, 96:11–21, 2018.
- [7] Valentina Breschi, Dario Piga, and Alberto Bemporad. Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica*, 73:155–162, 2016.
- [8] Raffaella Carloni, Ricardo G Sanfelice, Andrew R Teel, and Claudio Melchiorri. A hybrid control strategy for robust contact detection and force regulation. In *2007 American Control Conference*, pages 1461–1466. IEEE, 2007.
- [9] Antoni B Chan and Nuno Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):909–926, 2008.
- [10] Filippo Fabiani, Bartolomeo Stellato, Daniele Masti, and Paul J Goulart. A neural network-based approach to hybrid systems identification for control. *Automatica*, 2025. (In press).
- [11] Giancarlo Ferrari-Trecate, Marco Muselli, Diego Liberati, and Manfred Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.
- [12] Andrea Garulli, Simone Paoletti, and Antonio Vicino. A survey on switched and piecewise affine system identification. *IFAC Proceedings Volumes*, 45(16):344–355, 2012.
- [13] Arash Golabi, Nader Meskin, Roland Toth, and Javad Mohammadpour. A bayesian approach for lpv model identification and its application to complex processes. *IEEE Transactions on Control Systems Technology*, 25(6):2160–2167, 2017.
- [14] Massimo Guidolin. Markov switching models in empirical finance. In *Missing data methods: Time-series methods and applications*, pages 1–86. Emerald Group Publishing Limited, 2011.
- [15] Antti Honkela. *Nonlinear switching state-space models*. PhD thesis, Helsinki University of Technology, 2001.
- [16] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.
- [17] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [18] Jeffrey Humpherys, Preston Redd, and Jeremy West. A fresh look at the Kalman filter. *SIAM review*, 54(4):801–823, 2012.
- [19] Meng Jiao, Dongqing Wang, and Jianlong Qiu. A GRU-RNN based momentum optimized algorithm for SOC estimation. *Journal of Power Sources*, 459:228051, 2020.
- [20] Kon Johan, Roland Toth, Wijdeven Jeroen, Heertjes Marcel, and Oomen Tom. Guaranteeing stability in structured input-output models: With application to system identification. *IEEE Control Systems Letters*, 8:1565–1570, 2024.
- [21] Fabien Lauer. On the complexity of piecewise affine system identification. *Automatica*, 62:148–153, 2015.
- [22] Fabien Lauer, Gérard Bloch, and René Vidal. A continuous optimization framework for hybrid system identification. *Automatica*, 47(3):608–613, 2011.
- [23] Daniele Masti and Alberto Bemporad. Learning nonlinear state-space models using autoencoders. *Automatica*, 129:109666, 2021.
- [24] Henrik Ohlsson, Lennart Ljung, and Stephen Boyd. Segmentation of ARX-models using sum-of-norms regularization. *Automatica*, 46(6):1107–1111, 2010.
- [25] Dario Piga, Valentina Breschi, and Alberto Bemporad. Estimation of jump Box–Jenkins models. *Automatica*, 120:109126, 2020.
- [26] Dario Piga, Pepijn Cox, Roland Toth, and Vincent Laurain. Lpv system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53(C):329–338, 2015.
- [27] Riccardo Porreca, Samuel Drulhe, Hidde De Jong, and Giancarlo Ferrari-Trecate. Identification of parameters and structure of piecewise affine models of genetic networks. *IFAC Proceedings Volumes*, 42(10):587–592, 2009.
- [28] Vinay Prasad and B Wayne Bequette. Nonlinear system identification and model reduction using artificial neural networks. *Computers & Chemical Engineering*, 27(12):1741–1754, 2003.
- [29] Giulio Ripaccioli, Alberto Bemporad, Francis Assadian, Clement Dextreit, Stefano Di Cairano, and Ilya V Kolmanovskiy. Hybrid modeling, identification, and predictive control: An application to hybrid electric vehicle energy management. In *Hybrid Systems: Computation and Control: 12th International Conference, HSCC 2009, San Francisco, CA, USA, April 13–15, 2009. Proceedings 12*, pages 321–335. Springer, 2009.
- [30] Jacob Roll, Alberto Bemporad, and Lennart Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, 2004.
- [31] Anna Scampicchio, Alberto Giaretta, and Gianluigi Pillonetto. Nonlinear hybrid systems identification using kernel-based techniques. volume 51, pages 269–274. Elsevier, 2018.

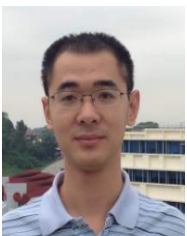
- [32] Thomas Schlegl, Martin Buss, and Günther Schmidt. A hybrid systems approach toward modeling and dynamical simulation of dextrous manipulation. *IEEE/ASME transactions on mechatronics*, 8(3):352–361, 2003.
- [33] Björn Schuller, Martin Wöllmer, Tobias Moosmayr, Günther Ruske, and Gerhard Rigoll. Switching linear dynamic models for noise robust in-car speech recognition. In *Pattern Recognition: 30th DAGM Symposium Munich, Germany, June 10-13, 2008 Proceedings 30*, pages 244–253. Springer, 2008.
- [34] Ji Shuyi, Zhang Zizhao, Ying Shihui, Wang Liejun, Zhao Xibin, and Yue Gao. Kullback–leibler divergence metric learning. *IEEE transactions on cybernetics*, 52(4):2047–2058, 2022.
- [35] Kelvin Tan, William J Parquette, and Meng Tao. A predictive algorithm for maximum power point tracking in solar photovoltaic systems through load management. *Solar Energy*, 265:112127, 2023.
- [36] Allan Timmermann. *Markov Switching Models in Finance*, pages 1–3. John Wiley & Sons, Ltd, 2015.
- [37] J Umenberger, J Wågberg, I.R Manchester, and T.B. Schon. Maximum likelihood identification of stable linear dynamical systems. *Automatica*, 96:280–292, 2018.
- [38] René Vidal, Alessandro Chiuso, and Stefano Soatto. Observability and identifiability of jump linear systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 4, pages 3614–3619. IEEE, 2002.
- [39] Jun Xu, Xiaolin Huang, and Shuning Wang. Adaptive hinging hyperplanes and its applications in dynamic system identification. *Automatica*, 45(10):2325–2332, 2009.
- [40] Pan Zhaojie, Li Chenyu, Plaza Antonio, Chanussot Jocelyn, and Hong Danfeng. Hyperspectral image classification with mamba. *IEEE Transactions on Geoscience and Remote Sensing*, page 1, 2024.



**Filippo Fabiani** received the B.Sc. degree in bio-engineering, the M.Sc. degree in automatic control engineering, and the Ph.D. degree in automatic control from the University of Pisa, Pisa, Italy, in 2012, 2015, and 2019, respectively. He is currently an Assistant Professor with the IMT School for Advanced Studies Lucca, Lucca, Italy. In 2018–2019, he was Postdoctoral Research Fellow with the Delft Center for Systems and Control, TU Delft, Delft, The Netherlands, while in 2019–2022, he was a Postdoctoral Research Assistant with the Control Group, Department of Engineering Science, University of Oxford, Oxford, U.K. His research interests include game theory, optimization and control of complex uncertain systems, with applications in power networks and automated driving.



**Yanxin Zhang** received his B.Sc. degree in the School of Mathematics and Statistics from the Huazhong University of Science and Technology, Wuhan, China, in 2019, and the M.Sc. degree in the School of Science from Jiangnan University, Wuxi, China, in 2023. He is currently pursuing the Ph.D. degree with the School of Automation, Beijing Institute of Technology, Beijing, China. His current research interests include system identification and parameter estimation of switching systems.



**Chengpu Yu** received the B.E. and M.E. degrees in electrical engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2014. He was with the Internet of Things Lab at Nanyang Technological University as a Research Associate, from 2013 to 2014, and with Delft Center for Systems and Control as a PostDoc, from 2014 to 2017. Since 2018, he has been with the Beijing Institute of Technology, Beijing, China, as a Full Professor. His research interests include system identification, distributed optimization, and optical imaging.