Hybrid system identification using a mixture of NARX experts with LASSO-based feature selection

Alessandro Brusaferri[a,b], Matteo Matteucci[b], Pietro Portolani[a], Stefano Spinelli[a,b], Andrea Vitali[a]
[a]*CNR-Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, Milan, Italy*
[b]*Politecnico di Milano - Department of Electronics, Informatics and Bioengineering, Milan, Italy*
$name.surname$@stiima.cnr.it, $name.surname$@polimi.it

***Abstract*—The availability of advanced hybrid system identification techniques is fundamental to extract knowledge in form of models from data streams. Starting from the current state of the art, we propose an approach based on a specialized architecture, conceived to address the peculiar integration of nonlinear dynamics and finite state switching behavior of hybrid systems. Following the Mixtures of Experts concept, we combine a set of Neural Network ARX (NNARX) models with a Gated Recurrent Units network with softmax output. The former are exploited to map specific nonlinear dynamical models representing the behavior of the system in each discrete mode of operation. The latter, operating as a neural switching machine, infers the unobserved active mode and learns the state-transition logic, conditioned on input-output data sequences. Besides, we integrate a LASSO based input features and model selection mechanism, aimed to extract the most informative lags over the sequences for each NNARX and calibrate the modes to be employed. The overall system is trained end-to-end. Experiments have been performed on a benchmark hybrid automata with nonlinear dynamics and transitions, showing the capability to achieve improved performances than conventional architectures.**

***Index Terms*—System Identification, Hybrid systems, Mixture of Experts, Neural Network, Automatic feature selection, LASSO**

I. Context and motivation

The widespread digitalization and the consequent implementation of Cyber Physical Systems in different domains from Industry4.0 and Smart Grid to Smart Cities and healthcare to cite a few, are posing new challenges and further increasing the demand for efficient and scalable system identification techniques [1]. While a broad spectrum of methods are available to tackle systems characterized by continuous dynamics, CPSs have an intrinsic hybrid nature, characterized by the non-trivial interaction of discrete and continuous elements, requiring enhanced hybrid system identification techniques [2]. Hybrid system identification targets the estimation, from available input output data sequences, of both the discrete modes and the sub-models governing the dynamics of the continuous state for each mode [3]. Automated model learning is particularity challenging for hybrid systems exhibiting complex nonlinear behavior and switching rules between dynamical regimes [4]. Most of the proposed approaches have focused on switched affine and piecewise affine models, constituted by a finite set of linear subsystems [1]. Due to their theoretical universal approximation capability, neural networks have also been widely investigated to learn dynamical models from data, showing the capability to extract mappings of complex nonlinear dynamical systems (see e.g., [5]) as well as representations of finite state machines (see e.g, [6]). Both feed-forward (FFNN) and recurrent architectures (RNN) have been employed for this purpose. In principle, they are able to identify also hybrid dynamical systems; however, the incorporation of specific structures into neural models, by considering the specific characteristics of the target problem, is receiving a lot of attention as a way to increase performance, sample efficienty, and explainability [7].

To target the identification of hybrid systems, a neural network is expected to learn both the finite state machine governing regime switching and a specific nonlinear dynamical model for each regime. Such pattern resembles the Mixture of Experts (MoE) model proposed in the seminal paper of Jacobs et al, generalizing finite mixture models to address problem decomposition [8]. MoE are composed by a set of regression functions and a gating mechanisms soft-partitioning the input space, to capture the sub-regions where the individual experts are reliable [9], and supporting the identification of piece-wise continuous systems. Despite being conceived to process static data, MoE models have been exploited also for the identification of non-stationary time series, including input output Hidden Markov Models [10]. Major critical issues of MoE model regard model selection (e.g., number of experts) and feature selection. The latter usually need to be performed specifically for both the mixing component and each expert (e.g., to capture mode related nonlinear map to input regressors). In practice, expensive manual procedures are often employed, by leveraging on the developer expertise [9]. In general, accuracy is often improved by discarding irrelevant and redundant features [11]. The Least Absolute Shrinkage and Selection Operator operator (LASSO) has been proposed for such purpose for linear MoE, often employing Gaussian experts [12].

In this work, we target Switching Nonlinear Auto-Regressive with Exogeneous inputs (SNARX) systems, representing a broad class of hybrid problems, including CPS. To this end, we propose an approach based on a specialized network architecture, including a LASSO-based automated feature selection mechanism. Our scope is to identify the overall behavior of the system in a single step, covering the

estimation of the discrete modes, the related switching logic as well as the nonlinear subsystems dynamics with unknown structure. The proposed method is applied to a benchmark hybrid system identification problem, showing improved performance than conventional network architectures. The rest of the paper proceeds as follows: Section II deepens the related works and open issues; Section III reports the developed network architecture and training method, then Section IV summarizes the results achieved.

II. Related works and contribution

As reported in [1], a broad spectrum of hybrid systems identification techniques has been proposed in the last two decades, mostly focusing on Piece-Wise Affine (PWA) systems with linear transition rules. In this context, the approaches that attained particular attention include the algebraic, clustering based, bounded-error, mixed integer programming and Bayesian learning. A detailed review is reported in [13]. More recently, authors in [14] proposed a technique based on multi-class linear separation and recursive clustering. Far fewer works address the nonlinear case [15]. An approach to identify piece-wise smooth and switched nonlinear systems based on Support Vector Machines (SVM) is proposed in [3], but it suffers limitations on the treatable data size. In [16] a reduced-size kernel models is introduced, assuming that the number of submodels and their regressors are known. In [15], the SVM model is enhanced by a robust sparsity term to control model complexity, extending previously proposed sparsification techniques for affine systems. A method for the segmentation of nonlinear systems is proposed in [17], exploiting a least squares SVM with the sum-of-norms regularization. Hybrid identification is approached by symbolic regression in [18] and complemented with model selection through sparsification in [4]. A two step technique is proposed in [1], including a clustering based subsystem inference phase followed by a transition inference phase, which considers a linear combination of over-determined dictionary matrices.

Neural networks have been investigated for hybrid system identification in [19] and [20], proposing respectively a feedforward network to learn a class of hybrid systems, and a recurrent network based switching Manifolds supervisor between local sub-network. Switching density networks [7] have been proposed to predict the parameters of hybrid control laws. Research efforts have been dedicated to learn piece-wise continuous time-series by MoE (see e.g,. [9] for a detailed review). Most notably, authors of [21] derived an extension to the original MoE based on a n-th order Markov model, introducing recurrence in the gate.
In a grammar inference context, authors of [10] proposed a recurrent version of the MoE, called the Input Output Hidden Markov Model architecture, composed by output networks and recurrent state networks. The internal state is computed as a linear combination of the outputs of the state networks, gated by the previously computed internal state. In [22], time series with switching regimes are tackled by a multilayer perceptron based gate combining a set of feed-forward neural network experts.
The aforementioned studies mainly target architectural patterns and related learning procedures. Previous works also addressed LASSO-based features/model selection by focusing on linear-in-parameters models, thus leading to simplified convex optimization problems (see e.g., [11]). However, to the best of our knowledge, the investigation of integrated methods supporting end-to-end learning including features/model selection in a MoE framework is still lacking within hybrid systems identification literature.

In this paper, we propose a hybrid system identification approach based on a specialized architecture constituted of a set of Neural Network-ARX (NN-ARX) models governed by a switching machine based on a Gated Recurrent Unit (GRU) network. Moreover, we integrate the log-likelihood function with weighted LASSO terms dedicated to inputs, hidden network layers and gates, performing automated feature selection. The main goal is to increase prediction performance in contexts where specific knowledge is not available (e.g., number of discrete modes, features lags, etc.), as detailed in the following sections.

III. Methods

Depending on the specific characteristics of the hybrid identification problem at hand, the interaction of discrete and continuous counterparts can be characterized by different modeling approaches, from piece-wise affine systems - determined by a polyhedral partition of the continuous space of the regressors -, to piece-wise nonlinear systems - with subdomains not restricted to polyhedral - up to the broader class of arbitrary switching systems [2]. Discrete states typically characterize the different operating modes of the system or changes in the dynamics, e.g., thresholds and dead-zones, behavioral switching, physical limits, etc. [13]. In jump systems, the active discrete state is provided as an exogenous input, thus the hybrid identification problem can be recast to a classical nonlinear identification problem, e.g., by learning each sub-model separately. However, in most of applications, the timed sequences of discrete states have to be inferred, as well as the initial conditions.

In this work, we target the broad class of Switching Nonlinear Autoregressive with Exogeneous inputs (SNARX) systems, defined as a composition of input-output subsystems in discrete time as follows:

$$\begin{cases} y(t) &= f_1(x_1(t)) + e(t) \quad \text{, when} \quad q(t) = 1 \\ & \dots \\ y(t) &= f_{\bar{q}}(x_{\bar{q}}(t)) + e(t) \quad \text{, when} \quad q(t) = \bar{q} \end{cases} \tag{1}$$

where $y(t)$ is the system output at sample time t, $\left\{x_j(t) = [\mathbf{y}_{t-h_j}, \mathbf{u}_{t-k_j}]\right\}_{j=1}^{\bar{q}}$ and $\left\{f_j[.] : \mathbb{R}^{n_{x_j}} \rightarrow \mathbb{R}\right\}_{j=1}^{\bar{q}}$ the collections of regression vectors and the nonlinear function mappings valid within each discrete state (or mode) $q(t) \in \{1, ..., \bar{q}\}$, $e(t)$ is a noise term assumed to be Gaussian distributed with zero mean and the same variance σ^2. It is worth noting that a single discrete state is active in each sample time. Besides, the number of lagged input $\mathbf{u}_{t-k_j}$

and measured outputs $\mathbf{y}_{t-h_j}$ are in general specific of each NARX submodel. The switching between discrete states is governed by a finite set of transitions with boolean conditions, unknown functions of both lagged input and output as well as temporized events (see e.g., [2] for further details). We formulate the method for a Single Input - Single Output (SISO) system to simplify notation, however it can be straightforwardly extended to the Multi Input-Multi Output (MIMO) case. The aim of the identification process is to learn the one-step prediction of the hybrid system (1) from an input-output dataset $\{x_i(t), y_i(t)\}_{i=1}^{N}$ of length N.

A. Neural network architecture

The network architecture, depicted in Figure 1, has been designed following the Mixture of Experts (MoE) concept. The rationale behind this choice is twofold. On the one hand, we aim to foster the identification of specialized sub-models (i.e., experts) characterizing the dynamics of the hybrid system within each discrete state. On the other hand, we target the implementation of hierarchical patterns shaped according to the specific characteristics of the problem class at hand. Specifically, the developed MoE architecture is formalized as:

$$\begin{aligned} P(y(t)|x(t),\theta) &= \sum_{j=1}^{n_e} P(y(t), j|x(t),\theta) \\ &= \sum_{j=1}^{n_e} g_j(x(t),\theta_g) P(y(t)|j, x(t),\theta_e) \end{aligned} \tag{2}$$

where $g_j(x(t),\theta_g)$ represents the probability of each expert given the inputs regressors, $n_e \in \mathbb{R}$ the configured number of experts and $P(y(t)|j,x(t),\theta_e)$ the expert-wise likelihood function. $\theta_g \in \mathbb{R}^{n_{\theta_g}}$ and $\theta_e \in \mathbb{R}^{n_{\theta_e}}$ summarizes the parameters sets of the mode gating and experts networks respectively, which size depends on the specific architectural shapes.
Considering the Gaussian assumption over the noise reported above, the latter is defined as:

$$P(y(t)|j,x(t),\theta_e) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(f_j(x(t))-y(t))^2} \tag{3}$$

Following the NARX formulation stated above, we chose the experts as feedforward neural networks with regressors input layers (i.e, NNARX). The discrete state network, aimed to learn the transition dynamics between the hybrid system modes, is modeled by means of a recurrent neural network (RNN). RNN exploits its capability to structure compressed representations of arbitrary long input sequence within the hidden state. Hence, such gating parametrization is expected to learn complex transition rules including short and long-term swithcing events and conditioning on input features. To address the vanishing gradient issue of traditional RNNs, we included Gated Recurrent Units cells (GRU), providing a computationally cheaper alternative to Long Short-Term Memory units. The investigation of alternative network architectures (e.g. recurrent network based experts, etc.) is left to future extensions of the present work. Formally, the network

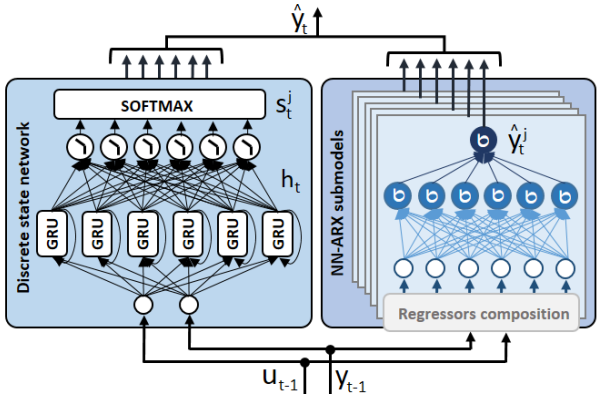

Fig. 1. Developed MoNNARX network architecture

components are defined as follows, where we include a single layer both for the recurrent and the feed-forward networks and employ subscripted time to ease notation:

$$\begin{aligned} z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\ r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\ h_t &= (1 - z_t) \odot \tanh(W_h x_t + U_h(r_t \odot h_{t-1} + b_h)) \\ &\quad + z_t \odot h_{t-1} \\ m_t &= \sigma(W_m h_t + b_m) \\ s_t^j &= \frac{\exp(m_t^j)}{\sum_{e=1}^{n_e} \exp(m_t^e)}, \qquad k_t^j = \sigma(W_k^j x_t + b_k^j) \\ y_t^j &= W_y^j k_t^j + b_y^j, \qquad y_t = \sum_{j=1}^{n_e} y_t^j . s_t^j \end{aligned} \tag{4}$$

where z_t defines the update gate, r_t the reset gate, $W_z, W_r, W_h \in \mathbb{R}^{n_h \times n_x}$, $U_z, U_r, U_h \in \mathbb{R}^{n_h \times n_h}$, $W_m \in \mathbb{R}^{n_m \times n_h}$, $W_k^j \in \mathbb{R}^{n_k \times n_x}$, $W_y \in \mathbb{R}^{n_y \times n_k}$ the weight matrices and $b_z, b_r, b_h \in \mathbb{R}^{n_h}$, $b_m \in \mathbb{R}^{n_m}$, $b_k^j \in \mathbb{R}^{n_k}$, $b_y^j \in \mathbb{R}^{n_y}$ the bias vectors. The gates include an element-wise sigmoid activation, $\sigma(z) = \frac{1}{1+e^{-z}}$, while an hyperbolic tangent activation $\tanh(x) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$, is used for the hidden state equation. The layer with output m_t has been also introduced to enable mapping complex nonlinearities within the transition function, since despite being deep when unfolded in time, RNNs still results shallow on certain computational paths. We employed $relu(z) = max(0, z)$ units since sparsely activated.

It is worth noting that the vector output s_t^j of the softmax layer, implementing the RNN-based experts gating $g_j(x(t),\theta_g)$, is a smooth version of the winner take all model, which is employed to obtain a trainable network architecture. Hence, it approximates, by training, the conditioned switching mechanisms between the modes-specific NARX predictions y_t^j through soft-partitioning. To enforce learning a sparse selector, we introduce a LASSO term on the layer outputs, as reported in the next subsection. Besides, the specific structures of experts and gating networks (e.g. in terms of layers, neurons in each layer, Back Propagation Through Time of the RNN, etc.) represent hyperparameters that must be properly tuned on the specific application at hand. To this end, we exploited cross-validation, as detailed in Section IV.

B. Network training approach

Typically, system identification models are trained by maximizing the likelihood of the training data, assuming i.i.d. samples and Gaussian noise.

Still, automated feature and model selection mechanisms have to be integrated, in order to achieve an efficient and scalable system identification technique, avoiding extensive manual trial and error. The selection of the features subset providing support to explain the problem is particularity critical for hybrid systems since typically characterized by multiple modes, specific timed correlations on both states and transitions, unknown number of change-points and related lag positions. Besides, working on a lower dimensional space with less features redundancies often support the performance of training procedures. Moreover, even by manually stating the features subset, there is still an infinite number network models, with increasing number of expert sub-components, capable of explaining the data [23].

To address such issues, we embed weighted LASSO-based sparsification terms within the conventional maximum likelihood training function related to: the input layers of the networks (with weight α) to perform selection of the features from the time series including both exogenous variables and past values of the measured output; the hidden layers of both networks sub-components (with weight β) to control complexity and consequent generalization capacity; the output activity of the discrete state selection network (with weight γ) to foster the learning procedure to approach input conditioned winner-take-all patterns; leading to the following objective:

$$\hat{\theta} = \underset{\theta}{argmin}\,[L(\theta) + \alpha\|W_X\|_1 + \beta\|W_H\|_1 + \gamma\|s_t\|_1] \quad (5)$$

where we used $L(\theta)$, W_X, W_H to summarize the likelihood, weights of input and hidden layers to simplify notation.

In general, specific penalties can be introduced for each layer of the network thus increasing control on the related regularization effects. However, the hyperparameter space increases accordingly, thus impacting on the grid of cross-validation. To balance control and tuning complexity, in this work we used single weights α, β for the input and net complexity since our major aim is to investigate input features and model selection. The exploration of more granular hyperparametrizations, as well as the integration of further regularization terms, are left to future extensions of the present work.

As introduced above, the major strength of LASSO operator resides in its capability to encourage sparse solutions, as opposed to alternative regularization methods based on quadratic forms (as ridge/weight decay) achieving contractions to small values, but not exactly zero. Even so, the non-differentiability at zero limits the straight application of conventional gradient based training algorithms. Such issue has been partially addressed in previous studies on hybrid system identification by focusing on linear-in-parameters models.

By retaining the representation capability of the full nonlinear-in-parameters form, two major families of learning approaches can be exploited [24]. The former recast to a constrained optimization problem, e.g., by introducing a set supplementary slack-variables to tackle the absolute values signs. The latter maintain the unconstrained optimization form, often exploiting smooth proxies, sub-gradients or thresholding. In this work, we implement the sub-gradient approach since computationally cheaper and more scalable than constrained formulation based methods [25]. The investigation of alternative solution techniques is foreseen as future extension of the present work. The overall network is trained end-to-end and the weighting parameters of the LASSO terms are tuned by a grid search procedure in cross-validation, as reported in the following section.

IV. Experiments and results

In this paper, we apply the method discussed in Section III on a benchmark nonlinear hybrid system stated opportunely to combine several challenging features both in state transitions and internal dynamics. The Multiple Input-Single Output (MISO) system is characterized by four discrete modes $q \in \mathcal{Q}$, in which the dynamical system evolution is described by a non-linear map $y(t) = f_j(x(t)) + e(t)$ for all $j \in \mathcal{Q}$. The regression vector $x(t) = [\mathbf{y}_{t-h}, \mathbf{u}_{1,\,t-k}, \ldots, \mathbf{u}_{m,\,t-k}]$ where $\mathbf{y}_{t-h} = [y(t-1), \ldots, y(t-h)]'$ and $\mathbf{u}_{m,\,t-k} = [u_m(t-1), u_m(t-1), \ldots, u_m(t-k)]'$ collect the lagged output measurements up to the time h and past values of the m^{th} input, with lag window k.

The nonlinear maps are reported in Table I. Note that to simplify the notation, here the lag subscript $\mathbf{y}_{t-h}, \mathbf{u}_{j,\,t-k}$ is removed: for all the modes $h = 3$ and $k = 2$. Moreover, in every mode the hidden state is incremented by $\tau(t) = \tau(t-1) + 1$. A common noise parameter have been applied as $\epsilon = 0.01$. The discrete-time hybrid automaton is depicted in Figure 2, where the vertices are the discrete modes while the edges of the directed graph represent the feasible transitions. These transitions are activated by guard conditions, reported in Table I. It is worth noting that these are defined by general Boolean expressions that include threshold conditions on nonlinear functions of output, inputs and their time derivatives. In addition, time-enabled transitions are also present: a hidden internal state is used in each mode to model a required dwell-time. In particular, the presence of time derivatives and dwell-time transitions requires the capability of the switching machine to deal with conditions based also on past data. The

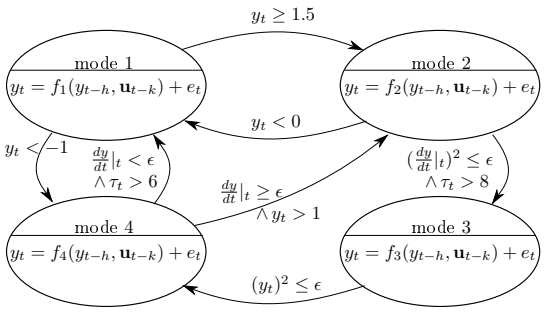

Fig. 2. Nonlinear hybrid automaton model

TABLE I
NONLINEAR MAPS AND GUARDS IN EACH MODE

Mode q	Nonlinear Map
1	$f_1 = a\,\mathbf{y} + b\,(\text{diag}(\mathbf{u}_1)\mathbf{u}_1)^{\odot\frac{1}{3}} + c\,\mathbf{u}_2^{\odot 3}$ where $a = [0.871, -0.235, 0.018]$, $b = [0.383, 0.068, -0.040]$ and $c = [1.329, -0.859, 0.136]$
2	$f_2 = a\,\mathbf{y} + b\,\text{diag}(0.8\mathbf{u}_1)(0.3\mathbf{u}_2) + c\,\mathbf{u}_2^{\odot 2}$ where $a = [1.646, -0.670, 0]$, $b = [0.414, -0.406, 0]$ and $c = [0.213, 0.192, 0]$
3	$f_3 = a\,\text{diag}(\mathbf{y})\,(0.15\mathbf{u}_2)^{\odot -1} + b\,\mathbf{u}_1^{\odot 2}$ where $a = [1.47624603635867, 0, 0]$, $b = [0.048, 0.110, 0.015]$
4	$f_4 = a\,\mathbf{y} + b\,\text{diag}(\mathbf{u}_1)\mathbf{u}_2 + c\,\mathbf{u}_1$ where $a = [0.455, -0.034, 0.001]$, and $b = [-0.185, 0.045, 0.009]$, and $c = [0.144, 0.116, 0.004]$
Notation:	Hadamard power: for $\alpha \in \mathbb{Q}$, $\mathbf{x}^{\odot\alpha} = [x_1^\alpha, \ldots, x_n^\alpha]'$

Mode q	Transition Guards		Reset
1	$q_{1\to 2}$	$y(t) > 1.5$	$\tau(t) = 0$
	$q_{1\to 4}$	$y(t) < -1$	
2	$q_{2\to 1}$	$y(t) < 0$	$\tau(t) = 0$
	$q_{2\to 3}$	$(\frac{dy}{dt})^2 < \epsilon \wedge \tau(t) > 8$	
3	$q_{3\to 4}$	$y(t)^2 < \epsilon \wedge u_1(t) > 0$	$\tau(t) = 0$
4	$q_{4\to 1}$	$\frac{dy}{dt} < \epsilon \wedge \tau(t) > 6$	$\tau(t) = 0$
	$q_{4\to 2}$	$\frac{dy}{dt} > \epsilon \wedge y(t) > 1$	

employed dataset, generated by a Matlab implementation of the benchmark hybrid system, is constituted by an overall sequence of 10000 input-output measurements.

The overall dataset is divided into training, validation and test sets composed of 60%, 30%, 10% of the samples respectively. Then, the training and validation sequences have been processed to construct batches of ordered samples used to train the network by supervised learning. To this end, the overall sequences are processed by sliding a window. The width of the window is a tunable hyperparameter, definining the extension of the search space considered for features selection from the raw input sub-sequences during training. In general, specific search windows could be configured for each subcomponents of the overall model (e.g., different length for NARX experts and discrete state network). However, in this work we employed a common hyperparameter, to reduce the dimensions of the search grid. Hence, the search window width is configured to cover the maximum lag horizon needed to capture specific substate dynamics or transition rules, considering both input and past-output data. The same hyperparameter configuration is set to the extension of Back Propagation Through Time and to build the NARX regressors $\{h_j, k_j\}_{j=1}^n$. Then, thanks to the developed automated feature selection mechanism, the unnecessary inputs/lags are excluded during training specifically for each subcomponents. The test set is processed only during the final one-shot experiments, employing the weight obtained by cross-validation.

The neural networks is deployed by means of Tensorflow 2.0, by a custom Keras model. Summarizing, the hyperparameters set includes the number of layers and units in the networks, the width of the sliding a window, the maximum number of experts to be employed (the exact subset is then chosen during training by the network exploiting the related selector) and the LASSO weights related to the input, output and hidden selectors. Network training is performed over 150 epochs with an early stop patience of 20 epochs (i.e., interrupting training when the objective stops decreasing) with a mini-batch size of 32 samples. We employ the Adam algorithm, conceived to tackle noisy and sparse gradients, as expectable in our case. Since the main goal of this work was to exploit whether the introduction of the MoNNRX specialized hierarchical structure support improved performance than conventional networks under comparable configurations, an extensive hyperparameter search has not been carried out. Future extension of this work will focus on the fine-tuning of those parameters and how optimization methods such as metaheuristics can be used to find their optimal values.

TABLE II
TEST SET RESULTS

	$Experts$	α	β	γ	$sMAPE$
FF-NNARX	-	0	1e-4	-	0.23
FF-NNARX	-	1e-2	1e-4	-	0.29
MoNNARX	4	1e-2	1e-4	1e-2	0.16
MoNNARX	10	0	1e-4	0	0.28
MoNNARX	10	1e-2	1e-4	1e-1	0.16

For final test set experiments, we employed the following setup: discrete state network with a recurrent layer of 30 GRU units stacked with a dense layer of 10 units; 10 NNARX with a dense layer with 30 sigmoid units stacked with a linear layer; window width of 10. Prediction performances have been evaluated by means of the symmetric Mean Absolute Percentage Error (sMAPE) to avoid the sensitivity to small values of conventional MAPE and obtain scale independent metrics. Table II reports the results obtained on the test set, while Figure 3 plots the predictions vs targets data. First of all, we performed experiments by a conventional feedforward architecture, capable to learn both continuous and finite state dynamics in an integrated representation, as shown in the previous studies reported in Section II. Both models works on the same input data space, thus the neural network operates as an overall NNARX structuring all modes within the hidden space. By cross validation, we did not observe sensible improvements by increasing the number of hidden units beyond 100 within the FF-NNARX, thus we used such configuration in test. To achieve comparable results, we included the input and network regularization components also in the FF-NNARX. Notably, the FF-NNARX achieved good prediction performances, thus learning both the different continuous dynamics of the modes and the discrete state switching. Surprisingly, the inclusion of the input feature selection term did not decrease the error. Principally, this was induced by the input layer sharing between the different modes embedded within the FF-NNARX, with consequent difficulties in performing features selection for each discrete state by a unique mechanism. Afterwards, we tested the MoNNARX model by setting the number of experts equal to the mode size of the benchmark problem, thus fostering the identification of one mapping, and related feature selection, for each state.

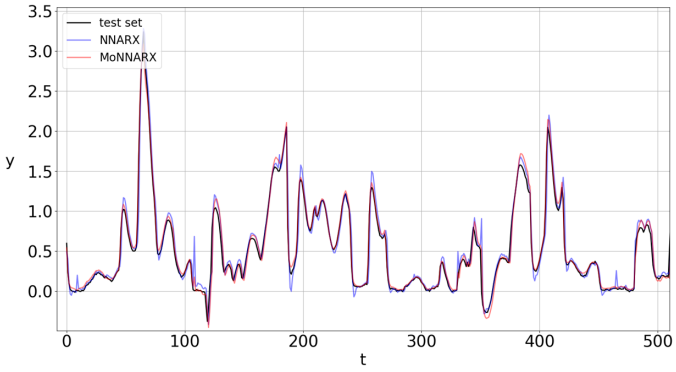

Fig. 3. Predicted vs target data over the test set

Finally, we experimented a MoNNARX network with 10 NNARX, to check eventual performance degradation (e.g., due to overfitting) and mode selection when the actual number is unknown, as common in real settings.
Notably, we obtained a sensible increase in prediciton accuracy with MoNNARX, as compared to the FF-NNARX model, thus showing that the introduction of a specialized network architecture improves the hybrid system identification. Besides, we observed stable performances even by doubling the number of NNARX experts than the target system modes. We report also the results obtained by setting $\alpha, \gamma = 0$, showing the importance of the LASSO-selection mechanism to achieve effective mixture of NARX experts architectures in practical applications where indications on the modes/features to be used is not available.

V. Conclusion and Next Steps

In this work we focused on the identification of one-step prediction models of hybrid systems from input-output data sequences. The complexity of such problem depends on which elements are assumed to be known a priori [13], e.g., number of modes, input feature, etc. Considering the requirements of real world applications, where such information are typically not available, we targeted the inference of the overall system, including the nonlinear dynamics related to each mode as well as the discrete states transitions. Besides, we considered as unknown also the set of input features lags, the currently active state, as well as the number of discrete states. To this end, we have proposed a hybrid system identification approach based on a specialized neural network architecture constituted of a set of Neural Network-ARX models governed by a switching machine based on a Gated Recurrent Unit network. Then, we included a LASSO-based automated feature/model selection mechanism, avoiding the complex manual procedures that has to be performed for each dynamical mode. By application to a benchmark hybrid system identification problem, we showed that the specialized hierarchical structure achieve improved performances than conventional general purpose network architectures, and we showed that the LASSO-selection mechanism is crucial to achieve effective mixture of NARX experts architectures when prior information is lacking, as common in real-life applications. Next developments will include the exploration of further specialized network architectures, the integration of optimization techniques for hyperparameter tuning, the investigation of rule-extraction techniques to enhance neural network explainability, and the application to practical case studies.

References

[1] Y. Yuan, X. Tang, W. Zhou, W. Pan, X. Li, H.-T. Zhang, H. Ding, and J. Goncalves, "Data driven discovery of cyber physical systems," *Nature Communications*, vol. 10, no. 1, p. 4894, 2019.

[2] J. Lunze and F. Lamnabhi-Lagarrigue, *Handbook of Hybrid Systems Control: Theory, Tools, Applications*. Cambridge University Press, 2009.

[3] F. Lauer and G. Bloch, "Switched and piecewise nonlinear hybrid system identification," in *Hybrid Systems: Computation and Control*, 2008.

[4] N. M. Mangan, T. Askham, S. L. Brunton, J. N. Kutz, and J. L. Proctor, "Model selection for hybrid dynamical systems via sparse regression," *Proceedings of the Royal Society*, vol. 475, 2019.

[5] A. Brusaferri, M. Matteucci, P. Portolani, and S. Spinelli, "Nonlinear system identification using a recurrent network in a bayesian framework," in *2019 IEEE 17th INDIN*, vol. 1, pp. 319–324, July 2019.

[6] H. Jacobsson, "Rule extraction from recurrent neural networks: A taxonomy and review," *Neural Computation*, vol. 17, 2005.

[7] M. Burke, Y. Hristov, and S. Ramamoorthy, "Hybrid system identification using switching density networks," *CoRR2019*, vol. abs/1907.04360.

[8] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive mixture of local expert," *Neural Computation*, vol. 3, pp. 78–88, 02 1991.

[9] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1177–1193, Aug 2012.

[10] Y. Bengio and P. Frasconi, "An input output hmm architecture," *Advances in Neural Information Processing Systems*, vol. 7, 12 1995.

[11] B. Peralta and A. Soto, "Embedded local feature selection within mixture of experts," *Information Sciences*, vol. 269, pp. 176 – 187, 2014.

[12] F. Chamroukhi, F. Lecocq, and H. Nguyen, *Regularized Estimation and Feature Selection in Mixtures of Gaussian-Gated Experts Models*, pp. 42–56. 01 2020.

[13] A. Garulli, S. Paoletti, and A. Vicino, "A survey on switched and piecewise affine system identification," vol. 45, no. 16, 2012. 16th IFAC Symposium on System Identification.

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

[15] V. Le, F. Lauer, L. Bako, and G. Bloch, "Learning nonlinear hybrid systems: from sparse optimization to support vector regression," 2013.

[16] V. L. Le, G. Bloch, and F. Lauer, "Reduced-size kernel models for nonlinear hybrid system identification," *IEEE Transactions on Neural Networks*, vol. 22, pp. 2398–2405, Dec 2011.

[17] T. Falck, H. Ohlsson, L. Ljung, J. A. Suykens, and B. D. Moor, "Segmentation of time series from nonlinear dynamical systems," pp. 13209 – 13214, 2011. 18th IFAC World Congress.

[18] D. L. Ly and H. Lipson, "Learning symbolic representations of hybrid dynamical systems," *J. Mach. Learn. Res.*, vol. 13, no. 1, 2012.

[19] N. Messai, J. Zaytoon, and B. Riera, "Using neural networks for the identification of a class of hybrid dynamic systems," *IFAC Proceedings Volumes*, vol. 39, no. 5, pp. 217 – 222, 2006.

[20] J. E. Velázquez-Velázquez, R. Galván-Guerra, and I. S. Baruch, "Hybrid recurrent neural network for nonlinear hybrid dynamical systems identification," in *2011 8th CCE*, Oct 2011.

[21] T. W. Cacciatore and S. J. Nowlan, "Mixtures of controllers for jump linear and non-linear plants," in *NIPS*, 1993.

[22] A. S. Weigend, M. Mangeas, and A. N. Srivastava, "Nonlinear gated experts for time series: discovering regimes and avoiding overfitting," *International journal of neural systems*, vol. 6 4, pp. 373–99, 1995.

[23] L. Bako, "Identification of switched linear systems via sparse optimization," *Automatica*, vol. 47, no. 4, pp. 668 – 677, 2011.

[24] S. Scardapane, D. Comminiello, A. Hussain, and A. Uncini, "Group sparse regularization for deep neural networks," *Neurocomputing*, 2017.

[25] A. Brusaferri, L. Fagiano, M. Matteucci, and A. Vitali, "Day ahead electricity price forecast by narx model with lasso based features selection," in *2019 IEEE 17th Industrial Informatics conference*.