Journal of Control, Automation and Electrical Systems (2019) 30:921–942
https://doi.org/10.1007/s40313-019-00528-0

Multivariable State-Space Recursive Identification Algorithm Based on Evolving Type-2 Neural-Fuzzy Inference System

Anderson Pablo Freitas Evangelista[1] · Ginalber Luiz de Oliveira Serra[2]

Received: 14 February 2019 / Revised: 9 August 2019 / Accepted: 24 September 2019 / Published online: 15 October 2019

Abstract

In this paper, a novel approach for state-space evolving type-2 neural-fuzzy identification of multivariable dynamic systems is proposed. According to adopted methodology, conditions for creating and merging clusters are used to perform the structural adaptation of the neural-fuzzy model. The center and shape of each cluster are estimated, defining all rules in the interval type-2 neural-fuzzy inference system. The degree of uncertainty on the shape of type-2 membership functions is computed through an extended Kalman filter-based learning mechanism. Once the type-2 membership functions (upper and lower membership values) are estimated, the fuzzy Markov parameters are computed from experimental data, and for each incoming information, the parameters of state-space linear models in the consequent proposition of inference system are recursively estimated. The efficiency and applicability of the proposed methodology are demonstrated through experimental results of modeling of an industrial dryer.

Keywords Type-2 neural-fuzzy inference system · Interval type-2 fuzzy sets · Evolving systems · Fuzzy state-space identification

1 Introduction

Quite often, the knowledge that is used to design a fuzzy system can be uncertain. According to Mendel and John (2002), there are, at least, four resources of uncertainties to built a fuzzy system: (1) the meanings of words which are used in the antecedent/consequent proposition of rules can be uncertain, i.e., words have different meanings for different people; (2) when the knowledge is extracted from a group of experts who do not agree among themselves; (3) measurements that activate a type-1 fuzzy system may be noisy and therefore uncertain; (4) the data set used to tune parameters of fuzzy systems may contain uncertainty and noise. Therefore, type-1 fuzzy systems are not able to handle severe uncertainties because their membership functions are crisp. To deal with the uncertainties mentioned above, type-2 fuzzy systems were proposed, where their membership functions are themselves fuzzy (ZADEH 1975).

✉ Ginalber Luiz de Oliveira Serra
ginalber@ifma.edu.br

Anderson Pablo Freitas Evangelista
anderson.pablo.evangelista@gmail.com

1 Federal University of Maranhão, Av. dos Portugueses, 1966, Bacanga, São Luís, MA CEP 65080-805, Brazil

2 Federal Institute of Education, Science and Technology of Maranhao, Av. Getúlio Vargas, 04, Monte Castelo, São Luís, MA CEP 65030-005, Brazil

1.1 State-of-the-Art

The concept of a type-2 fuzzy set was introduced by Zadeh as an extension of the concept of ordinary fuzzy set (type-1 fuzzy set) (ZADEH 1975). However, the first works specifically dealing with type-2 fuzzy sets were published in the 1990s. In Karnik and Mendel (1998), the initial concepts about type-2 fuzzy system structure are presented such as operations among type-2 fuzzy sets.

A type-2 fuzzy system is a set of **IF-THEN** rules, where the antecedent and/or consequent propositions are composed of type-2 fuzzy sets. In type-2 fuzzy set (also called as general type-2 fuzzy set), each membership value is a fuzzy set in [0, 1]. Such sets can be used to represent uncertainty related to the shape of the membership function due to the use of uncertain information to generate it. This characteristic provides a degree of freedom for modeling problems with uncertainties in order to mitigate its effects. Thus, it is concluded that type-2 fuzzy system can handle better with uncertainties than its type-1 counterpart (Oscar Castillo

2007; Castillo 2011; Mendel 2014). The structure of a type-2 fuzzy system is similar to type-1 fuzzy system; however, the process to compute the output in type-2 fuzzy systems is performed by two steps: type reducing and defuzzification (Mendel and John 2002; Mendel et al. 2006; Oscar Castillo 2007). A disadvantage of type-2 fuzzy system is its application in real-time problem due to the processing time, once the general type-2 fuzzy sets increase the computational complexity for performing the type-reducing process (Mendel et al. 2006; Pratama et al. 2017).

A special case of type-2 fuzzy set, the interval type-2 fuzzy set, was introduced in Karnik et al. (1999) and Liang and Mendel (2000b), which is a simplification of the general type-2 fuzzy set. In an interval type-2 fuzzy set, the third dimension is always equal to one. Hence, the Footprint of Uncertainty (FoU) of an interval type-2 fuzzy set is defined by upper and lower membership functions. Compared to general type-2 fuzzy systems, the type-reduction operation for an interval type-2 fuzzy system is computationally less intensive. Based on interval type-2 fuzzy systems theory, many type-reduction algorithms have been proposed (Liang and Mendel 2000a; Wu and Mendel 2002; Nie and Tan 2008) and applied to several areas, such as control (Hagras 2004; Li et al. 2015), temporal series prediction (Lin et al. 2013; Lee et al. 2014) and system identification (Abiyev et al. 2013; Kayacan et al. 2015). In Li et al. (2016), a fault detection design scheme for interval type-2 Takagi–Sugeno fuzzy systems with sensor fault based on a novel fuzzy observer is proposed. The interval type-2 fuzzy model is used for modeling the nonlinear system, where the uncertain parameters are handled by the lower and upper membership degrees with weighting functions. The fault detection scheme is composed of the interval type-2 fuzzy model and interval type-2 observer, which derived condition to detect faults for a considered application. In Gao et al. (2019b), a distributed filtering scheme to deal with the fault detection problem of nonlinear stochastic systems with wireless sensor networks is presented. The nonlinear stochastic systems are represented by interval type-2 Takagi–Sugeno fuzzy system, and distributed interval type-2 fuzzy filters are designed for each sensor of the wireless sensor network.

The analysis of input–output experimental data from a dynamic system in order to find the proper rules and FoU for interval type-2 fuzzy modeling is difficult, time consuming and requires expert knowledge. Therefore, the problem of designing interval type-2 fuzzy inference systems and make good use of their properties is still open and an important research field. In this case, evolving methodologies have been proposed as one of the approaches to design interval type-2 fuzzy inference systems (Juang and Tsao 2008; Tung et al. 2013; Lin et al. 2015), as well as optimization/evolutionary methods (Kumar and Kumar 2017; Antonelli et al. 2017). In Das et al. (2015), an evolving interval type-2 neural-fuzzy inference system with sequential learning is proposed. A data-driven type-reduction procedure is used so that the computing of the output is fast and accurate. In Mohammadzadeh et al. (2016), a novel H_∞-based adaptive fuzzy control for the synchronization of fractional-order chaotic systems is presented. The consequent parameters are tuned based on adaptation laws that are derived from Lyapunov stability analysis. The antecedent part is estimated by clustering algorithm considering that the upper limit of type-2 membership degree and the rule database is optimized using a modified invasive weed optimization algorithm. In El-Nagar (2018), a structure of a recurrent interval type-2 Takagi–Sugeno–Kang fuzzy neural inference system for nonlinear time-varying dynamic systems identification is proposed. The antecedent and consequent parameters are updated based on the Lyapunov function to achieve inference system stability. In Pratama et al. (2017), an evolving recurrent fuzzy neural network structure, which presents two recurrent layers, is presented. In the antecedent proposition, multivariable Gaussian membership functions are used, whereas the consequent propositions are formed by the nonlinear wavelet functions. The evolving mechanism is governed by type-2 data quality method and the adaptation of the consequent parameters is performed by fuzzily weighted generalized recursive least squares, with application to time-series prediction in order to handle uncertainties inherent to the data.

As mentioned above in type-2 fuzzy model-based methodologies, it is observed that a set of multiple-inputs and single-output (MISO) type-2 fuzzy structures are commonly used to represent multivariable dynamic systems. Also, single variable membership functions are formulated in evolving context, where the clustering algorithm partition the input space in fuzzy regions, defining the membership function for each antecedent variable, and the relation of interval type-2 membership functions for each rule is computed by some t-norm. However, this approach may cause the loss of information due to low interaction between the antecedent variables (Kim et al. 1998; Lemos et al. 2011). With regard to the type reducer method, Nie and Tan (2008) and Wu (2013) approaches are generally adopted in literature, where they are more suitable to real-time application because they demand computationally lighter burden. In a comparative way, according to the proposed methodology, a single multiple-inputs and multiple-output (MIMO) interval type-2 neural-fuzzy structure is considered for multivariable dynamic systems identification. Regarding the membership functions, the interval type-2 multivariable Gaussian membership function with uncertain dispersion is adopted, whose choice is motivated by the following aspects: this function prevents information loss about antecedent variables interactions; induces more reliable input space partition because it is capable of covering arbitrary contours of data clouds; the type-reducing process based on extension of Begian–

Melek–Mendel and the Li–Yi–Zhao methods (Wu 2013) to multivariable fuzzy context; the evolving multivariable Gaussian fuzzy clustering algorithm as learning structure method in order to estimate the antecedent parameters and determine the number of rules in the interval type-2 state-space neural-fuzzy model; the extended Kalman filter for estimating the degrees of uncertainties associated with the shape of interval type-2 multivariable Gaussian membership functions; and a recursive state-space identification algorithm based on fuzzy Markov parameters for estimating the consequent parameters.

1.2 Contributions

The main contributions of the proposed methodology can be described as follows:

- Efficiency in tracking the dynamic uncertainty inherent to the experimental data set, adaptively. The evolving clustering algorithm estimates the center and shape of the clusters, defining the number of rules in type-2 neural-fuzzy inference system. The compatibility degree and the arousal index are important parameters to be computed for a new incoming information, considering all created clusters. Based on these metrics, conditions for variation in the number of rules, through creating or merging clusters, are used to allow the adaptation to a new structure of type-2 state-space neural-fuzzy inference system;
- Efficiency in tracking the nonlinearity inherent to the experimental data set, recursively. A fuzzy recursive least squares-based algorithm is used for estimating the state-space linear models in the consequence of the rules. The algorithm computes the fuzzy Markov parameters for all rules, and from fuzzy Markov parameters, the linear state-space parameters are computed, recursively. Once the i-th state-space model represents the linear behavior of the dynamic system to be identified in an operating point defined by each new sample, respectively, it can be inferred the efficiency of the obtained evolving type-2 neural-fuzzy inference system, through the evolving linear combination of state-space models in the consequence of each rule, in tracking the nonlinearity inherent to the experimental data set;
- Robustness to outliers inherent to experimental data. The evolving clustering algorithm computes the compatibility degree of new incoming information, i.e., its membership degree associated with each cluster indicates whether or not the information belongs to the knowledge base. It is also computed the arousal index, in the sense of evaluating, if the new incoming information represents persistently a new dynamic behavior not yet learned by the interval type-2 neural-fuzzy inference system. If the new incoming information is out of knowledge base and is persistent along the time, a new operating point is added to the inference system. If the new incoming information is out of knowledge base and is not persistent along the time, it is considered as noise, disturbance or outlier, and discarded by inference system. Besides that, an extended Kalman filter algorithm is used for estimating the interval type-2 fuzzy set, i.e., the parametrizations of upper and lower membership functions so that the effects of noise, disturbance and outliers are also attenuated in stages of compositional rule and type reducer.
- Low computational effort, memory allocation and execution time. For a new incoming information, the Markov parameters are computed and used for estimating of the linear state-space model in the consequent proposition of the interval type-2 neural-fuzzy inference system, via fuzzy version of recursive least squares method. Once the adopted methodology does not compute and allocate large Henkel matrices, and avoid the discontinuity problem induced by singular value decomposition in a recursive approach, which is more useful as compared to formulations found in several fuzzy state-space system identification methods, it can be inferred a faster processing for experimental data-based modeling, which is useful for real-time applications.

2 Interval Type-2 Neural-Fuzzy Takagi–Sugeno Inference System

The structure of the interval type-2 neural-fuzzy inference system, adopted in this paper, is shown in Fig. 1 (Abiyev et al. 2011; Tung et al. 2013; Lin et al. 2014; Wang and Kumbasar 2019). It is divided into five layers: input layer, fuzzification layer, consequent layer, type-2 compositional rule of inference and type reducer layer.

The adopted general form of the IF-THEN propositions is given by

$$\textbf{Rule}^i: \ \textbf{IF } \mathbf{z}_k \text{ is } \tilde{Z}^i$$
$$\textbf{THEN } \begin{cases} \mathbf{x}^i_{k+1} = \mathbf{A}^i\mathbf{x}^i_k + \mathbf{B}^i\mathbf{u}_k \\ \mathbf{y}^i_k = \mathbf{C}^i\mathbf{x}^i_k + \mathbf{D}^i\mathbf{u}_k \end{cases} \tag{1}$$

where $i = 1, 2, \ldots, c$ is the rule number, $\mathbf{z}_k = [z_{1,k}\ z_{2,k}\ \cdots\ z_{n_z,k}] \in \Re^{n_z}$ is the antecedent input variable, $\mathbf{A}^i \in \Re^{n\times n}$, $\mathbf{B}^i \in \Re^{n\times m}$, $\mathbf{C}^i \in \Re^{p\times n}$ and $\mathbf{D}^i \in \Re^{p\times m}$ are the state-space matrices of the local linear model from each rule, $\mathbf{x}^i_k = [x^i_{1,k}\ x^i_{2,k}\ \cdots\ x^i_{n,k}] \in \Re^n$ is the local state vector for i-th rule, $\mathbf{y}^i_k = [y^i_{1,k}\ y^i_{2,k}\ \cdots\ y^i_{p,k}] \in \Re^p$ is the local output vector for i-th rule, $\mathbf{u}_k = [u_{1,k}\ u_{2,k}\ \cdots\ u_{m,k}] \in \Re^m$ is input signal vector and $\tilde{Z}^i$ is the interval type-2 fuzzy set of the i-th rule. The details of each layer are discussed in the sequel.

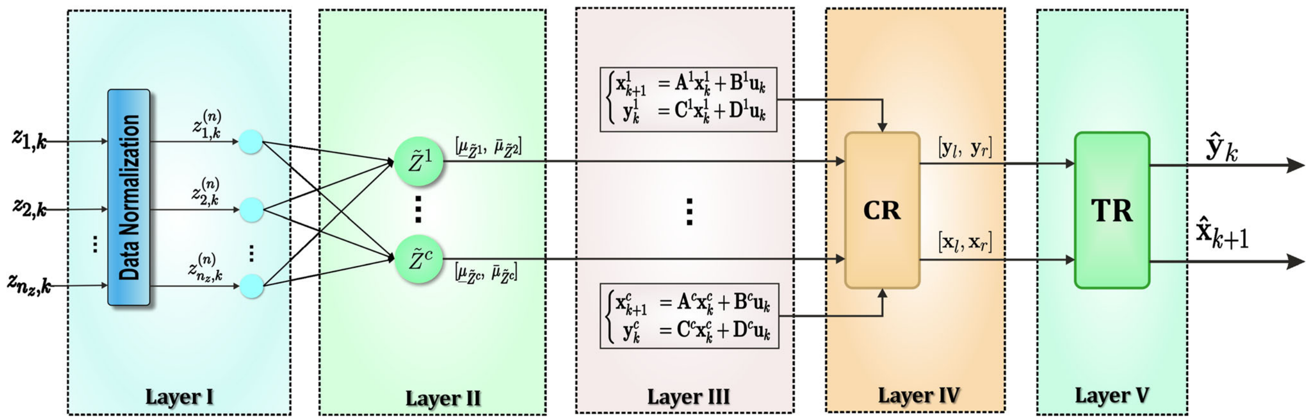


Fig. 1 Configuration of the proposed interval type-2 neural-fuzzy inference system

2.1 Layer I: Input Layer

The first layer of the type-2 neural-fuzzy inference system is the input layer. The nodes of this layer represent the physical variables related to dynamic system to be identified. Since these physical variables can operate in different ranges, it is useful to normalize the experimental data and adjust them in a unified range. Therefore, the max–min normalization operator is adopted, in the sense of overcoming the ill-conditioning and guaranteeing stable convergence of antecedent parameters, as well as allowing the constructing of new characteristics from input space. The max–min normalization operator is given by

$$z_{j,k}^{(n)} = \frac{z_{j,k} - z_j^{\min}}{z_j^{\max} - z_j^{\min}}, \text{ for } j = 1, 2, \ldots, n_z \tag{2}$$

where $z_{j,k}^{(n)} \in [0, 1]$, $z_j^{\min}$ and $z_j^{\max}$ are the maximum and minimum values of j-th feature, respectively, and $\mathbf{z}_{n,k}^{(n)} = [z_{1,k}^{(n)}\ z_{2,k}^{(n)}\ \cdots\ z_{n_z,k}^{(n)}]$ is the normalized input vector.

2.2 Layer II: Fuzzification Layer

The second layer of the interval type-2 neural-fuzzy inference system has the function of performing the fuzzification operation. Each node in this layer represents an interval type-2 fuzzy set. Thus, each interval type-2 fuzzy set is computed by an evolving multivariable Gaussian fuzzy clustering algorithm, which also adjusts the shape of membership function and defines new rules from input space in online manner as well. The fuzzification operation corresponds to the mapping $\Re^{n_z} \to \Re^1$, where the normalized experimental data in the input nodes vector $\mathbf{z}_{n,k}^{(n)}$ belongs to the interval type-2 fuzzy set $\tilde{Z}^i$ with an interval type-2 membership degree $\tilde{\mu}_{\tilde{Z}^i}^i(\mathbf{z}_k^{(n)}) = \left[\underline{\mu}_{\tilde{Z}^i}^i(\mathbf{z}_k^{(n)}),\ \bar{\mu}_{\tilde{Z}^i}^i(\mathbf{z}_k^{(n)})\right]$, computed as follows:

$$\tilde{\mu}_{\tilde{Z}^i}^i(\mathbf{z}_k^{(n)}) = \left[m_{\underline{\mu}} \exp\left(-\frac{1}{2} M_{\underline{\mathbf{\Sigma}}^i}(\mathbf{z}_k^{(n)}, \mathbf{z}^{i*})\right),\right. \\ \left. \exp\left(-\frac{1}{2} M_{\bar{\mathbf{\Sigma}}^i}(\mathbf{z}_k^{(n)}, \mathbf{z}^{i*})\right)\right] \tag{3}$$

where $m_{\underline{\mu}}$ corresponds to an adjustment parameter, chosen by an expert, and used to guarantee the uncertainty interval between lower and upper membership functions. The Mahalanobis distance functions $M_{\underline{\mathbf{\Sigma}}^i}(\bullet)$ and $M_{\bar{\mathbf{\Sigma}}^i}(\bullet)$ are defined by (Babuska 2012; Pratama et al. 2017):

$$M_{\underline{\mathbf{\Sigma}}^i}(\mathbf{z}_k^{(n)}, \mathbf{z}^{i*}) = (\mathbf{z}_k^{(n)} - \mathbf{z}^{i*})^T (\underline{\mathbf{6}}^i)^{-1} (\mathbf{z}_k^{(n)} - \mathbf{z}^{i*}) \tag{4}$$

$$M_{\bar{\mathbf{\Sigma}}^i}(\mathbf{z}_k^{(n)}, \mathbf{z}^{i*}) = (\mathbf{z}_k^{(n)} - \mathbf{z}^{i*})^T (\bar{\mathbf{\Sigma}}^i)^{-1} (\mathbf{z}_k^{(n)} - \mathbf{z}^{i*}) \tag{5}$$

where $\mathbf{z}^{i*} = [z_1^{i*}\ z_2^{i*}\ \cdots\ z_{n_z}^{i*}]$ is the center vector and $\underline{\mathbf{\Sigma}}^i$ and $\bar{\mathbf{\Sigma}}^i$ being the lower and upper dispersion matrices (symmetric and positive definite), respectively, which define the spread of the i-th lower and upper multivariate Gaussian membership function, in the sense of covering arbitrary contours of data clouds in input space. The same type of membership function used in this paper is very useful in other relevant approaches (Han et al. 2019; Gao et al. 2019a), but is important highlighting that other types of fuzzy membership functions can be used in the proposed approach, such as interval type-2 exponential membership function whose implementation would also imply in aspects as computational complexity and robustness (Wang et al. 2019).

2.3 Layer III: Consequent Layer

The third layer of the interval type-2 neural-fuzzy inference system has the objective of performing an arithmetic operation in function of linguistic variables of the antecedent, as a consequent proposition. The adopted arithmetic operation is based on linear state-space model structure, which allows the representation of the dynamic behavior of multivariate physical system. Thus, the i-th consequent proposition is given by

$$\mathbf{y}_k^i = \mathbf{C}^i\mathbf{x}_k^i + \mathbf{D}^i\mathbf{u}_k \tag{6}$$

$$\hat{\mathbf{x}}_{k+1}^i = \mathbf{A}^i\mathbf{x}_k^i + \mathbf{B}^i\mathbf{u}_k \tag{7}$$

2.4 Layer IV: Type-2 Compositional Rule of Inference

The fourth layer relates type-2 membership degrees from antecedent in layer II with the linear state-space models from consequent in layer III. Thus, the interval type-2 states $\left[\mathbf{x}_l, \mathbf{x}_r\right]$ and interval type-2 outputs $\left[\mathbf{y}_l, \mathbf{y}_r\right]$ of the interval type-2 neural-fuzzy inference system are computed through the composition of all rules of inference, where the indexes l and r represent the lower and upper bounds, respectively. The adopted procedure is given as follows:

$$\mathbf{y}_r = \max\left[\frac{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)})\mathbf{y}^i}{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)}) + \sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})}, \frac{\sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})\mathbf{y}^i}{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)}) + \sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})}\right] \tag{8}$$

$$\mathbf{y}_l = \min\left[\frac{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)})\mathbf{y}^i}{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)}) + \sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})}, \frac{\sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})\mathbf{y}^i}{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)}) + \sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})}\right] \tag{9}$$

$$\mathbf{x}_r = \max\left[\frac{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)})\mathbf{x}^i}{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)}) + \sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})}, \frac{\sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})\mathbf{x}^i}{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)}) + \sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})}\right] \tag{10}$$

$$\mathbf{x}_l = \min\left[\frac{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)})\mathbf{x}^i}{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)}) + \sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})}, \frac{\sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})\mathbf{x}^i}{\sum_{i=1}^{c}\bar{\mu}^i(\mathbf{z}_k^{(n)}) + \sum_{i=1}^{c}\underline{\mu}^i(\mathbf{z}_k^{(n)})}\right] \tag{11}$$

2.5 Layer V: Type Reducer Layer

The fifth layer performs the type reducing for computing type-1 states $\hat{\mathbf{x}}_{k+1}$ and type-1 outputs $\hat{\mathbf{y}}_k$ of the interval type-2 neural-fuzzy inference system. The procedure is defined by two arithmetic operations and is given by

$$\hat{\mathbf{x}}_{k+1} = \begin{bmatrix} \upsilon_1^{x,l} & 0 & \cdots & 0 \\ 0 & \upsilon_2^{x,l} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \upsilon_n^{x,l} \end{bmatrix}\mathbf{x}_k^l + \begin{bmatrix} \upsilon_1^{x,r} & 0 & \cdots & 0 \\ 0 & \upsilon_2^{x,r} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \upsilon_n^{x,r} \end{bmatrix}\mathbf{x}_k^r \tag{12}$$

$$\hat{\mathbf{y}}_{k} = \begin{bmatrix} \upsilon_1^{y,l} & 0 & \cdots & 0 \\ 0 & \upsilon_2^{y,l} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \upsilon_p^{y,l} \end{bmatrix}\mathbf{y}_k^l + \begin{bmatrix} \upsilon_1^{y,r} & 0 & \cdots & 0 \\ 0 & \upsilon_2^{y,r} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \upsilon_p^{y,r} \end{bmatrix}\mathbf{y}_k^r \tag{13}$$

where $\upsilon_j^{x,l}$, $\upsilon_j^{x,r}$, $\upsilon_j^{y,l}$ and $\upsilon_j^{y,r}$ are adaptive weights, which are updated by recursive least squares, as described in Algorithm 1. The initial value of covariance error matrices $\mathbf{P}_{d,0}^i$ and $\mathbf{P}_{s,0}^i$ satisfies $\mathbf{P}_{d,0}^i = 1/r_1\mathbf{I}$ and $\mathbf{P}_{s,0}^i = 1/r_2\mathbf{I}$, respectively, so that $r_1 > 0$, $r_2 > 0$, $\boldsymbol{\upsilon}_{d,k}^j = \left[\upsilon_j^{y,l} \;\; \upsilon_j^{y,r}\right]$ and $\boldsymbol{\upsilon}_{s,k}^j = \left[\upsilon_j^{x,l} \;\; \upsilon_j^{x,r}\right]$.

Algorithm 1: type reducer estimation - type_reducer_rls(•)

input : $\mathbf{y}_r, \mathbf{y}_l, \mathbf{x}_r, \mathbf{x}_l, , \boldsymbol{\upsilon}_{d,k}^j, \boldsymbol{\upsilon}_{s,k}^j, \mathbf{P}_o$
output: $\mathbf{t}_r, \mathbf{t}_l, \mathbf{P}_o$

for $j = 1$ **to** p **do**

$$\mathbf{P}_{d,k}^j = \frac{1}{\beta_1}\mathbf{P}_{d,k-1}^j\left[\mathbf{I} - \frac{[y_j^r \;\; y_j^l]^T[y_j^r \;\; y_j^l]\mathbf{P}_{d,k-1}^j}{\beta_1 + [y_j^r \;\; y_j^l]\mathbf{P}_{d,k-1}^j[y_j^r \;\; y_j^l]^T}\right] \tag{14}$$

$$\boldsymbol{\upsilon}_{d,k}^j = \boldsymbol{\upsilon}_{d,k}^j + \mathbf{P}_{d,k}^j\left[y_{j,k} - [y_j^r \;\; y_j^l]\boldsymbol{\upsilon}_{d,k}^j\right] \tag{15}$$

end

for $j = 1$ **to** n **do**

$$\mathbf{P}_{s,k}^j = \frac{1}{\beta_1}\mathbf{P}_{s,k-1}^j\left[\mathbf{I} - \frac{[x_j^l \;\; x_j^r]^T[x_j^l \;\; x_j^r]\mathbf{P}_{s,k-1}^j}{\beta_1 + [x_j^l \;\; x_j^r]\mathbf{P}_{s,k-1}^j[x_j^l \;\; x_j^r]^T}\right] \tag{16}$$

$$\boldsymbol{\upsilon}_{s,k}^j = \boldsymbol{\upsilon}_{s,k}^j + \mathbf{P}_{s,k}^j\left[x_{j,k} - [x_j^l \;\; x_j^r]\boldsymbol{\upsilon}_{s,k}^j\right] \tag{17}$$

end

2.6 Issues on Representation of Uncertainties

The interval type-2 neural-fuzzy structure is considered as a nonlinear function $f(\mathbf{z}_k) : \Re^{n_z} \rightarrow \Re^p$. In this sense, the data vector $\mathbf{z}_k$ is normalized and fuzzified for computing an interval membership grade $\tilde{\mu}^i_{\tilde{Z}^i}(\mathbf{z}_k^{(n)}) = \left[\underline{\mu}^i_{\tilde{Z}^i}(\mathbf{z}_k^{(n)}), \bar{\mu}^i_{\tilde{Z}^i}(\mathbf{z}_k^{(n)})\right]$. Thus, the interval output value $\tilde{\mathbf{y}} = [\mathbf{y}_l, \mathbf{y}_r]$ is obtained, which represents the uncertainties in the considered real dynamic system output, i.e., it is assumed that $\mathbf{y}_l \leq \mathbf{y}^* \leq \mathbf{y}_r$, where $\mathbf{y}^*$ is the real output, $\mathbf{y}_l$ is the inferior limit of the uncertainty and $\mathbf{y}_r$ is the superior limit of the uncertainty.

3 Learning Algorithm: Mathematical Formulation

In this section, a mathematical description of the proposed methodology is presented. The algorithms for evolving estimation of type-2 antecedent parameters and recursive estimation of consequent parameters are formulated.

3.1 Evolving Estimation of Type-2 Antecedent Parameters

The adopted clustering algorithm partitions the input space into interval type-2 multivariable Gaussian membership functions, characterized by a center $\mathbf{z}_k^{i*} \in \Re^{n_z}$ and a dispersion matrix (symmetric and positive definite) $\hat{\mathbf{\Sigma}}_k^i \in \Re^{n_z \times n_z}$, with $i = 1, 2, \ldots, c$ where c is the number of clusters.

The evolving estimation of type-2 antecedent parameters is based on participatory learning (PL) (Yager 1990; Silva et al. 2014; Filho and Serra 2018). The PL approach takes into account the impact of a new information to knowledge base of the interval type-2 neural-fuzzy inference system, i.e., the compatibility/ incompatibility of an incoming information implies for updating/keeping the current knowledge base, which makes the antecedent estimation more robust to effects of noise and outliers inherent to experimental data. Thus, two important variables are defined: the compatibility degree and the arousal index. The compatibility degree ρ^i of an incoming information $\mathbf{z}_k^{(n)}$ to the i-th cluster is computed as follows:

$$\rho_k^i = \exp\left(-\frac{1}{2} M_{\hat{\mathbf{\Sigma}}_k^i}(\mathbf{z}_k^{(n)}, \mathbf{z}_k^{i*})\right) \tag{18}$$

with

$$M_{\hat{\mathbf{\Sigma}}_k^i}(\mathbf{z}_k^{(n)}, \mathbf{z}_k^{i*}) = \left(\mathbf{z}_k^{(n)} - \mathbf{z}_k^{i*}\right)^T (\hat{\mathbf{\Sigma}}_k^i)^{-1} \left(\mathbf{z}_k^{(n)} - \mathbf{z}_k^{i*}\right) \tag{19}$$

where $\rho_k^i \in [0, 1]$, so that for $\rho_k^i = 1$ means that incoming information is completely compatible to i-th cluster.

The arousal index a_k^i supervises the creation of new rules based on compatibility degree of incoming information within a window of time. It is defined as the cumulative probability for occurring the violations t_v of a compatibility degree threshold, within a window w of time, and computed as follows:

$$a_k^i = P(T_{v,k}^i < t_v) = \sum P(T_{v,k}^i = t_v) \tag{20}$$

so that the probability function $P(T_{v,k}^i = t_v)$ is characterized by binomial distribution with probability of successful λ, computed as follows:

$$P(T_{v,k}^i = t_v^i) = \binom{w}{t_v^i} \lambda^{t_v^i} (1-\lambda)^{w-t_v^i}, \; t_v = 0, \ldots, w \tag{21}$$

where the number of violations $t_{v,k}^i$ of the compatibility degree threshold in i-th cluster is given by

$$t_{v,k}^i = \begin{cases} \sum_{j=0}^{w-1} o_{k-j}^i & k > w \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

with

$$o_k^i = \begin{cases} 1, & \text{for } \rho_k^i < T_\rho \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

where T_ρ is a compatibility degree threshold.

3.1.1 Cluster Creation Mechanism

Among several metrics used for creating new clusters (Angelov and Buswell 2002; Lughofer 2008; Maciel et al. 2013; Costa and Serra 2015; Pires and Serra 2018), in evolving fuzzy clustering algorithms, the compatibility degree and the arousal index are considered and defined as **Condition A**, which are given by

$$\textbf{IF } \rho_k^i < T_\rho \; \forall i \textbf{ AND } a_k^\chi > T_a \textbf{ THEN } \mathbf{z}^{c_k+1*} = \mathbf{z}_k^{(n)} \tag{24}$$

where $\chi = \arg\max \rho_k^i$, c_k is the number of clusters at discrete time k and T_a is the arousal threshold.

3.1.2 Cluster Adaptation Mechanism

In case of a new cluster has not been created, the inference system considers the adaptation of cluster with highest compatibility degree for incoming information, as follows:

$$G_k = \alpha(\rho_k^\chi)^{1-a_k^\chi} \tag{25}$$

$$\mathbf{z}_k^{\chi*} = \mathbf{z}_{k-1}^{\chi*} + G_k(\mathbf{z}_k^{(n)} - \mathbf{z}_{k-1}^{\chi*}) \tag{26}$$

$$\hat{\mathbf{\Sigma}}_k^\chi = (1 - G_k)(\hat{\mathbf{\Sigma}}_{k-1}^\chi - G_k(\mathbf{z}_k^{(n)} - \mathbf{z}_{k-1}^{\chi*})(\mathbf{z}_k^{(n)} - \mathbf{z}_{k-1}^{\chi*})^T) \tag{27}$$

where $\alpha \in [0, 1]$ is the learning rate and G_k is the gain of adaptation at discrete time k.

3.1.3 Merging Clusters Mechanism

The merging criteria are based on compatibility degree between the cluster χ, which was updated (or created), and an existing cluster j, at discrete time k, as follows:

$$\rho_k(\mathbf{z}_k^{\chi *}, \mathbf{z}_k^{j*}) = \exp\left(-\frac{1}{2} M_{\hat{\boldsymbol{\Sigma}}_k^j}(\mathbf{z}_k^{\chi *}, \mathbf{z}_k^{j*})\right) \quad (28)$$

and, so called as **Condition B** is given by

$$\textbf{IF } \rho_k(\mathbf{z}_k^{j*}, \mathbf{z}_k^{\chi *}) > T_\rho \textbf{ OR } \rho_k(\mathbf{z}_k^{\chi *}, \mathbf{z}_k^{j*}) > T_\rho$$
$$\textbf{THEN} \text{ merge } [\mathbf{z}^{\chi *}, \mathbf{z}^{j*}] \quad (29)$$

Algorithm 2 shows the flow of the participatory evolving fuzzy clustering algorithm.

3.1.4 Estimation of Type-2 Membership Functions

The footprint of uncertainty is represented by a type-2 dispersion matrix $\tilde{\boldsymbol{\Sigma}}_k^i = [\underline{\boldsymbol{\Sigma}}_k^i, \bar{\boldsymbol{\Sigma}}_k^i]$, which is defined as follows:

$$\tilde{\boldsymbol{\Sigma}}_k^i = \left[\hat{\boldsymbol{\Sigma}}_k^i - \boldsymbol{\Psi}_k^i, \hat{\boldsymbol{\Sigma}}_k^i + \boldsymbol{\Psi}_k^i\right] \quad (30)$$

where $\boldsymbol{\Psi}_k^i \in \Re^{n_z \times n_z}$ is a diagonal matrix and $\boldsymbol{\Psi}_k^i = diag([\psi_{1,k}^i \ \psi_{2,k}^i \ \cdots, \ \psi_{n_z,k}^i])$. The estimation of type-2 membership functions can be formulated as follows:

$$\boldsymbol{\psi}_{k+1}^i = f(\boldsymbol{\psi}) + \omega_k = \mathbf{I}\boldsymbol{\psi}_k^i + \omega_k \quad (31)$$
$$\check{\mathbf{y}}_k^i = h(\boldsymbol{\psi}_k^i) + \upsilon_k = \gamma_k^i(\mathbf{z}_{n,k})\mathbf{y}_k^i + \upsilon_k \quad (32)$$

where $\check{\mathbf{y}}^i = f(\boldsymbol{\psi}^i)$ is the output of type-2 inference system in i-th rule, $\boldsymbol{\psi}^i$ is the unknown uncertainty degree of dispersion matrix $\hat{\boldsymbol{\Sigma}}_k^i$, ω_k is a process noise and υ_k is a measurement noise, which are assumed with Gaussian distribution. Type-2 activation degree of i-th rule is given by

$$\gamma_k^i(\mathbf{z}_k^{(n)}) = \frac{(\bar{\mu}^i(\mathbf{z}_k^{(n)}) + \underline{\mu}^i(\mathbf{z}_k^{(n)}))}{\sum_{j=1}^c \bar{\mu}^j(\mathbf{z}_k^{(n)}) + \sum_{j=1}^c \underline{\mu}^j(\mathbf{z}_k^{(n)})} \quad (33)$$

The optimal estimation of type-2 membership functions minimizes the following loss function:

$$\mathbf{V}(\boldsymbol{\psi}, k) = \frac{1}{2}\sum_{j=1}^{k}\left(\check{\mathbf{y}}_j - \check{\mathbf{y}}_j^i\right)^T\left(\check{\mathbf{y}}_j^{(i)} - \check{\mathbf{y}}_j^i\right) \quad (34)$$

where

$$\check{\mathbf{y}}_j^{(i)} = \gamma_k^i(\mathbf{z}_j^{(n)})\mathbf{y}_j \quad (35)$$
$$\check{\mathbf{y}}_j^i = \gamma_k^i(\mathbf{z}_j^{(n)})\mathbf{y}_j^i \quad (36)$$

for $j = 1, 2, \ldots, c_k$.

Algorithm 2: Evolving Multi-variable Gaussian Fuzzy Clustering - eMG_clustering(•)

input : $\mathbf{z}^{(n)}, \mathbf{z}^*, \hat{\boldsymbol{\Sigma}}, w, \hat{\boldsymbol{\Sigma}}_0, \boldsymbol{\psi}, \boldsymbol{\psi}_0, c$
output: $\mathbf{z}^*, \hat{\boldsymbol{\Sigma}}, \boldsymbol{\psi}, \boldsymbol{\rho}, created, merged$

```
% compute ρ^i and a^i for all clusters;
for i = 1 to c do
    Compute M(z^(n), z^i*) - Eq. (19);
    Compute ρ^i - Eq. (18);
    if ρ^i < T_ρ then
        o^i = 1
    else
        o^i = 0
    end
    if k > w then
        nv^i = Σ_{l=0}^{w-1} o^i_{k-l};
        a^i = p(T^i_{k,v} < t^i_v);
    else
        a^i = 0
    end
end
χ = arg max_i ρ^i;
if Condition A (24) then
    % Create a new cluster
    c = c + 1;
    z^c* = z^(n);
    Σ̂^c = Σ̂_0;
    ψ^c = ψ_0;
    χ = c;
    created = true;
else
    Update cluster using Eqs. (25)-(27)
end
% Check for similar clusters
for i = 1 to c do
    if Condition B (29) then
        % Merge similar clusters [χ, i]
        z^χ* = (z^i* + z^χ*)/2;
        Σ̂^χ = Σ̂_0;
        c = c - 1;
        merged = [χ, i];
    end
end
```

Using an estimation algorithm based on extended Kalman filter, the uncertain degree $\boldsymbol{\psi}^i$ is computed as follows:

$$\mathbf{K}_k^i = \mathbf{P}_{f,k}^i(\mathbf{H}_k^i)^T\left[\mathbf{R}_k + \mathbf{H}_k^i\mathbf{P}_{f,k}^i(\mathbf{H}_k^i)^T\right]^{-1} \quad (37)$$
$$\boldsymbol{\psi}_{k+1}^i = \boldsymbol{\psi}_k^i + \mathbf{K}_k^i\left(\check{\mathbf{y}}_k^{(i)} - \check{\mathbf{y}}_k^i\right) \quad (38)$$
$$\mathbf{P}_{f,k+1}^i = \mathbf{P}_{f,k}^i - \mathbf{K}_k^i\mathbf{H}_k^i\mathbf{P}_{f,k}^i + \mathbf{Q}_k \quad (39)$$

where $\mathbf{K}_k^i$ is the Kalman gain, $\mathbf{P}_{f,k}^i \in \Re^{n_z \times n_z}$ ($\mathbf{P}_{f,0}^i = r_3\mathbf{I}$ and $r_3 > 0$) is the covariance matrix of residues, $\mathbf{R}_k \in \Re^{p \times p}$ is the measurement noise covariance matrix and $\mathbf{Q}_k \in \Re^{n_z \times n_z}$ is the process noise covariance matrix. The observation matrix $\mathbf{H}_k^i \in \Re^{p\times n}$, which is defined as Jacobian matrix, is given by

$$\mathbf{H}_k^i = \frac{\partial h(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}}\bigg|_{\boldsymbol{\psi}=\boldsymbol{\psi}_k} \tag{40}$$

According to Haykin (2001), the covariance matrix $\mathbf{R}_k$ can be given as a function of inverse learning rate, such that $\mathbf{R}_k = \zeta^{-1}\mathbf{W}_k^{-1}$. Thus, Eq. (37) can be formulated as follows:

$$\mathbf{K}_k^i = \mathbf{P}_{f,k}^i\mathbf{H}_k^i\left[\zeta^{-1}\mathbf{W}_k^{-1} + \mathbf{H}_k^i\mathbf{P}_{f,k}^i(\mathbf{H}_k^i)^T\right]^{-1} \tag{41}$$

where $\zeta \in [0, 1]$ is learning rate and $\mathbf{W}_k$ is the weight matrix. The implementation of parameter-based Kalman filter learning procedure is shown in Algorithm 3.

Algorithm 3: EKF learning - EKF(•)

input : $\boldsymbol{\psi}_k, \mathbf{z}_k^{(n)}, \mathbf{x}_k, \mathbf{y}_k, \mathbf{u}_k, \mathbf{z}^*, \hat{\boldsymbol{\Sigma}}, \mathbf{P}_{f,k}, \zeta, \mathbf{C}, \mathbf{D}, c$
output: $\boldsymbol{\psi}_{k+1}, \mathbf{P}_{f,k+1}$

for $i = 1$ **to** c **do**
  Compute $[\underline{\boldsymbol{\Sigma}}_k^i, \bar{\boldsymbol{\Sigma}}_k^i]$, - Eq. (30);
  Compute $\tilde{\mu}_{\tilde{Z}^i}^i = [\underline{\mu}_{\tilde{Z}^i}^i, \bar{\mu}_{\tilde{Z}^i}^i]$ - Eqs. (3)-(5);
  compute γ_i - Eq. (33);
  compute $\check{\mathbf{y}}_k^{(i)}$ and $\check{\mathbf{y}}_k^i$ - Eqs. (35)-(36);
  compute $\mathbf{H}^i$ - Eq. (40);
end

% $\boldsymbol{\psi}^i$ and $\mathbf{P}_f$ update;
for $i = 1$ **to** c **do**
  Compute $\mathbf{K}_k^i$ - Eq. (41);
  Compute $\boldsymbol{\psi}_{k+1}^i$ - Eq. (38);
  Compute $\mathbf{P}_{f,k+1}^i$ - Eq. (39);
end

3.2 Recursive Estimation of Type-2 Consequent Parameters

According to Eq. (1), in Sect. 2, the local state-space linear model in the consequence associated with i-th rule of the interval type-2 neural-fuzzy inference system is given by

$$\begin{aligned}\hat{\mathbf{x}}_{k+1}^i &= \mathbf{A}^i\hat{\mathbf{x}}_k^i + \mathbf{B}^i\mathbf{u}_k\\ \hat{\mathbf{y}}_k^i &= \mathbf{C}^i\hat{\mathbf{x}}_k^i + \mathbf{D}^i\mathbf{u}_k\end{aligned} \tag{42}$$

Considering the inclusion of a state observer to Eq. (42), it has

$$\begin{aligned}\hat{\mathbf{x}}_{k+1}^i &= \mathbf{A}^i\hat{\mathbf{x}}_k^i + \mathbf{B}^i\mathbf{u}_k + \mathbf{L}^i\mathbf{e}_k\\ \hat{\mathbf{y}}_k^i &= \mathbf{C}^i\hat{\mathbf{x}}_k^i + \mathbf{D}^i\mathbf{u}_k + \mathbf{e}_k^i\end{aligned} \tag{43}$$

where $\mathbf{e}_k^i = \mathbf{y}_k^i - \hat{\mathbf{y}}_k^i$ and $\mathbf{L}^i$ denote the white innovation sequence and local Kalman gain matrix for i-th rule, respectively.

Assuming that controllability and observability properties are satisfied (Houtzager et al. 2012; Ni et al. 2018a), Eq. (43) can be formulated as follows:

$$\hat{\mathbf{x}}_k^i = (\mathbf{A}^i)^{q_p}\hat{\mathbf{x}}_{k-q_p}^i + \mathcal{C}_B^i\bar{\mathbf{u}}_{k-q_p} + \mathcal{C}_L^i\bar{\mathbf{e}}_{k-q_p}^i \tag{44}$$

$$\bar{\mathbf{y}}_{k+q_f}^i = \mathcal{O}^i\hat{\mathbf{x}}_k^i + \mathcal{G}^i\bar{\mathbf{u}}_{k+q_f} + \mathcal{H}^i\bar{\mathbf{e}}_{k+q_f}^i \tag{45}$$

where

$$\bar{\mathbf{u}}_{k-q_p} = \begin{bmatrix}\mathbf{u}_{k-q_p}\\ \mathbf{u}_{k-q_p+1}\\ \vdots\\ \mathbf{u}_{k-1}\end{bmatrix}, \quad \bar{\mathbf{u}}_{k+q_f} = \begin{bmatrix}\mathbf{u}_k\\ \mathbf{u}_{k+1}\\ \vdots\\ \mathbf{u}_{k+q_f-1}\end{bmatrix}$$

$$\bar{\mathbf{y}}_{k-q_p}^i = \begin{bmatrix}\mathbf{y}_{k-q_p}^i\\ \mathbf{y}_{k-q_p-1}^i\\ \vdots\\ \mathbf{y}_{k-1}^i\end{bmatrix}, \quad \bar{\mathbf{y}}_{k+q_f}^i = \begin{bmatrix}\mathbf{y}_k^i\\ \mathbf{y}_{k+1}^i\\ \vdots\\ \mathbf{y}_{k+q_f-1}^i\end{bmatrix}$$

$$\bar{\mathbf{e}}_{k-q_p}^i = \begin{bmatrix}\mathbf{e}_{k-q_p}^i\\ \mathbf{e}_{k-q_p+1}^i\\ \vdots\\ \mathbf{e}_{k-1}^i\end{bmatrix}, \quad \bar{\mathbf{e}}_{k+q_f}^i = \begin{bmatrix}\mathbf{e}_k^i\\ \mathbf{e}_{k+1}^i\\ \vdots\\ \mathbf{e}_{k+q_f-1}^i\end{bmatrix}$$

$$\mathcal{O}^i = \begin{bmatrix}\mathbf{C}^i\\ \mathbf{C}^i\mathbf{A}^i\\ \vdots\\ \mathbf{C}^i(\mathbf{A}^i)^{q_f-1}\end{bmatrix}$$

$$\mathcal{C}_B^i = \left[(\mathbf{A}^i)^{q_p-1}\mathbf{B}^i \;\; (\mathbf{A}^i)^{q_p-2}\mathbf{B}^i \;\cdots\; \mathbf{A}^i\mathbf{B}^i \;\; \mathbf{B}^i\right]$$

$$\mathcal{C}_L^i = \left[(\mathbf{A}^i)^{q_p-1}\mathbf{L}^i \;\; (\mathbf{A}^i)^{q_p-2}\,\mathbf{L}^i \;\cdots\; \mathbf{A}^i\mathbf{L}^i \;\; \mathbf{L}^i\right]$$

$$\mathcal{G}^i = \begin{bmatrix}\mathbf{D}^i & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{C}^i\mathbf{B}^i & \mathbf{D}^i & \ddots & \vdots\\ \vdots & \vdots & \ddots & \mathbf{0}\\ \mathbf{C}^i(\mathbf{A}^i)^{q_f-2}\mathbf{B} & \mathbf{C}^i(\mathbf{A}^i)^{q_f-3}\mathbf{B} & \cdots & \mathbf{D}^i\end{bmatrix}$$

$$\mathcal{H}^i = \begin{bmatrix}\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{C}^i\mathbf{L}^i & \mathbf{I} & \ddots & \vdots\\ \vdots & \vdots & \ddots & \mathbf{0}\\ \mathbf{C}^i(\mathbf{A}^i)^{q_f-2}\mathbf{L} & \mathbf{C}^i(\mathbf{A}^i)^{q_f-3}\mathbf{L} & \cdots & \mathbf{I}\end{bmatrix}$$

where $\mathcal{O}^i \in \Re^{pq_f\times n}$ is the observability matrix, $\mathcal{C}_B^i \in \Re^{n\times q_p m}$ and $\mathcal{C}_L^i \in \Re^{n\times pq_p}$ are extend controllability matrix, $\mathcal{G}^i \in \Re^{q_f p\times q_f m}$ and $\mathcal{H}^i \in \Re^{q_f m\times q_f m}$ are the impulse matrices with a lower block triangular structure and q_p and q_f are the past time size and the future time size, respectively.

3.2.1 Computation of Fuzzy Markov Parameters

According to Chiuso and Picci (2005), a vector autoregressive with exogenous inputs (VARX) predictor used as approximator of the dynamic system in Eq. (43) can provide a consistent estimation of Markov parameters. Thus, let the VARX predictor is given by

$$\hat{\mathbf{y}}_k^i = \sum_{j=0}^{q} \boldsymbol{\Xi}_{k-j}^{i,(u)} \mathbf{u}_{k-j} + \sum_{j=1}^{q} \boldsymbol{\Xi}_{k-j}^{i,(y)} \hat{\mathbf{y}}_{k-j}^i \tag{46}$$

where $\hat{\mathbf{y}}_k^i$ is the predicted output at sample k using the inputs at samples $k, k-1, \ldots, k-p$ and using the outputs at samples $k-1, \ldots, k-p$. Therefore, assuming that the dynamic system in Eq. (43) is asymptotically stable, the transition matrix $\mathbf{A}^i$ is nilpotent, i.e., its eigenvalues are inside the unity circle and, for some integer $q_p \in \mathbb{N}$, the expression $(\mathbf{A}^i)^{q_p} \approx \mathbf{0}$ is satisfied. Thus, the contribution of the state vector $\mathbf{x}_{k-q_p}^i$ in Eq. (44) can be made arbitrarily small for large values of q_p, leading in a systemic simplification, which affects the matrices $\boldsymbol{\Xi}^{i,(u)}$ and $\boldsymbol{\Xi}^{i,(y)}$ of the VARX in Eq. (46) related to the Markov parameters of the system, where

$$\boldsymbol{\Xi}_{k-j}^{i,(u)} = \begin{cases} \mathbf{D}^i & \text{if } j = 0 \\ \mathbf{C}^i (\mathbf{A}^i)^{j-1} \mathbf{B}^i & \text{if } j > 0 \end{cases} \tag{47}$$

$$\boldsymbol{\Xi}_{k-j}^{i,(y)} = \mathbf{C}^i (\mathbf{A}^i)^{j-1} \mathbf{L}^i \tag{48}$$

From Eqs. (47) and (48), the Markov parameters matrix (Jer-Nan Juang 2011; Wu et al. 2015) can be defined as follows:

$$\boldsymbol{\Xi}^i = \left[\boldsymbol{\Xi}_{k-p}^{i,(u)}, \cdots, \boldsymbol{\Xi}_k^{i,(u)}, \boldsymbol{\Xi}_{k-p}^{i,(y)} \cdots \boldsymbol{\Xi}_{k-1}^{i,(y)} \right] \tag{49}$$

and Eq. (46) can be formulated as follows:

$$\mathbf{y}_k^i = \boldsymbol{\Xi}^i \boldsymbol{\delta}_k^i + \mathbf{e}_k \tag{50}$$

where $\boldsymbol{\delta}_k^i = \left[\bar{\mathbf{u}}_{k-q_p}^T \;\; \mathbf{u}_k^T \;\; (\bar{\mathbf{y}}_{k-q_p}^i)^T \right]^T$. Considering $k > q_p$, the batch representation of Eq. (50) is given by

$$\mathbf{Y}_k^i = \boldsymbol{\Xi}^i \boldsymbol{\Delta}_k^i \tag{51}$$

where

$$\mathbf{Y}_k^i = \left[\mathbf{y}_{q_p+1}^i \;\; \mathbf{y}_{q+2}^i \;\cdots\; \mathbf{y}_k^i \right], \tag{52}$$

$$\boldsymbol{\Delta}_k^i = \left[\boldsymbol{\delta}_{q_p+1}^i \;\; \boldsymbol{\delta}_{q_p+2}^i \;\cdots\; \boldsymbol{\delta}_k^i \right] \tag{53}$$

Thus, the output of the interval type-2 neural-fuzzy inference system is given by

$$\mathbf{Y}_k = \sum_{i=1}^{c} \boldsymbol{\Xi}^i \boldsymbol{\Delta}_k^i \boldsymbol{\Gamma}_k^i \tag{54}$$

where $\boldsymbol{\Gamma}_k^i = \mathrm{diag}\left(\left[\gamma_{q_p+1}^i, \gamma_{q_p+2}^i, \ldots, \gamma_k^i\right]\right)$, so that $\gamma_{q_p+1}^i$ is computed by Eq. (33).

The local approach-based batch solution for Eq. (54), in the sense of ensuring interpretability of interval type-2 neural-fuzzy inference system, is given by least square (LS) solution, as follows:

$$\boldsymbol{\Xi}_k^i = \mathbf{Y}_k \boldsymbol{\Gamma}_k^i (\boldsymbol{\Delta}_k^i)^T \left[\boldsymbol{\Delta}_k^i \boldsymbol{\Gamma}_k^i (\boldsymbol{\Delta}_k^i)^T \right]^{-1} \tag{55}$$

where the relation $q_p \geq q_f \geq n/p$ must be satisfied so that Eq. (55) can be solved by the least mean square method (Ni et al. 2018a; Houtzager et al. 2012), $\mathbf{Y}_k = \left[\mathbf{y}_{q_p+1} \, \mathbf{y}_{q_p+2} \cdots \mathbf{y}_k \right]$ is the output matrix, and the Markov parameters $\boldsymbol{\Xi}_k^i$ minimizes the loss function:

$$\mathbf{V}(\boldsymbol{\Xi}^i, k) = \frac{1}{2} \sum_{j=1}^{k} \gamma_j^i \left(\mathbf{y}_j - \mathbf{y}_j^i \right)^T \left(\mathbf{y}_j - \mathbf{y}_j^i \right) \tag{56}$$

Expanding Eq. (55) for discrete time of $k+1$, it has

$$\boldsymbol{\Xi}_{k+1}^i = \mathbf{Y}_{k+1} \boldsymbol{\Gamma}_{k+1}^i (\boldsymbol{\Delta}_{k+1}^i)^T \left[\boldsymbol{\Delta}_{k+1}^i \boldsymbol{\Gamma}_{k+1}^i (\boldsymbol{\Delta}_{k+1}^i)^T \right]^{-1} \tag{57}$$

where $\mathbf{Y}_{k+1} = \left[\mathbf{Y}_k \; \mathbf{y}_{k+1} \right]$ and $\boldsymbol{\Delta}_{k+1}^i = \left[\boldsymbol{\Delta}_k^i \; \boldsymbol{\delta}_{k+1}^i \right]$. So, from Eq. (57), the covariance matrix is defined as follows:

$$\begin{aligned} \mathbf{P}_{k+1}^i &= \left[\boldsymbol{\Delta}_{k+1}^i \boldsymbol{\Gamma}_{k+1}^i (\boldsymbol{\Delta}_{k+1}^i)^T \right]^{-1} \\ &= \left[\sum_{j=1}^{k+1} \gamma_j^i \boldsymbol{\delta}_j^i (\boldsymbol{\delta}_j^i)^T \right]^{-1} \\ &= \left[(\mathbf{P}_k^i)^{-1} + \gamma_{k+1}^i \boldsymbol{\delta}_{k+1}^i (\boldsymbol{\delta}_{k+1}^i)^T \right]^{-1} \end{aligned} \tag{58}$$

Using the matrix inversion lemma in Eq. (58), it has

$$\begin{aligned} \mathbf{P}_{k+1}^i &= \mathbf{P}_k^i - \frac{\gamma_{k+1}^i \mathbf{P}_k^i \boldsymbol{\delta}_{k+1}^i (\boldsymbol{\delta}_{k+1}^i)^T \mathbf{P}_k^i}{1 + \gamma_{k+1}^i (\boldsymbol{\delta}_{k+1}^i)^T \mathbf{P}_k^i \boldsymbol{\delta}_{k+1}^i} \\ &\boxed{= \mathbf{P}_k^i \left[\mathbf{I} - \frac{\gamma_{k+1}^i \boldsymbol{\delta}_{k+1}^i (\boldsymbol{\delta}_{k+1}^i)^T \mathbf{P}_k^i}{1 + \gamma_k^i (\boldsymbol{\delta}_{k+1}^i)^T \mathbf{P}_k^i \boldsymbol{\delta}_{k+1}^i} \right]} \end{aligned} \tag{59}$$

Substituting (58) in (57), results in

$$\begin{aligned} \boldsymbol{\Xi}_{k+1}^i &= \left[\mathbf{Y}_k \;\; \mathbf{y}_{k+1} \right] \begin{bmatrix} \boldsymbol{\Gamma}_k^i & \mathbf{0} \\ \mathbf{0} & \gamma_{k+1}^i \end{bmatrix} \left[\boldsymbol{\Delta}_k^i \;\; \boldsymbol{\delta}_{k+1}^i \right]^T \mathbf{P}_{k+1}^i \\ &= \left[\mathbf{Y}_k \boldsymbol{\Gamma}_k^i (\boldsymbol{\Delta}_k^i)^T + \gamma_{k+1}^i \mathbf{y}_{k+1} (\boldsymbol{\delta}_{k+1}^i)^T \right] \mathbf{P}_{k+1}^i \\ &\boxed{= \boldsymbol{\Xi}_k^i + \gamma_{k+1}^i \left[\mathbf{y}_{k+1} - \boldsymbol{\Xi}_k^i \boldsymbol{\delta}_{k+1}^i \right] (\boldsymbol{\delta}_{k+1}^i)^T \mathbf{P}_{k+1}^i} \end{aligned} \tag{60}$$

where Eq. (60) is the recursive estimation of fuzzy Markov parameters.

For time-varying dynamics, a pragmatic approach allows the following criterion:

$$\mathbf{V}(\boldsymbol{\Xi}^i, k) = \frac{1}{2}\sum_{j=1}^{k} \beta_2^{k-j} \left[\gamma_j^i \left(\mathbf{y}_j - \mathbf{y}_j^i\right)^T \left(\mathbf{y}_j - \mathbf{y}_j^i\right)\right] \quad (61)$$

The parameter $\boldsymbol{\Xi}^i$, which minimizes (61), can be estimated, recursively, as follows:

$$\mathbf{P}_{k+1}^i = \frac{1}{\beta_2}\mathbf{P}_k^i \left[\mathbf{I} - \frac{\gamma_{k+1}^i \boldsymbol{\delta}_{k+1}^i (\boldsymbol{\delta}_{k+1}^i)^T \mathbf{P}_k^i}{\beta_2 + \gamma_k^i (\boldsymbol{\delta}_{k+1}^i)^T \mathbf{P}_k^i \boldsymbol{\delta}_{k+1}^i}\right] \quad (62)$$

$$\boldsymbol{\Xi}_{k+1}^i = \boldsymbol{\Xi}_k^i + \gamma_{k+1}^i \left[\mathbf{y}_{k+1} - \boldsymbol{\Xi}_k^i \boldsymbol{\delta}_{k+1}^i\right] (\boldsymbol{\delta}_{k+1}^i)^T \mathbf{P}_{k+1}^i \quad (63)$$

where the initial values of $\mathbf{P}_0^i$ must satisfy $\mathbf{P}_0^i = r_4\mathbf{I}$ and $r_4 > 0$.

3.2.2 Computation of State-Space Matrices

From Markov parameters $\boldsymbol{\Xi}_k^i$, the matrices $\boldsymbol{\Lambda}_k^i$ and $\boldsymbol{\Upsilon}_k^i$ are defined as follows:

$$\boldsymbol{\Lambda}_k^i = \begin{bmatrix} \boldsymbol{\Xi}_{k-q_p}^{i,(u)} & \boldsymbol{\Xi}_{k-q_p+1}^{i,(u)} & \cdots & \boldsymbol{\Xi}_{k-q_p+q_f-1}^{i,(u)} & \cdots & \boldsymbol{\Xi}_{k-1}^{i,(u)} \\ \mathbf{0} & \boldsymbol{\Xi}_{k-q_p}^{i,(u)} & \cdots & \boldsymbol{\Xi}_{k-q_p+q_f-2}^{i,(u)} & \cdots & \boldsymbol{\Xi}_{k-2}^{i,(u)} \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Xi}_{k-q_p}^{i,(u)} & \cdots & \boldsymbol{\Xi}_{k-q_f}^{i,(u)} \end{bmatrix} \quad (64)$$

$$\boldsymbol{\Upsilon}_k^i = \begin{bmatrix} \boldsymbol{\Xi}_{k-q_p}^{i,(y)} & \boldsymbol{\Xi}_{k-q_p+1}^{i,(y)} & \cdots & \boldsymbol{\Xi}_{k-q_p+q_f-1}^{i,(y)} & \cdots & \boldsymbol{\Xi}_{k-1}^{i,(y)} \\ \mathbf{0} & \boldsymbol{\Xi}_{k-q_p}^{i,(y)} & \cdots & \boldsymbol{\Xi}_{k-q_p+q_f-2}^{i,(y)} & \cdots & \boldsymbol{\Xi}_{k-2}^{i,(y)} \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Xi}_{k-q_p}^{i,(y)} & \cdots & \boldsymbol{\Xi}_{k-q_f}^{i,(y)} \end{bmatrix} \quad (65)$$

The state vector $\mathbf{x}_k^i$ in Eq. (44) can be computed in function of $\bar{\mathbf{u}}_{k-q_p}$, $\bar{\mathbf{y}}_{k-q_p}$, $\boldsymbol{\Lambda}_k^i$ and $\boldsymbol{\Upsilon}_k^i$ as follows (Ni et al. 2018a, b):

$$\mathbf{x}_k^i = \mathbf{S}\left[\boldsymbol{\Lambda}_k^i \bar{\mathbf{u}}_{k-q_p} + \boldsymbol{\Upsilon}_k^i \bar{\mathbf{y}}_{k-q_p}\right] \quad (66)$$

where $\mathbf{S} \in \Re^{n\times(mf-n)}$ is a positive defined matrix, which can be selected by user, in the sense of ensuring recursive computation of state vector $\mathbf{x}_k^i$ be convergent (Ni et al. 2018a). From Eq. (66), the fuzzy state vector $\tilde{\mathbf{x}}_k$ is estimated as follows:

$$\tilde{\mathbf{x}}_k = \sum_{i=1}^{c_k} \gamma_k^i \mathbf{x}_k^i \quad (67)$$

For recursive estimation of fuzzy matrices $\left[\mathbf{A}_k^i, \mathbf{B}_k^i, \mathbf{C}_k^i, \mathbf{D}_k^i\right]$, Eq. (42) can be formulated as follows:

$$\mathbf{x}_{k+1}^i = \left[\mathbf{A}_k^i \;\; \mathbf{B}_k^i\right]\begin{bmatrix}\mathbf{x}_k^i \\ \mathbf{u}_k\end{bmatrix} = \boldsymbol{\Theta}_k^{i,(x)} \boldsymbol{\nu}_k \quad (68)$$

$$\mathbf{y}_k^i = \left[\mathbf{C}_k^i \;\; \mathbf{D}_k^i\right]\begin{bmatrix}\mathbf{x}_k^i \\ \mathbf{u}_k\end{bmatrix} + \mathbf{e}_k = \boldsymbol{\Theta}_k^{i,(y)} \boldsymbol{\nu}_k + \mathbf{e}_k \quad (69)$$

Let the loss functions $\mathbf{V}(\boldsymbol{\Theta}_k^{i,(x)}, k)$ and $\mathbf{V}(\boldsymbol{\Theta}_k^{i,(y)}, k)$ for the case of time-varying dynamics are given by

$$\mathbf{V}(\boldsymbol{\Theta}_k^{i,(x)}, k) = \frac{1}{2}\sum_{j=1}^{k} \beta_3^{k-j} \left[\gamma_j^i \left(\mathbf{x}_j - \mathbf{x}_j^i\right)^T \left(\mathbf{x}_j - \mathbf{x}_j^i\right)\right] \quad (70)$$

$$\mathbf{V}(\boldsymbol{\Theta}_k^{i,(y)}, k) = \frac{1}{2}\sum_{j=1}^{k} \beta_4^{k-j} \left[\gamma_j^i \left(\mathbf{y}_j - \mathbf{y}_j^i\right)^T \left(\mathbf{y}_j - \mathbf{y}_j^i\right)\right] \quad (71)$$

where $\mathbf{x}_j^i = \boldsymbol{\Theta}_j^{i,(x)} \boldsymbol{\nu}_j$. The recursive least-square solution for $\boldsymbol{\Theta}_k^{i,(x)}$ and $\boldsymbol{\Theta}_k^{i,(y)}$ is given by

$$\mathbf{P}_k^{i,(x)} = \frac{1}{\beta_3}\mathbf{P}_{k-1}^{i,(x)} \left[\mathbf{I} - \frac{\gamma_k^i \boldsymbol{\nu}_{k-1}^i (\boldsymbol{\nu}_{k-1}^i)^T \mathbf{P}_{k-1}^{i,(x)}}{\beta_3 + \gamma_k^i (\boldsymbol{\nu}_{k-1}^i)^T \mathbf{P}_{k-1}^{i,(x)} \boldsymbol{\nu}_{k-1}^i}\right] \quad (72)$$

$$\boldsymbol{\Theta}_k^{i,(x)} = \boldsymbol{\Theta}_{k-1}^{i,(x)} + \gamma_k^i \left[\mathbf{x}_k^i - \boldsymbol{\Theta}_{k-1}^{i,(x)} \boldsymbol{\nu}_{k-1}^i\right] \mathbf{P}_k^{i,(x)} \quad (73)$$

$$\mathbf{P}_k^{i,(y)} = \frac{1}{\beta_4}\mathbf{P}_{k-1}^{i,(y)} \left[\mathbf{I} - \frac{\gamma_k^i \boldsymbol{\nu}_{k-1}^i (\boldsymbol{\nu}_{k-1}^i)^T \mathbf{P}_{k-1}^{i,(y)}}{\beta_4 + \gamma_k^i (\boldsymbol{\nu}_{k-1}^i)^T \mathbf{P}_{k-1}^{i,(x)} \boldsymbol{\nu}_{k-1}^i}\right] \quad (74)$$

$$\boldsymbol{\Theta}_k^{i,(y)} = \boldsymbol{\Theta}_{k-1}^{i,(y)} + \gamma_k^i \left[\mathbf{y}_{k-1} - \boldsymbol{\Theta}_{k-1}^{i,(y)} \boldsymbol{\nu}_{k-1}^i\right] \mathbf{P}_k^{i,(y)} \quad (75)$$

where the initial values of $\mathbf{P}_0^{i,(x)}$ are usually selected as $\mathbf{P}_0^{i,(x)} = r_5\mathbf{I}$, such that $r_5 > 0$; $\mathbf{P}_0^{i,(y)} = r_6\mathbf{I}$, with $r_6 > 0$.

The steps for recursive estimation of type-2 consequent parameters are shown through Algorithm 4, in the sequel.

3.3 Issues on Initialization: Parameters and Structure

Initially, once there is no knowledge about the physical system, the number of rules is zero. The first incoming information becomes the center of first membership function, i.e., $\mathbf{z}^{1*} = \mathbf{z}_1$ and the initial type-2 dispersion matrix, and is given by $\tilde{\boldsymbol{\Sigma}}_0 = [\hat{\boldsymbol{\Sigma}}_0 - \boldsymbol{\Psi}_0, \hat{\boldsymbol{\Sigma}}_0 + \boldsymbol{\Psi}_0]$, where $\hat{\boldsymbol{\Sigma}}_0 = \sigma_1 \mathbf{I}^{n_z}$ with $\sigma_1 \in [10^{-2}, 0.5]$ and $\boldsymbol{\Psi}_0 = \sigma_2\ \mathbf{1}^{1\times n_z}$ with $\sigma_2 \in [10^{-10}, 10^{-4}]$.

Algorithm 4: Recursive Fuzzy State-Space Identification Algorithm: RFSIA(●)

input : $\mathbf{x}_k, \mathbf{y}_k, \Xi_k, \mathbf{P}_k, \boldsymbol{\Theta}_k^{(x)}, \mathbf{P}_k^{(x)}, \mathbf{P}_k^{(y)}, \boldsymbol{\Theta}_k^{(y)}, \boldsymbol{\gamma}_k, c$
output: $\Xi_{k+1}, \mathbf{P}_{k+1}, \boldsymbol{\Theta}_k^{i,(x)}, \mathbf{P}_k^{i,(x)}, \mathbf{P}_{k+1}^{i,(y)}, \boldsymbol{\Theta}_{k+1}^{i,(y)}$

$\mathbf{S} = \left[\mathbf{I}_{n\times n} \quad \mathbf{0}_{n\times(mf-n)}\right]$;
$\boldsymbol{\gamma}_k = \left[\gamma_k^1 \ \gamma_k^2 \ \cdots \ \gamma_k^c\right]$;

%***Step 1***: *Update the fuzzy Markov parameter* Ξ^i
for $i = 1$ **to** c **do**
 Compute $\mathbf{P}_{k+1}^i$ - Eq. (62);
 Compute Ξ_{k+1}^i - Eq. (63);
end

%***Step 2***: *construct the matrices* $\boldsymbol{\Lambda}_k^i$ *and* $\boldsymbol{\Upsilon}_k^i$
for $i = 1$ **to** c **do**
 construct $\boldsymbol{\Lambda}_k^i$ - Eq. (64);
 construct $\boldsymbol{\Upsilon}_k^i$ - Eq. (65);
end

%***Step 3***: *Estimate the local state vector* $\mathbf{x}_k^i$
for $i = 1$ **to** c **do**
 estimate $\mathbf{x}_k^i$ - Eq. (66);
end

%***Step 4***: *Update the matrices* $\boldsymbol{\Theta}^{i,(x)}$ *and* $\boldsymbol{\Theta}^{i,(y)}$
for $i = 1$ **to** c **do**
 Compute $\mathbf{P}_k^{i,(x)}$ - Eq. (72);
 Compute $\boldsymbol{\Theta}_k^{i,(x)}$ - Eq. (73);
 Compute $\mathbf{P}_{k+1}^{i,(y)}$ - Eq. (74);
 Compute $\boldsymbol{\Theta}_{k+1}^{i,(y)}$ - Eq. (75);
end

The first local model is given by

$$\mathbf{A}_0 = \kappa \mathbf{I}^n, \quad \mathbf{B}_0 = \kappa \mathbf{I}^{n\times m}$$
$$\mathbf{C}_0 = \kappa \mathbf{I}^{p\times n}, \quad \mathbf{D}_0 = \kappa \mathbf{I}^{p\times m}$$

where $\kappa \in [0, 1]$ is an initialization parameter used to set the state-space matrices, and

$$\boldsymbol{\Theta}_0^{i,(x)} = \left[\mathbf{A}_0 \ \ \mathbf{B}_0\right]$$
$$\boldsymbol{\Theta}_0^{i,(y)} = \left[\mathbf{C}_0 \ \ \mathbf{D}_0\right]$$

In the evolving clustering algorithm, four parameters are used: the compatibility threshold $T_\rho \in \ ]0, 1[$, learning rate $\alpha \in [10^{-5}, 10^{-1}]$, window size w and probability of successful λ. The T_ρ governs the ability to generate rules of the clustering algorithm, where as $T_\rho \approx 0$ there is practically no generation of cluster; and as $T_\rho \approx 1$ many clusters will be created; therefore, it is suggested to choose a value between 0.2 and 0.7. The arousal threshold is given by $T_a = 1 - \lambda$, where λ depends of w. The ranges for values of λ, given w, according to Lemos et al. (2011), are defined as follows:

$$\lambda \geq \begin{cases} 0.01 & \text{If} \quad w \geq 100 \\ 0.05 & \text{If} \ \ 20 \leq w < 100 \\ 0.1 & \text{If} \ \ 10 \leq w < 20 \end{cases} \tag{76}$$

In the consequent estimation, the forgetting factor β_j for $j = 1, 2, 3, 4$ is a user-defined parameter which can be given by $1 > \beta_j > 0.95$ so as to reduce the influence of old data on the estimation of the parameters (Juang 1994; Jer-Nan Juang 2011). Finally, the complete learning procedure of the proposed interval type-2 neural-fuzzy inference system is shown in Algorithm 5.

3.4 Issues on Convergence of the Tracking Errors

So to be clear how to ensure the convergence of the tracking errors, let an interval type-2 neural-fuzzy inference system with c rules. The i-th submodel output can be estimated from its Markov parameters for $k > q_p$, according to Eq. (50). The matrix $\boldsymbol{\Gamma}_k^i$ can be factored by

$$\begin{aligned} \boldsymbol{\Gamma}_k^i &= \breve{\boldsymbol{\Gamma}}_k^i \breve{\boldsymbol{\Gamma}}_k^i \\ &= \begin{bmatrix} \sqrt{\gamma_{q_p+1}^i} & 0 & 0 \cdots & 0 \\ 0 & \sqrt{\gamma_{q_p+2}^i} & 0 \cdots & 0 \\ \vdots & \vdots & \vdots \ddots & \vdots \\ 0 & 0 & 0 \cdots & \sqrt{\gamma_k^i} \end{bmatrix} \\ &\quad \times \begin{bmatrix} \sqrt{\gamma_{q_p+1}^i} & 0 & 0 \cdots & 0 \\ 0 & \sqrt{\gamma_{q_p+2}^i} & 0 \cdots & 0 \\ \vdots & & \vdots \ddots & \vdots \\ 0 & 0 & 0 \cdots & \sqrt{\gamma_k^i} \end{bmatrix} \end{aligned} \tag{77}$$

and, the least square (LS) solution Ξ_{LS}^i using Eq. (55), is formulated by

$$\Xi_{LS}^i = \breve{\mathbf{Y}}_k^i (\breve{\boldsymbol{\Delta}}_k^i)^T \left[\breve{\boldsymbol{\Delta}}_k^i (\breve{\boldsymbol{\Delta}}_k^i)^T\right]^{-1} \tag{78}$$

where

$$\breve{\mathbf{Y}}_k^i = \mathbf{Y}_k^i \breve{\boldsymbol{\Gamma}}_k^i \tag{79}$$
$$\breve{\boldsymbol{\Delta}}_k^i = \boldsymbol{\Delta}_k^i \breve{\boldsymbol{\Gamma}}_k^i \tag{80}$$

Assumption 1 The multivariable nonlinear system can be represented by an interval type-2 state-space neural-fuzzy structure (Fig. 1) with c_R rules, such that the rule form is given by

Algorithm 5: Main Algorithm

Initialize $r_1, r_2, r_3, r_4, , r_5, r_6, \beta_1, \beta_2, \beta_3, , \alpha, \zeta, \lambda, T_\rho, w, q_p, q_f$ $\sigma_1, \sigma_2, \mathbf{S}, \mathbf{W}$
$k = 1$;
Read sample $\mathbf{z}_k^{(n)}$;
% Initialize the first cluster
$c = 1$;
$\mathbf{z}^{c*} = \mathbf{z}_k^{(n)}, \hat{\mathbf{\Sigma}}^c = \hat{\mathbf{\Sigma}}_0, \boldsymbol{\psi}^c = \boldsymbol{\psi}_0, \mathbf{\Xi}_k^c = \mathbf{0}$;
$\mathbf{\Theta}_k^{(x)} = \mathbf{\Theta}_0, \mathbf{\Theta}_k^{(y)} = \mathbf{\Theta}_0^{(y)}$
repeat
  $k = k + 1$; Read sample $\mathbf{z}_k^{(n)}$;
  %prediction procedure;
  for $i = 1$ **to** c **do**
    compute $\mathbf{x}_k^i$ - Eq. (66);
    compute $\mathbf{y}_k^i$ - Eq. (6);
    compute $\tilde{\mu}^i_{\tilde{Z}^i} = [\underline{\mu}^i_{\tilde{Z}^i}, \bar{\mu}^i_{\tilde{Z}^i}]$ - Eqs. (3)-(5);
  end
  compute the model output $\hat{\mathbf{y}}_k$ - Eqs. (8), (9), (13);
  %Type Reducer Estimation - (Algorithm 1);
  $[\boldsymbol{\upsilon}^j_{d,k}, \boldsymbol{\upsilon}^j_{s,k}, \mathbf{P}_d, \mathbf{P}_s] =$
  type_reducer_rls($\mathbf{y}^r, \mathbf{y}^l, \boldsymbol{\upsilon}^j_{d,k}, \boldsymbol{\upsilon}^j_{s,k}, \mathbf{P}_d, \mathbf{P}_s$);
  %evolving Clustering procedure - (Algorithm 2)
  $[\mathbf{z}^*, \hat{\mathbf{\Sigma}}, \boldsymbol{\psi}, \boldsymbol{\rho}_k, c_k, created, merged] =$
  eMG_clustering($\mathbf{z}^{(n)}, \mathbf{z}^*, \hat{\mathbf{\Sigma}}, w, \hat{\mathbf{\Sigma}}_0, \boldsymbol{\psi}, \boldsymbol{\psi}_0, c_k$) ;
  if $created$ **then**
    $\mathbf{\Xi}_k^c = \frac{\sum_{i=1}^{c-1} \rho_k^i \mathbf{\Xi}_k^i}{\sum_{i=1}^{c-1} \rho_k^i}$;
    $\mathbf{\Theta}_k^{c,(x)} = \frac{\sum_{i=1}^{c-1} \rho_k^i \mathbf{\Theta}_k^{i,(x)}}{\sum_{i=1}^{c-1} \rho_k^i}$;
    $\mathbf{\Theta}_k^{c,(y)} = \frac{\sum_{i=1}^{c-1} \rho_k^i \mathbf{\Theta}_k^{i,(y)}}{\sum_{i=1}^{c-1} \rho_k^i}$;
    $\mathbf{P}^i_{f,k} = r_3\mathbf{I}, \ \mathbf{P}^i_k = r_4\mathbf{I}$,
    $\mathbf{P}_k^{i,(x)} = r_5\mathbf{I}, \ \mathbf{P}_k^{i,(y)} = r_6\mathbf{I}$,
  end
  if $merged \neq \varnothing$ **then**
    $\mathbf{\Xi}_k^{idx} = \frac{\rho_k^{idx}\mathbf{\Xi}_k^{idx} + \rho_k^i \mathbf{\Xi}_k^i}{\rho_k^{idx} + \rho_k^i}$;
    $\mathbf{P}^{idx}_{f,k} = r_3\mathbf{I}, \ \mathbf{P}^{idx}_k = r_4\mathbf{I}$,;
    $\mathbf{P}_k^{idx,(x)} = r_5\mathbf{I}, \ \mathbf{P}_k^{idx,(y)} = r_6\mathbf{I}$,;
  end
  %Consequent estimation - (Algorithm 4);
  $[\mathbf{\Xi}_{k+1}, \mathbf{P}_{k+1}, \mathbf{\Theta}_k^{i,(x)}, \mathbf{P}_k^{i,(x)}, \mathbf{P}_{k+1}^{i,(y)}, \mathbf{\Theta}_{k+1}^{i,(y)}] =$
  RFSIA($\mathbf{x}_k, \mathbf{y}_k, \mathbf{\Xi}_k, \mathbf{P}_k, \mathbf{\Theta}_k^{(x)}, \mathbf{P}_k^{(x)}, \mathbf{P}_k^{(y)}, \mathbf{\Theta}_k^{(y)}, \boldsymbol{\gamma}_k, c$) ;
  Extract the local state-space matrices $\left[\mathbf{A}^i, \mathbf{B}^i, \mathbf{C}^i, \mathbf{D}^i\right]$ from $\mathbf{\Theta}^{i,(x)}$ and $\mathbf{\Theta}^{i,(y)}$;
  %Type-2 parameter estimation - (Algorithm 3);
  $[\boldsymbol{\psi}_{k+1}, \mathbf{P}_{f,k+1}] =$
  EKF($\boldsymbol{\psi}_k, \mathbf{z}_k^{(n)}, \mathbf{x}_k, \mathbf{y}_k, \mathbf{u}_k, \mathbf{z}^*, \hat{\mathbf{\Sigma}}, \mathbf{P}_{f,k}, \zeta, \mathbf{C}, \mathbf{D}, c$);
until $\mathbf{z}_{k+1} = \varnothing$;

$$\textbf{Rule}^{i=1,2,\ldots,c_R}: \ \textbf{IF } \mathbf{z}_k \text{ is } \tilde{Z}^i$$

$$\textbf{THEN} \begin{cases} \mathbf{x}_{k+1}^i = \mathbf{A}_R^i \mathbf{x}_k^i + \mathbf{B}_R^i \mathbf{u}_k \\ \mathbf{y}_k^i = \mathbf{C}_R^i \mathbf{x}_k^i + \mathbf{D}_R^i \mathbf{u}_k \end{cases} \tag{81}$$

where the $(\bullet)_R$ (subscript R) represents the real parameters of the interval type-2 state-space neural-fuzzy model.

Let the batch computation of the output is given by

$$\mathbf{Y}_k^i = \mathbf{\Xi}_R^i \mathbf{\Delta}_k^i + \mathbf{E}_k \tag{82}$$

where $\mathbf{Y}_k^i = \left[\mathbf{y}^i_{q_p+1} \ \mathbf{y}^i_{q+2} \ \cdots \ \mathbf{y}^i_k\right]$ and $\mathbf{E}_k$ is an error matrix. Through Assumption 1 and considering $c = c_R$, substituting Eq. (82) in Eq. (79), Eq. (78) is reformulated as follows:

$$\mathbf{\Xi}_{LS}^i = \mathbf{\Xi}_R^i + \breve{\mathbf{E}}_k^i (\breve{\mathbf{\Delta}}_k^i)^T \left[\breve{\mathbf{\Delta}}_k^i (\breve{\mathbf{\Delta}}_k^i)^T\right]^{-1} \tag{83}$$

where $\breve{\mathbf{E}}_k^i = \mathbf{E}_k \breve{\mathbf{\Gamma}}_k^i$. Let the following covariance matrix

$$\mathbf{P}_{\breve{\delta}}^i = \left[\breve{\mathbf{\Delta}}_k^i (\breve{\mathbf{\Delta}}_k^i)^T\right] = \left[\frac{1}{k - q_p} \sum_{j=q_p}^{k} \breve{\boldsymbol{\delta}}_j \breve{\boldsymbol{\delta}}_j^T\right] \tag{84}$$

where $\breve{\boldsymbol{\delta}}_j = \sqrt{\gamma_j^i}\boldsymbol{\delta}_j = \sqrt{\gamma_j^i}\left[\bar{\mathbf{u}}_{k-q_p}^T \ \ \mathbf{u}_k^T \ \ (\bar{\mathbf{y}}^i_{k-q_p})^T\right]^T$. Thus, Eq. (83) is formulated as follows:

$$\mathbf{\Xi}_{LS}^i = \mathbf{\Xi}_R^i + \frac{1}{k - q_p}\left[\sum_{j=q_p}^{k} \breve{\mathbf{e}}_j^i \breve{\boldsymbol{\delta}}^T\right](\mathbf{P}_{\breve{\delta}}^i)^{-1} \tag{85}$$

In order to analyze the behavior of the LS solution for $k \to \infty$, it is appropriate to assume that $\mathbf{e}_k$ is stationary stochastic processes and $\mathbf{u}_k$ is a quasi-stationary process, so that

$$\mathbf{P}_u = \frac{1}{k}\sum_{j=1}^{k} \mathbf{u}_k \mathbf{u}_{k-\tau}^T \tag{86}$$

converges (with probability 1) when $k \to \infty$ (see Ljung 1999), i. e.,

$$\lim_{k\to\infty} \mathbf{P}_u = \overline{\mathbf{P}}_u \tag{87}$$

Thus, analyzing $\breve{\mathbf{P}}_{\delta}^i$ and $\left[\frac{1}{k-q_p}\sum_{j=q_p}^{k} \breve{\mathbf{e}}_j^i \breve{\boldsymbol{\delta}}^T\right]$ when $k \to \infty$, it has

$$\lim_{k\to\infty} \mathbf{P}_{\breve{\delta}}^i \longrightarrow \overline{\mathbf{P}}_{\breve{\delta}}^i \tag{88}$$

$$\frac{1}{k - q_p}\sum_{j=q_p}^{k} \breve{\mathbf{e}}_j^i \breve{\boldsymbol{\delta}}^T = \mathbf{P}_{\breve{\delta}\breve{e}}^i \tag{89}$$

$$\lim_{k\to\infty} \mathbf{P}_{\breve{\delta}\breve{e}}^i \longrightarrow \overline{\mathbf{P}}_{\breve{\delta}\breve{e}}^i \tag{90}$$

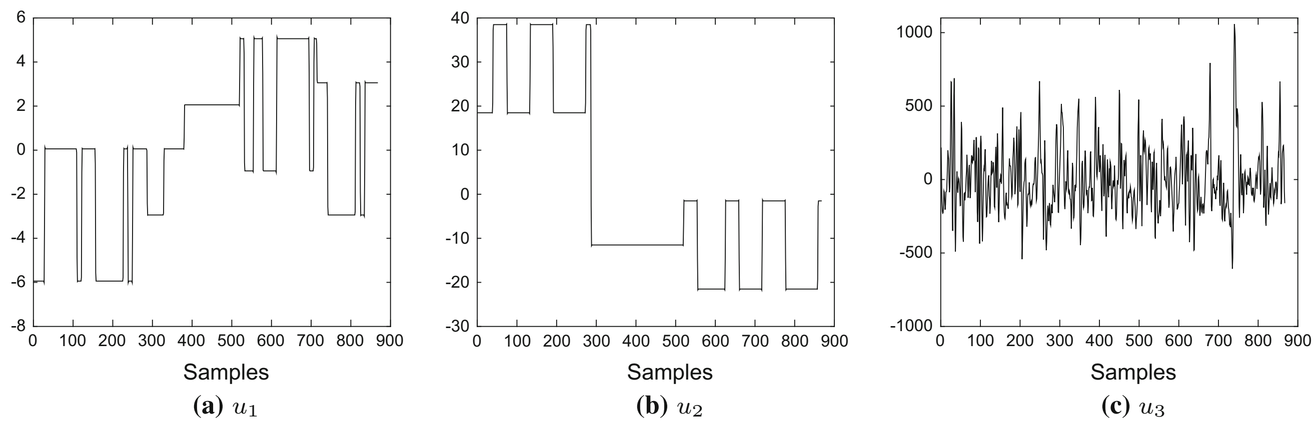


Fig. 2 Industrial dryer data set: inputs

In other words, $\mathbf{P}^i_{\breve{\delta}}$ converges (with probability 1) to $\overline{\mathbf{P}}^i_{\breve{\delta}}$ and similarly, $\mathbf{P}^i_{\breve{\delta}\breve{\mathbf{e}}}$ converges (with probability 1) to $\overline{\mathbf{P}}^i_{\breve{\delta}\breve{\mathbf{e}}}$ when $k \to \infty$ (see Ljung 1999). Thus, for $k \to \infty$, Eq. (85) is given by

$$\boldsymbol{\Xi}^i_{LS} = \boldsymbol{\Xi}^i_R + \overline{\mathbf{P}}^i_{\breve{\delta}\breve{\mathbf{e}}}(\overline{\mathbf{P}}^i_{\breve{\delta}})^{-1} \tag{91}$$

Hence, for the parametric estimation of the i-th submodel to be **convergent** and **consistent**, i. e., $\boldsymbol{\Xi}^i_{LS} \to \boldsymbol{\Xi}^i_R$ when $k \to \infty$, the following necessary conditions must be satisfied (Ljung 1999):

(C.1): $\overline{\mathbf{P}}^i_{\breve{\delta}}$ is nonsingular. It is satisfied if $\mathbf{u}_k$ and $\mathbf{e}_k$ must be independent and $\mathbf{u}_k$ must be persistently exciting, with k sufficiently large;

(C.2): $\mathbf{e}_k$ is described by a white noise sequence. In this case $\overline{\mathbf{P}}^i_{\breve{\delta}\mathbf{e}} = \mathbf{0}$, because $\mathbf{e}_k$ is not dependent of data until time instant $k-1$.

(C.3): In consequence of (C.1), the activation degree must be greater than zero, i. e., $\gamma^i_j > 0 \ \forall i, \ \forall j$.

3.5 Issues on Stability of Estimation

Let recursive estimation of fuzzy Markov parameters is given by Eq. (60):

$$\boldsymbol{\Xi}^i_{k+1} = \boldsymbol{\Xi}^i_k + \left[\breve{\mathbf{y}}^i_{k+1} - \boldsymbol{\Xi}^i_k \breve{\boldsymbol{\delta}}^i_{k+1}\right] (\breve{\boldsymbol{\delta}}^i_{k+1})^T \mathbf{P}^i_{k+1} \tag{92}$$

Considering $(\breve{\boldsymbol{\delta}}^i_{k+1})^T \mathbf{P}^i_{k+1} = \breve{\mathbf{G}}^i_k$, Eq. (92) is reformulated as follows:

$$\boldsymbol{\Xi}^i_{k+1} = \boldsymbol{\Xi}^i_k + \left[\breve{\mathbf{y}}_{k+1} - \boldsymbol{\Xi}^i_k \breve{\boldsymbol{\delta}}^i_{k+1}\right] \breve{\mathbf{G}}^i_{k+1} \tag{93}$$

Reformulating Eq. (93), it has

$$\boldsymbol{\Xi}^i_{k+1} = \boldsymbol{\Xi}^i_k (\mathbf{I} - (\breve{\boldsymbol{\delta}}^i_{k+1}) \breve{\mathbf{G}}^i_{k+1}) + \breve{\mathbf{y}}_{k+1} \breve{\mathbf{G}}^i_{k+1} \tag{94}$$

If a parametric error $\mathbf{e}_{\boldsymbol{\Xi}^i_{k_0}}$ is introduced at time k_0, from Eq. (94), the error propagation from instant k_0 to $k+1$ is given by

$$\Delta_{\boldsymbol{\Xi}^i_k} = \mathbf{e}_{\boldsymbol{\Xi}^i_{k_0}} \prod_{j=k_0}^{k+1} (\mathbf{I} - (\breve{\boldsymbol{\delta}}^i_{j+1}) \breve{\mathbf{G}}^i_{j+1}) \tag{95}$$

Based on Assumption 1 and $c = c_R$, the error propagation proprieties are dependent on

$$\boldsymbol{\Phi}(k_0, k+1) = \prod_{j=k_0}^{k+1} (\mathbf{I} - (\breve{\boldsymbol{\delta}}^i_{j+1}) \breve{\mathbf{G}}^i_{j+1}) \tag{96}$$

Based on Eq. (58), the matrix $\breve{\mathbf{P}}^i_k$, can be expressed as follows:

$$\breve{\mathbf{P}}^i_k = \left[\sum_{j=q_p}^{k} \beta_2^{k-j} \breve{\boldsymbol{\delta}}_j \breve{\boldsymbol{\delta}}_j^T\right]^{-1} \tag{97}$$

Thus, Eq. (96) is expressed as follows:

$$\boldsymbol{\Phi}(k_0, k+1) = \beta_2^{k-k_0+1} \breve{\mathbf{P}}^i_{k_0} (\breve{\mathbf{P}}^i_{k+1})^{-1} \tag{98}$$

Since $(\breve{\mathbf{P}}^i_k)^{-1}$ remains uniformly bounded and the forgetting factor $\beta_2 < 1$, it can be stated that the effect of a single error in recursive estimation given by Eq. (93) decays exponentially (Ljung and Ljung 1985). Being $\beta_2 = 1$, so $\breve{\mathbf{P}}^i_{k_0} (\breve{\mathbf{P}}^i_k)^{-1} \leq \mathbf{I}$, which means **stability** estimation.

4 Experimental Results

In this section, the efficiency and applicability of the proposed methodology, according to the theoretical formulation presented in Sect. 3, are discussed by the identification of an industrial dryer (Chou and Maciejowski 1997; Santos and

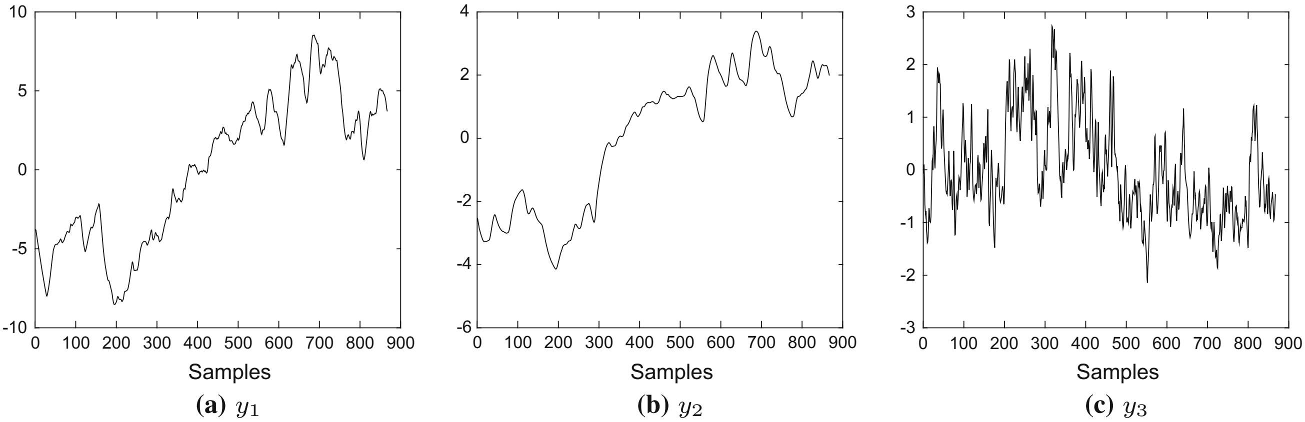


Fig. 3 Industrial dryer data set: outputs

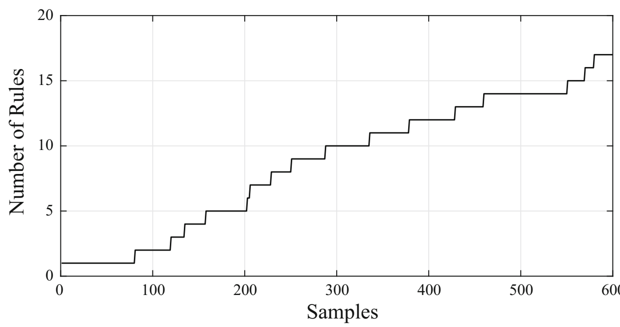


Fig. 4 Variation in the number of rules during the evolving identification of the industrial dryer

Barreto 2018), which has three inputs (fuel flow rate u_1, hot gas exhaust u_2 and raw material u_3) and three outputs (dry bulb temperature y_1, wet bulb temperature y_2 and moisture content of the raw material when it leaves the dryer y_3). The industrial dryer receives the raw material continuously, and the moisture in it is removed by passing hot gas. The efficiency of this process depends on both the temperature and the rate of flow of the hot gas.

The experimental data set consists of 867 inputs and outputs samples, as shown in Figs. 2 and 3, which are available at the database for the identification of systems (DaISy)[1] repository. According to this repository, the physical units are omitted due to some pre-processing applied to data set. In order to represent the dynamic behavior of the industrial dryer, it used an interval type-2 state-space TS neural-fuzzy inference system, with the following generalized rule base:

$$\textbf{Rule } i: \ \textbf{IF } \mathbf{z}_k = [\mathbf{u}_{k-1}\ \mathbf{y}_{k-1}] \text{ is } \tilde{Z}^i$$
$$\textbf{THEN} \begin{cases} \mathbf{x}^i_{k+1} = \mathbf{A}^i\mathbf{x}^i_k + \mathbf{B}^i\mathbf{u}_k \\ \mathbf{y}^i_{k+3} = \mathbf{C}^i\mathbf{x}^i_k + \mathbf{D}^i\mathbf{u}_k \end{cases} \tag{99}$$

where the center of $\tilde{Z}^i$ is given by $\mathbf{z}^{i*}_k = [\mathbf{u}^{i*}_{k-1}\ \mathbf{y}^{i*}_{k-1}]$.

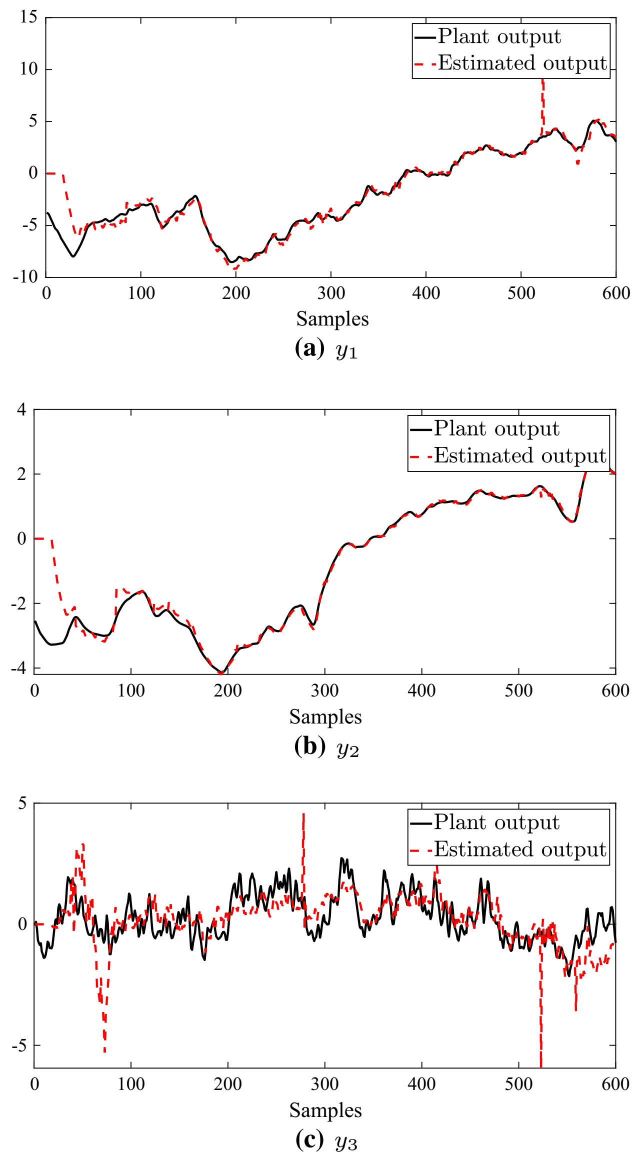


Fig. 5 Performance of identification in training process for industrial dryer

[1] See https://homes.esat.kuleuven.be/~tokka/daisydata.html.

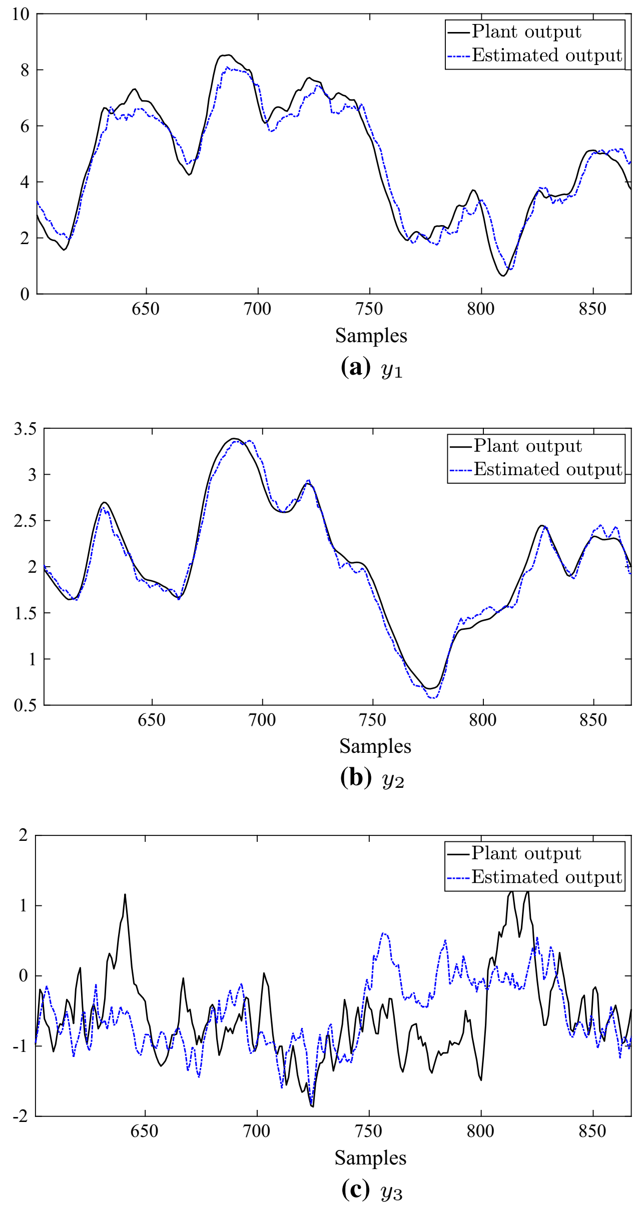


Fig. 6 Performance of interval type-2 neural-fuzzy model for industrial dryer validation data set

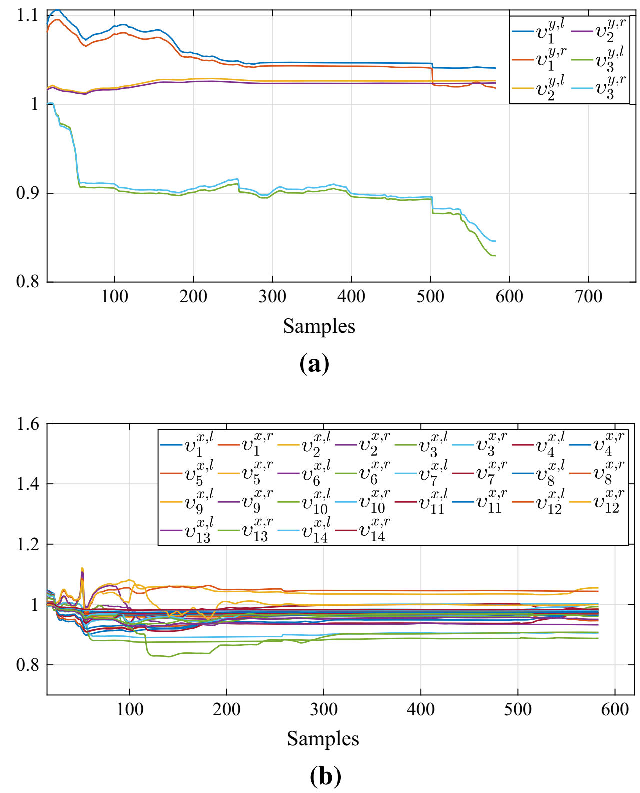


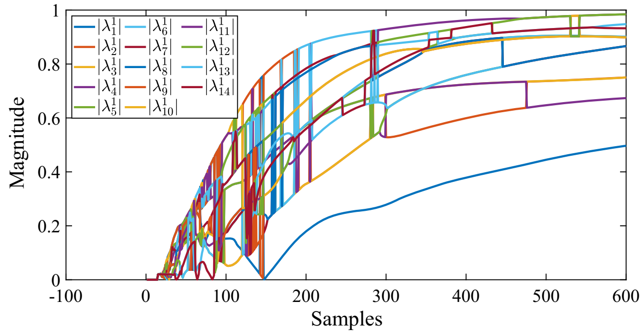
Fig. 7 The recursive estimation of the parameters $\upsilon_j^{x,l}$, $\upsilon_j^{x,r}|^{j=1,\ldots,15}$, $\upsilon_j^{y,l}$ and $\upsilon_j^{y,r}$ of type reducer layer: **a** the parameters $\upsilon_j^{y,l}$ and $\upsilon_j^{y,r}$ are used to compute the crisp outputs and **b** the parameters $\upsilon_j^{x,l}$ and $\upsilon_j^{x,r}$ are used to compute the crisp states

The parametrization used for implementing the algorithm of the proposed methodology is defined as follows: $m_{\underline{\mu}} = 0.9$ (maximum value of lower membership function); $\lambda = 0.07$ (probability of success); $\alpha = 1 \times 10^{-5}$ (clustering learning rate); $T_\rho = 0.77$ (compatibility threshold); $w = 80$ (clustering window size); $\sigma_1 = 0.3$ (initial dispersion constant); $\sigma_2 = 10^{-3}$ (initial degree of uncertainty); $\eta = 10^{-2}$ (EKF learning rate); $\beta_1 = 1$ (forgetting factor—type reducer parameters); $\beta_2 = 1 - 10^{-4}$ (forgetting factor—Markov parameter estimation); $\beta_3 = 1 - 10^{-6}$ (forgetting factor—$[\mathbf{A}^i\ \mathbf{B}^i]$ estimation); $\beta_4 = 1 - 10^{-6}$ (forgetting factor—$[\mathbf{C}^i\ \mathbf{D}^i]$ estimation); $r_1 = 10^{-3}$ (constant initialization of covariance matrix—state type reduction); $r_2 = 10^{-3}$ (constant initialization of covariance matrix—output type reduction); $r_3 = 20$ (constant initialization of

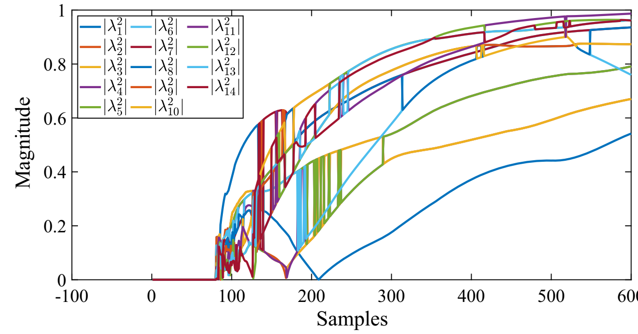
Table 1 Comparative analysis according to RMSE values for validation of the obtained models based on the proposed methodology and the methodologies proposed in Santos and Barreto (2018) (industrial dryer—MIMO system)

Models	RMSE - y_1 (mean ± std)	RMSE - y_2 (mean ± std)	RMSE - y_3 (mean ± std)
FS-LSSVR	$0.635 \pm 9.10 \times 10^{-3}$	$0.179 \pm 4.50 \times 10^{-3}$	$\mathbf{0.458 \pm 7.40 \times 10^{-3}}$
RFS-LSSVR	$0.631 \pm 1.02 \times 10^{-2}$	$0.182 \pm 4.60 \times 10^{-3}$	$0.473 \pm 4.41 \times 10^{-3}$
R^{2}FS-LSSVR	$0.641 \pm 8.02 \times 10^{-3}$	$0.178 \pm 2.79 \times 10^{-3}$	$0.478 \pm 5.90 \times 10^{-3}$
Proposed	$\mathbf{0.484 \pm 1.02 \times 10^{-7}}$	$\mathbf{0.089 \pm 1.07 \times 10^{-7}}$	$0.646 \pm 1.51 \times 10^{-7}$

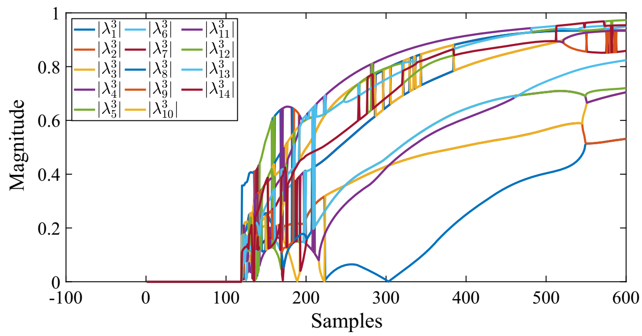
The better results are given in bold



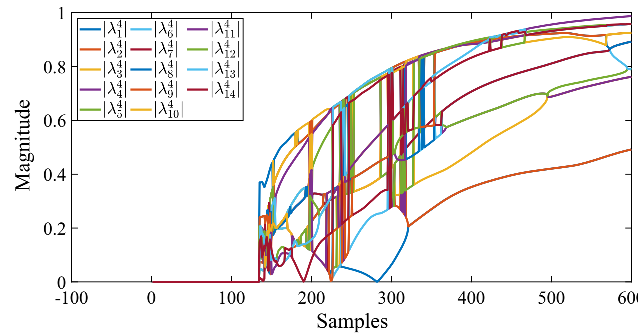
(a) eigenvalues of 1-st sub-model



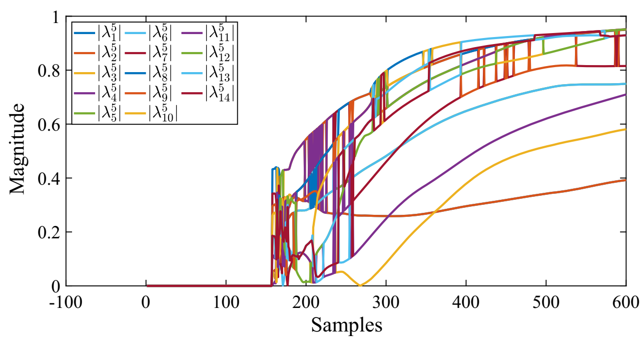
(b) eigenvalues of 2-nd sub-model



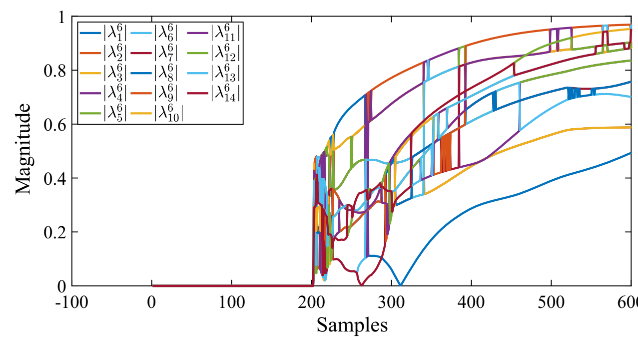
(c) eigenvalues of 3-rd sub-model



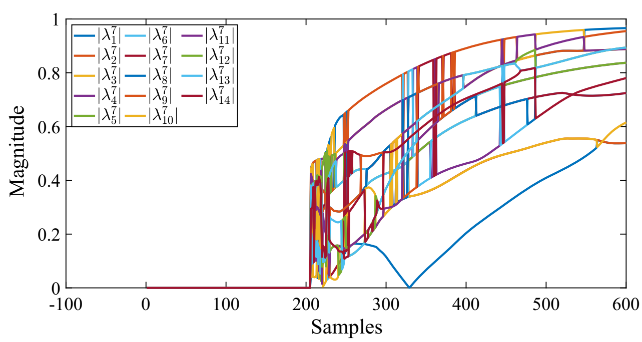
(d) eigenvalues of 4-th sub-model



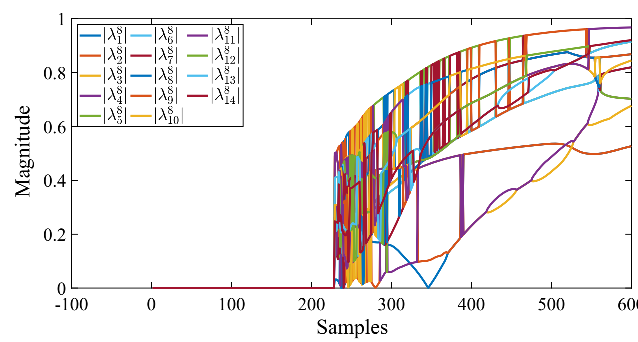
(e) eigenvalues of 5-th sub-model



(f) eigenvalues of 6-th sub-model

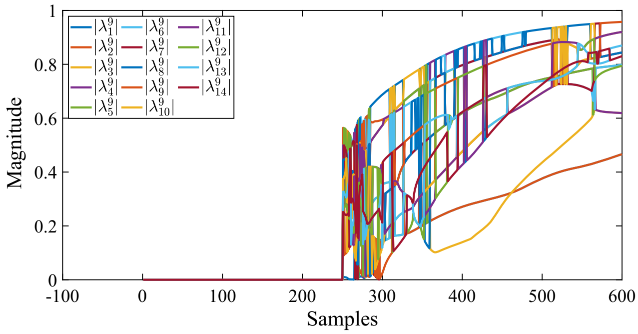


(g) eigenvalues of 7-th sub-model

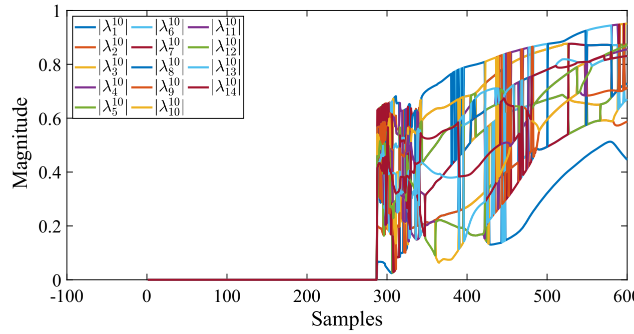


(h) eigenvalues of 8-th sub-model

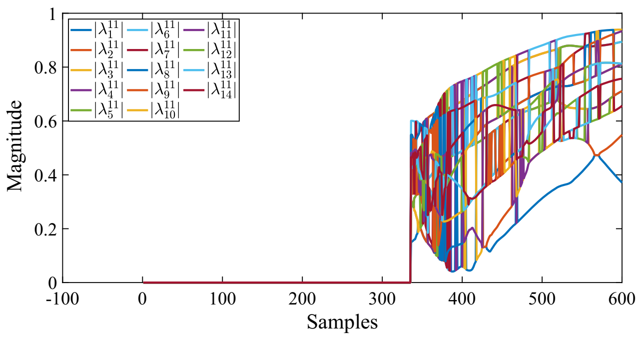
Fig. 8 Variation in the eigenvalues of $\mathbf{A}^i|^{i=1,\ldots,16}$ during the training step, where it is observed that each submodel of the interval type-2 neural-fuzzy model are stable, since the magnitude of the eigenvalues is less than 1



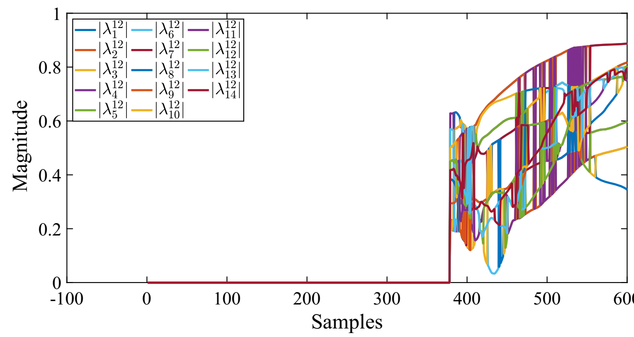
(i) eigenvalues of 9-th sub-model



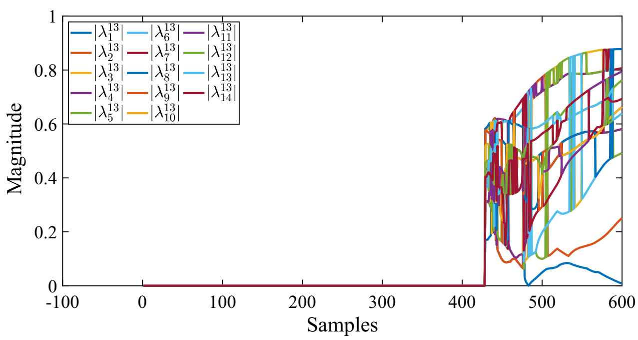
(j) eigenvalues of 10-th sub-model



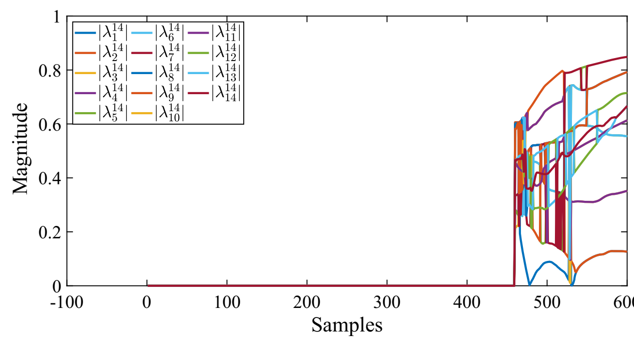
(k) eigenvalues of 11-th sub-model



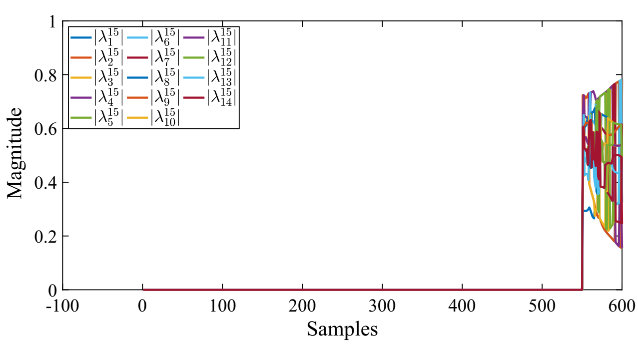
(l) eigenvalues of 12-th sub-model



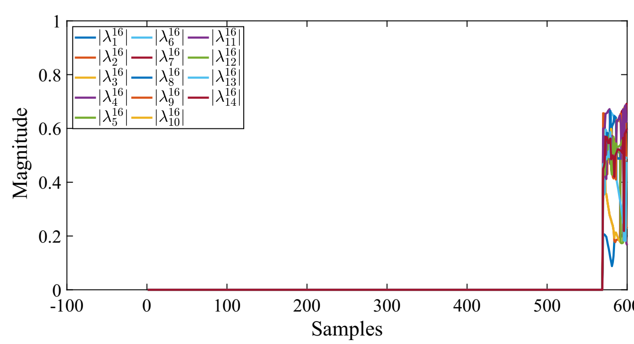
(m) eigenvalues of 13-th sub-model



(n) eigenvalues of 14-th sub-model



(o) eigenvalues of 15-th sub-model



(p) eigenvalues of 16-th sub-model

Fig. 8 continued

covariance matrix—EKF); $r_4 = 10^4$ (constant initialization of covariance matrix—Markov Parameters estimation); $r_5 = 10^2$ (constant initialization of covariance matrix—[**A B**] estimation); $r_6 = 10^3$ (constant initialization of covariance matrix—[**C D**] estimation); $q_p = 13$ (past window size); $q_f = 6$ (future window size); $n = 13$ (order of

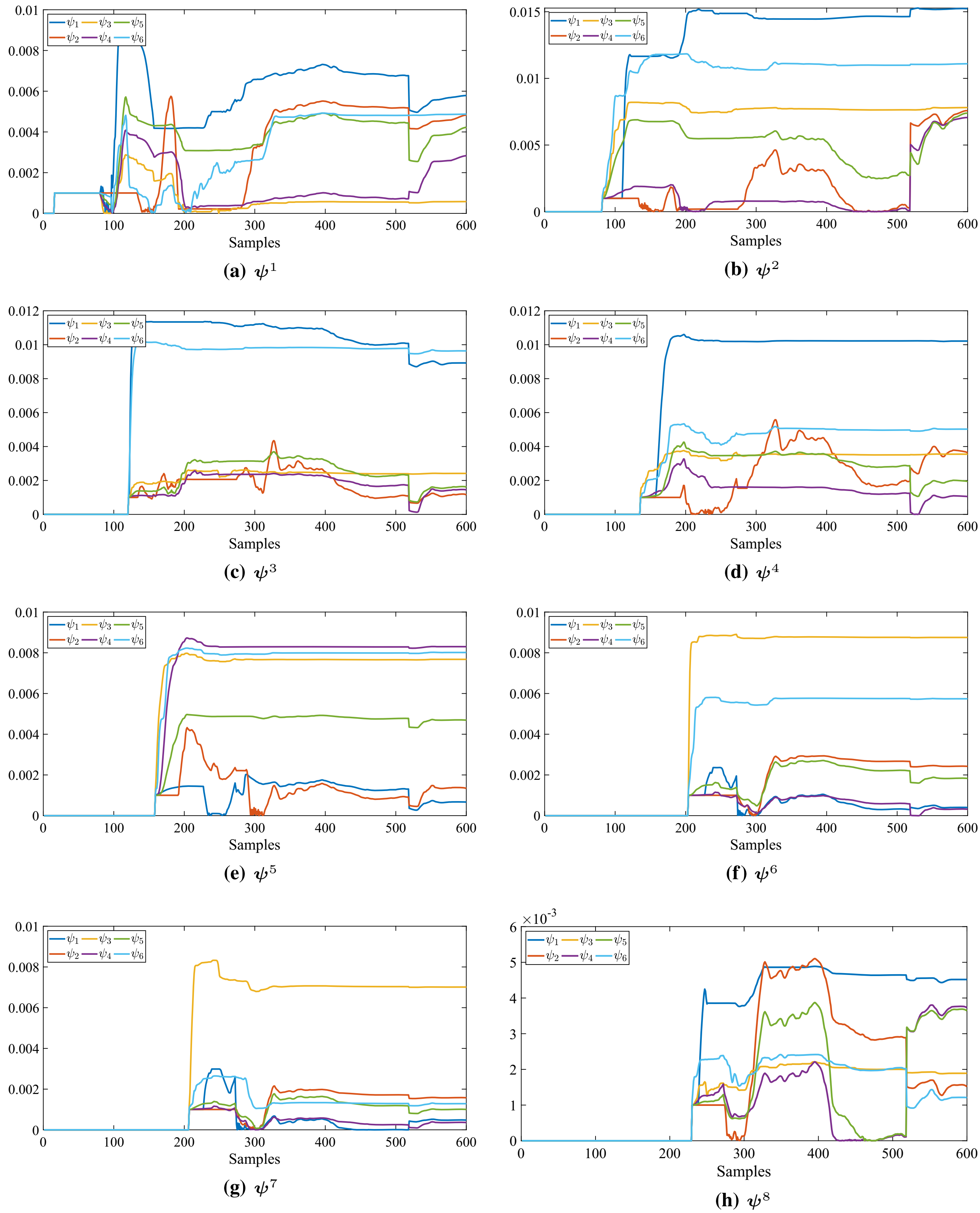


Fig. 9 Recursive estimation of $\psi^i|^{i=1,\ldots,16}$ which represents the degrees of uncertainty associated with the membership functions of the interval type-2 neural-fuzzy model for industrial dryer identification

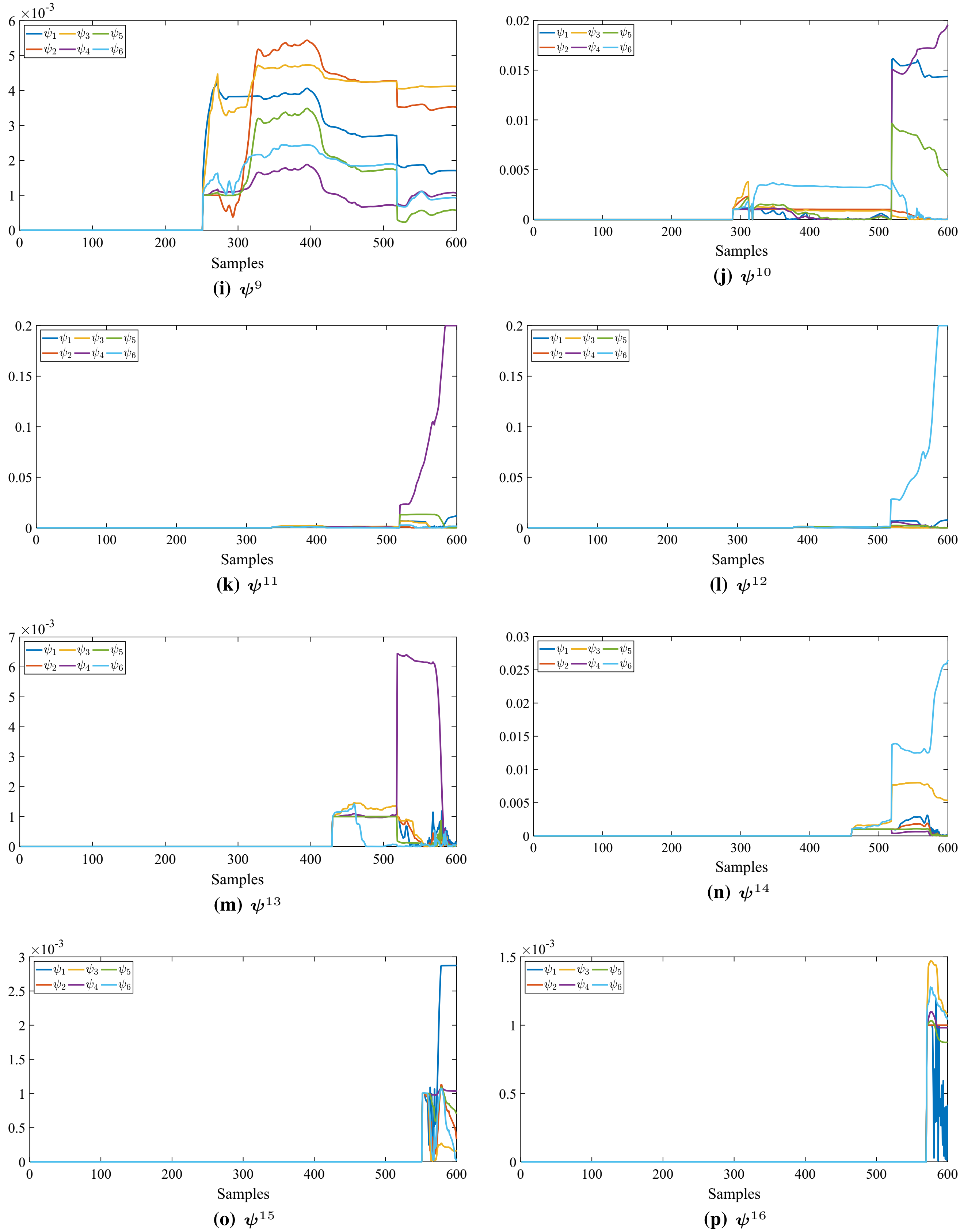


Fig. 9 continued

the system); $\mathbf{S} = [\mathbf{I}^{n\times n}\ \mathbf{0}^{n\times(q_p q_f - n)}]$; and $\kappa = 0$ (constant of initialization of consequent parameters). For this application, the first 600 samples are used for online training and the last 267 samples make up the experimental data set for validation. At the beginning of online training step, type-2 neural-fuzzy model has no rules, i.e., the initial condition used in the simulations considers that all parameters are set up to zero and, after that, the knowledge base is estimated from data stream.

The evolving procedure is illustrated in Fig. 4, where it can be observed the variation in the number of rules. At sample 81, the condition A (Eq. 24) is satisfied and the second rule is created; at sample 120, the third rule is created; at sample 135, the fourth rule is created; the number of rule is 5 at sample 158; the number of rule is 6 at sample 203; the number of rule is 10 at sample 288; the number of rule is 15 at sample 551; the number of rule is 16 at sample 570; and finally, the interval type-2 neural-fuzzy inference system has 17 rules. It can be noted that the number of rules of the fuzzy model is changed according to a new dynamic behavior of the industrial dryer, i.e., the structure of the interval type-2 neural-fuzzy inference system changes according to the data stream, adaptively.

For evaluation of the proposed methodology, the following quality measures are defined:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2} \tag{100}$$

where y is the plant output, $\hat{y}$ is the estimated output and N is the size of the data set.

The estimation of the industrial dryer outputs $\mathbf{y} = [y_1,\ y_2,\ y_3]$ performed in the training step is shown in Fig. 5, and in the validation step is shown in Fig. 6. The comparative analyses using RMSE criterion are shown in Table 1. According to Table 1, some potentialities of the proposed methodology are demonstrated, such as partition and representation of the complex operating range of the industrial dryer through the stable estimation (see Sect. 3.5) of local MIMO linear dynamic models, where the nonlinearities and uncertainties are represented by combination and definition of the interval type-2 multivariable Gaussian membership functions from a data stream, respectively. However, it is observed a higher RMSE value for estimating the output y_3. The cause of that, compared to estimates of outputs y_1 and y_2, can be the high variability in the behavior of output y_3, due possibly to noise, serving as a limiting factor for learning procedures. This limitation can be overcome via instrumental variables methods (Serra and Bottura 2007; Filho and Serra 2018), consisting in processing the uncorrelated noise from the data stream, which will be considered for further works.

The recursive estimation of the parameters $\upsilon_j^{x,l}$, $\upsilon_j^{x,r}$, $\upsilon_j^{y,l}$ and $\upsilon_j^{y,r}$ of type reducer layer is shown in Fig 7, which were used for performing the weighted sum between the lower and upper limits of uncertainty region obtained in layer IV.

The magnitude of eigenvalues computed from recursive estimation of transition matrices $\mathbf{A}^i|^{i=1,\ldots,16}$, in training step, is shown in Fig. 8, whose behavior supports the assumption that transition matrices are nilpotent, implying that each submodel is stable, according to the theoretical foundations presented in Sect. 3.2. Once the i-th state-space linear model represents the linear dynamic behavior of the industrial dryer in an operating point defined from each new sample, respectively, it can be concluded that, from the proposed methodology, the efficiency of the obtained evolving interval type-2 neural-fuzzy inference system, through the evolving linear combination of state-space models in the consequence, in tracking the nonlinearity inherent to the experimental data set, recursively.

The estimation of vector $\boldsymbol{\psi}^i|^{i=1,\ldots,16}$, which represents the uncertainties on shape of i-th membership function, is shown in Fig 9, where it is observed that the proposed methodology estimates the uncertainty degree $\boldsymbol{\psi}^i$ from experimental data, adjusting the footprint of uncertainty on the interval type-2 membership functions, adaptively.

According to Figs. 7, 8 and 9, i.e., considering the temporal response of recursive estimation of the coefficients related to type reducer, the eigenvalues of transition matrices and coefficients related to degree of uncertainty, in general, the numerical stability of the proposed approach can be inferred, once their bounded behavior were guaranteed in processing the experimental data, implying to important properties as consistency and convergence (Mendel 1995).

5 Conclusions

In this paper, an approach to evolving type-2 neural-fuzzy identification of multivariable dynamic systems was proposed. Considering the experimental results, some aspects were highlighted, such as adaptive fully data driven, once the structure of the interval type-2 neural-fuzzy inference system is changed according to the new behavior from experimental data; robustness to outliers, once the interval type-2 fuzzy sets are estimated according to data stream, computing the uncertainty degree on the antecedent, adaptively; and the parametric estimation are convergent and stable according to conditions in Sects. 3.4 and 3.5. The computational efficiency and applicability were justified by the following:

– The computation and processing of the Henkel matrix, for each rule, as well as its decomposition into singular values, were not necessary. This factor increases the exe-

cution speed and reduces the storage cost, once that less variables were used to estimate the consequent parameters.
– The evolving methodology was initiated considering a neural-fuzzy model without rules, so that all rules were born from the data stream.

For further works, the use of QR decomposition with Householder reflections to guarantee numerical robustness in estimation of the Markov parameters and state-space matrices, in order to overcome possible problems related to large time of parametric convergence, as well as the use of instrumental variable methods for overcoming the high variability of data stream due to presence of noise, are of particular interest.

References

Abiyev, R. H., Kaynak, O., Alshanableh, T., & Mamedov, F. (2011). A type-2 neuro-fuzzy system based on clustering and gradient techniques applied to system identification and channel equalization. *Applied Soft Computing*, *11*(1), 1396–1406. https://doi.org/10.1016/j.asoc.2010.04.011.

Abiyev, R. H., Kaynak, O., & Kayacan, E. (2013). A type-2 fuzzy wavelet neural network for system identification and control. *Journal of the Franklin Institute*, *350*(7), 1658–1685. https://doi.org/10.1016/j.jfranklin.2013.04.020.

Angelov, P., & Buswell, R. (2002). Identification of evolving fuzzy rule-based models. *IEEE Transactions on Fuzzy Systems*, *10*(5), 667–677.

Antonelli, M., Bernardo, D., Hagras, H., & Marcelloni, F. (2017). Multiobjective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification. *IEEE Transactions on Fuzzy Systems*, *25*(2), 249–264. https://doi.org/10.1109/tfuzz.2016.2578341.

Babuska, R. (2012). *Fuzzy modeling for control*. Amsterdam: Springer.

Castillo, O. (2011). *Type-2 fuzzy logic in intelligent control applications*. Amsterdam: Springer.

Chiuso, A., & Picci, G. (2005). Consistency analysis of some closed-loop subspace identification methods. *Automatica*, *41*(3), 377–391. https://doi.org/10.1016/j.automatica.2004.10.015.

Chou, C. T., & Maciejowski, J. (1997). System identification using balanced parametrizations. *IEEE Transactions on Automatic Control*, *42*(7), 956–974. https://doi.org/10.1109/9.599975.

Costa, E. B. M, & Serra, G. L. O. (2015). Optimal recursive fuzzy model identification approach based on particle swarm optimization. In *2015 IEEE 24th international symposium on industrial electronics (ISIE)*. Institute of Electrical & Electronics Engineers (IEEE).

Das, A. K., Subramanian, K., & Sundaram, S. (2015). An evolving interval type-2 neurofuzzy inference system and its metacognitive sequential learning algorithm. *IEEE Transactions on Fuzzy Systems*, *23*(6), 2080–2093. https://doi.org/10.1109/tfuzz.2015.2403793.

El-Nagar, A. M. (2018). Nonlinear dynamic systems identification using recurrent interval type-2 TSK fuzzy neural network: A novel structure. *ISA Transactions*, *72*, 205–217. https://doi.org/10.1016/j.isatra.2017.10.012.

Filho, O. D. R., & Serra, G. L. O. (2018). Recursive fuzzy instrumental variable based evolving neuro-fuzzy identification for non-stationary dynamic system in a noisy environment. *Fuzzy Sets and Systems*, *338*, 50–89. https://doi.org/10.1016/j.fss.2017.05.016.

Gao, Y., Liu, J., Wang, Z., & Wu, L. (2019a). Interval type-2 FNN-based quantized tracking control for hypersonic flight vehicles with prescribed performance. In *IEEE transactions on systems, man, and cybernetics: systems* (pp. 1–13). https://doi.org/10.1109/tsmc.2019.2911726.

Gao, Y., Xiao, F., Liu, J., & Wang, R. (2019b). Distributed soft fault detection for interval type-2 fuzzy-model-based stochastic systems with wireless sensor networks. *IEEE Transactions on Industrial Informatics*, *15*(1), 334–347. https://doi.org/10.1109/tii.2018.2812771.

Hagras, H. A. (2004). A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on Fuzzy Systems*, *12*(4), 524–539. https://doi.org/10.1109/tfuzz.2004.832538.

Han, M., Zhong, K., Qiu, T., & Han, B. (2019). Interval type-2 fuzzy neural networks for chaotic time series prediction: A concise overview. *IEEE Transactions on Cybernetics*, *49*(7), 2720–2731. https://doi.org/10.1109/tcyb.2018.2834356.

Haykin, S. S. (2001). *Kalman filtering and neural networks*. New York: Wiley.

Houtzager, I., van Wingerden, J. W., & Verhaegen, M. (2012). Recursive predictor-based subspace identification with application to the real-time closed-loop tracking of flutter. *IEEE Transactions on Control Systems Technology*, *20*(4), 934–949. https://doi.org/10.1109/tcst.2011.2157694.

Jer-Nan Juang, M. Q. P. (2011). *Identification and control of mechanical systems*. Cambridge: Cambridge University Press.

Juang, C. F., & Tsao, Y. W. (2008). A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning. *IEEE Transactions on Fuzzy Systems*, *16*(6), 1411–1424. https://doi.org/10.1109/tfuzz.2008.925907.

Juang, J. (1994). *Applied system identification*. Upper Saddle River: Prentice Hall.

Karnik, N. N., & Mendel, J. M. (1998). Introduction to type-2 fuzzy logic systems. In *1998 IEEE international conference on fuzzy systems proceedings. IEEE world congress on computational intelligence (Cat. No.98CH36228)*. IEEE. https://doi.org/10.1109/fuzzy.1998.686240.

Karnik, N. N., Mendel, J. M., & Liang, Q. (1999). Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, *7*(6), 643–658. https://doi.org/10.1109/91.811231.

Kayacan, E., Kayacan, E., & Khanesar, M. A. (2015). Identification of nonlinear dynamic systems using type-2 fuzzy neural networks–a novel learning algorithm and a comparative study. *IEEE Transactions on Industrial Electronics*, *62*(3), 1716–1724. https://doi.org/10.1109/tie.2014.2345353.

Kim, E., Park, M., Kim, S., & Park, M. (1998). A transformed input-domain approach to fuzzy modeling. *IEEE Transactions on Fuzzy Systems*, *6*(4), 596–604. https://doi.org/10.1109/91.728458.

Kumar, A., & Kumar, V. (2017). Evolving an interval type-2 fuzzy PID controller for the redundant robotic manipulator. *Expert Systems with Applications*, *73*, 161–177. https://doi.org/10.1016/j.eswa.2016.12.029.

Lee, C. H., Chang, F. Y., & Lin, C. M. (2014). An efficient interval type-2 fuzzy CMAC for chaos time-series prediction and synchronization. *IEEE Transactions on Cybernetics*, *44*(3), 329–341. https://doi.org/10.1109/tcyb.2013.2254113.

Lemos, A., Caminhas, W., & Gomide, F. (2011). Multivariable gaussian evolving fuzzy modeling system. *IEEE Transactions on Fuzzy Systems*, *19*(1), 91–104. https://doi.org/10.1109/tfuzz.2010.2087381.

Li, H., Wu, C., Shi, P., & Gao, Y. (2015). Control of nonlinear networked systems with packet dropouts: Interval type-2 fuzzy model-based approach. *IEEE Transactions on Cybernetics*, *45*(11), 2378–2389. https://doi.org/10.1109/tcyb.2014.2371814.

Li, H., Gao, Y., Shi, P., & Lam, H. K. (2016). Observer-based fault detection for nonlinear systems with sensor fault and limited communication capacity. *IEEE Transactions on Automatic Control*, *61*(9), 2745–2751. https://doi.org/10.1109/tac.2015.2503566.

Liang, Q., & Mendel, J. M. (2000a). Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters. *IEEE Transactions on Fuzzy Systems*, *8*(5), 551–563. https://doi.org/10.1109/91.873578.

Liang, Q., & Mendel, J. M. (2000b). Interval type-2 fuzzy logic systems: Theory and design. *The IEEE Transactions on Fuzzy Systems*, *8*(5), 535–550.

Lin, C. T., Pal, N. R., Wu, S. L., Liu, Y. T., & Lin, Y. Y. (2015). An interval type-2 neural fuzzy system for online system identification and feature elimination. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(7), 1442–1455. https://doi.org/10.1109/tnnls.2014.2346537.

Lin, Y. Y., Chang, J. Y., Pal, N. R., & Lin, C. T. (2013). A mutually recurrent interval type-2 neural fuzzy system (MRIT2nfs) with self-evolving structure and parameters. *IEEE Transactions on Fuzzy Systems*, *21*(3), 492–509. https://doi.org/10.1109/tfuzz.2013.2255613.

Lin, Y. Y., Liao, S. H., Chang, J. Y., & Lin, C. T. (2014). Simplified interval type-2 fuzzy neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(5), 959–969. https://doi.org/10.1109/tnnls.2013.2284603.

Ljung, L. (1999). *System identification: Theory for the user*. Boston: Prince Hall.

Ljung, S., & Ljung, L. (1985). Error propagation properties of recursive least-squares adaptation algorithms. *Automatica*, *21*(2), 157–167. https://doi.org/10.1016/0005-1098(85)90110-4.

Lughofer, E. (2008). FLEXFIS: A robust incremental learning approach for evolving Takagi–Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, *16*(6), 1393–1410.

Maciel, L., Gomide, F., & Ballini, R. (2013). Enhanced evolving participatory learning fuzzy modeling: An application for asset returns volatility forecasting. *Evolving Systems*, *5*(2), 75–88. https://doi.org/10.1007/s12530-013-9099-0.

Mendel, J. (2014). *Introduction to type-2 fuzzy logic control*. Hoboken: Wiley-Blackwell.

Mendel, J. M. (1995). Lessons in estimation theory for signal processing, communications, and control. Prentice Hall. https://www.ebook.de/de/product/6392380/jerry_m_mendel_lessons_in_estimation_theory_for_signal_processing_communications_and_control.html.

Mendel, J. M., & John, R. I. B. (2002). Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, *10*(2), 117–127. https://doi.org/10.1109/91.995115.

Mendel, J. M., John, R. I., & Liu, F. (2006). Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, *14*(6), 808–821. https://doi.org/10.1109/tfuzz.2006.879986.

Mohammadzadeh, A., Ghaemi, S., Kaynak, O., & Khanmohammadi, S. (2016). Robust H_∞-based synchronization of the fractional-order chaotic systems by using new self-evolving nonsingleton type-2 fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, *24*(6), 1544–1554.

Ni, Z., Liu, J., & Wu, Z. (2018a). Identification of the time-varying modal parameters of a spacecraft with flexible appendages using a recursive predictor-based subspace identification algorithm. *Proceedings of the Institution of Mechanical Engineers. Part G: Journal of Aerospace Engineering*,. https://doi.org/10.1177/0954410018770560.

Ni, Z., Liu, J., Wu, Z., & Shen, X. (2018b). Identification of the state-space model and payload mass parameter of a flexible space manipulator using a recursive subspace tracking method. *Chinese Journal of Aeronautics*,. https://doi.org/10.1016/j.cja.2018.05.005.

Nie, M., & Tan, W. W. (2008). Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In *2008 IEEE international conference on fuzzy systems (IEEE World Congress on Computational Intelligence)*. IEEE. https://doi.org/10.1109/fuzzy.2008.4630559.

Oscar Castillo, P. M. (2007). *Type-2 fuzzy logic: Theory and applications*. Berlin: Springer.

Pires, D., & Serra, G. (2018). An approach for fuzzy Kalman filter modeling based on evolving clustering of experimental data. *Journal of Intelligent and Fuzzy Systems*, *35*(2), 1819–1834. https://doi.org/10.3233/JIFS-17087.

Pratama, M., Lu, J., Lughofer, E., Zhang, G., & Er, M. J. (2017). An incremental learning of concept drifts using evolving type-2 recurrent fuzzy neural networks. *IEEE Transactions on Fuzzy Systems*, *25*(5), 1175–1192. https://doi.org/10.1109/tfuzz.2016.2599855.

Santos, J. D. A., & Barreto, G. A. (2018). Novel sparse LSSVR models in primal weight space for robust system identification with outliers. *Journal of Process Control*, *67*, 129–140. https://doi.org/10.1016/j.jprocont.2017.04.001.

Serra, G., & Bottura, C. (2007). An IV-QR algorithm for neuro-fuzzy multivariable online identification. *IEEE Transactions on Fuzzy Systems*, *15*(2), 200–210. https://doi.org/10.1109/tfuzz.2006.879997.

Silva, A., Caminhas, W., Lemos, A., & Gomide, F. (2014). Real-time nonlinear modeling of a twin rotor MIMO system using evolving neuro-fuzzy network. In *2014 IEEE symposium on computational intelligence in control and automation (CICA)*. IEEE. https://doi.org/10.1109/cica.2014.7013229.

Tung, S., Quek, C., & Guan, C. (2013). eT2fis: An evolving type-2 neural fuzzy inference system. *Information Sciences*, *220*, 124–148.

Wang, J., & Kumbasar, T. (2019). Parameter optimization of interval type-2 fuzzy neural networks based on PSO and BBBC methods. *IEEE/CAA Journal of Automatica Sinica*, *6*(1), 247–257. https://doi.org/10.1109/JAS.2019.1911348.

Wang, J., Luo, W., Liu, J., & Wu, L. (2019). Adaptive type-2 FNN-based dynamic sliding mode control of DC-DC boost converters. In *IEEE transactions on systems, man, and cybernetics: systems* (pp. 1–12). https://doi.org/10.1109/tsmc.2019.2911721.

Wu, C. Y., Tsai, J. S. H., Guo, S. M., Shieh, L. S., Canelon, J. I., Ebrahimzadeh, F., et al. (2015). A novel on-line observer/kalman filter identification method and its application to input-constrained active fault-tolerant tracker design for unknown stochastic systems. *Journal of the Franklin Institute*, *352*(3), 1119–1151. https://doi.org/10.1016/j.jfranklin.2014.12.004.

Wu, D. (2013). Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: Overview and comparisons. *IEEE Transactions on Fuzzy Systems*, *21*(1), 80–99. https://doi.org/10.1109/tfuzz.2012.2201728.

Wu, H., & Mendel, J. M. (2002). Introduction to uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. In *10th IEEE international conference on fuzzy systems. (Cat. No.01CH37297)*. IEEE. https://doi.org/10.1109/fuzz.2001.1009041.

Yager, R. (1990). A model of participatory learning. *IEEE Transactions on Systems, Man, and Cybernetics*, *20*(5), 1229–1234. https://doi.org/10.1109/21.59986.

Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-1. *Information Sciences*, *8x*, 199–249.