Received: 27 February 2021 | Revised: 2 November 2021 | Accepted: 5 January 2022

DOI: 10.1002/rnc.6024

SPECIAL ISSUE ARTICLE

WILEY

Learning event-triggered control based on evolving data-driven fuzzy granular models

Luiz A. Q. Cordovil Jr[1] | **Pedro H. S. Coutinho**[1] | **Iury Bessa**[1,2] | **Márcia L. C. Peixoto**[1] | **Reinaldo Martínez Palhares**[3]

[1]Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Belo Horizonte, Minas Gerais, Brazil

[2]Department of Electricity, Federal University of Amazonas, Manaus, Amazonas, Brazil

[3]Department of Electronics Engineering, Federal University of Minas Gerais, Belo Horizonte, Minas Gerais, Brazil

Correspondence
Reinaldo Martínez Palhares, Department of Electronics Engineering, Federal University of Minas Gerais, Av. Antonio Carlos, 6627, Belo Horizonte, Minas Gerais 31270-901, Brazil.
Email: rpalhares@ufmg.br

Funding information
Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Grant/Award Numbers: 307933/2018-0, 164692/2020-7; Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Grant/Award Number: 001; Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), Grant/Award Number: PPM-00053-17; Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM)

Abstract

This article proposes a data-stream-driven event-triggered control strategy using evolving fuzzy models learned by granulation of input–output samples of nonlinear systems with unknown time-varying dynamics. The evolving fuzzy model is obtained online from a data stream ensuring data coverage based on the principle of justifiable granularity and controlled by an event-triggering learning mechanism dependent on the model accuracy. This evolving fuzzy model is used to design event-triggered fuzzy controller to stabilize networked control systems while reducing the used communication resources. The event-triggered learning mechanism is employed to determine the instants in which the event-triggered fuzzy controller should be redesigned. Numerical examples illustrate the effectiveness of the proposed learning event-triggered fuzzy control algorithm.

KEYWORDS

data-driven modeling, evolving fuzzy systems, fuzzy granular computing, learning event-triggered control

1 | INTRODUCTION

The data transmission between physical and computational layers has been increasingly integrated by shared communication networks.[1] In contrast to traditional feedback control systems, in which the plant, sensors, actuators, and controller are connected based on the point-to-point protocol, networked control systems (NCSs) provide more flexible architectures with reduced installation costs and better maintainability.[1,2] As a result, NCSs have been considered in several applications, such as car automation,[3] micro-grids,[4] and unmanned vehicles.[5] For more applications, see References 1 and 6. Nevertheless, the intrinsic limited network bandwidth has motivated the development of strategies to reduce communication resources consumption still preserving desired control performance requirements.

To avoid wasting scarce communication resources frequently encountered in data packet-based communication in NCSs, instead of usual purely time specifications, the state or output information is considered in resource-aware control techniques to reduce the number of transmissions while preserving closed-loop stability or performance.[7] The resource-aware control techniques include event-triggered control (ETC)[8-11] and self-triggered control.[7] In self-triggered schemes, the communication resources are saved without continuous monitoring of the state information. For instance, this strategy has been recently exploited for distributed control of nonlinear systems with unknown control directions.[12,13] The main advantage of ETC systems is the presence of a feedback mechanism introduced into the sampling and communication process, such that the measurement data are transmitted to the controller, only when it is necessary to ensure the closed-loop stability and/or performance requirements, thus offering the possibility of immediate compensation.[7] This is the main motivation of this work to focus on ETC. However, the application of ETC to complex systems demands accurate models that are able to cope with uncertainties and time-varying dynamics.

The task of modeling real-world systems consists of developing methodologies that capture the dynamics of processes, which might be nonlinear and time-varying, and describing them with a certain precision. The control of those systems requires the characterization of the input–output relationships. Fuzzy control has been effectively employed as modeling tool for controlling nonlinear and time-varying systems.[14] On the one hand, when mathematical models of the plant are available, fuzzy model-based control approaches can be employed to obtain an approximate or exact Takagi–Sugeno (T-S) fuzzy representation of the plant whose structure is adopted to construct an appropriate fuzzy control law.[15,16] Considering Lyapunov stability arguments, control design conditions are generally derived in the form of linear matrix inequalities (LMIs), which can be efficiently solved using semidefinite programming. On the other hand, when only partial information about the system's model is available, fuzzy adaptive approaches are used to obtain approximations for the unknown/uncertain terms of the plant.[17,18] Unfortunately, in the aforementioned works it is necessary that a fuzzy rule base be beforehand constructed. When the physical models are not available and only systems' observations (measurements) of inputs and outputs data, data-based approaches are suitable alternatives for controlling complex dynamics systems subject to time-varying dynamics, measurement errors, uncertainties, and even information unavailability.[19,20] Based on these aspects, the flexibility provided by data-driven approaches is a helpful property on control systems design, especially when it comes to adaptability and robustness.

Recently, data-based approaches to design ETC strategies have been attracted much attention, since the accurate description of complex systems is often unavailable. The existing approaches include, for instance, iterative learning control,[21] reinforcement learning control,[22,23] Q-learning,[24] model-free adaptive control,[25] and adaptive dynamic programming.[26] However, most of those approaches are unable to deal with uncertain time-varying dynamics. In Reference 27, it is shown that evolving fuzzy models can be effectively employed to design robust control strategies for nonlinear systems, possibly uncertain and time-varying, with unknown dynamics based only on the measured inputs and outputs. This data-driven fuzzy modeling is parameterized considering a data stream from scratch and the learning fuzzy controller is updated online, whenever the local fuzzy rules change, solving an LMI problem. Indeed, evolving fuzzy models are applied to a broad class of problems such as fault diagnosis and prognostics,[28] forecasting,[29] classification,[30] clustering,[31,32] and identification,[33] due to the flexibility and adaptability of the fuzzy rule base in terms of their parameters and semantics, which are useful properties to deal with (nonlinear) time-varying dynamics in data-stream-driven approaches.

In this article, a new learning-based ETC strategy is proposed. The main motivation is the problem of networked control of nonlinear systems with unknown and time-varying dynamics. To address this problem, a data-stream-driven learning process is proposed by employing the evolving ellipsoidal fuzzy information granules (EEFIGs) to obtain T-S fuzzy auto-regressive with exogenous inputs (ARX) models that are translated into a T-S fuzzy state-space representation. The model learning is performed through the justifiable formation and evolution of information granules to represent knowledge uncovered from data streams. The proposed data-driven control algorithm comprises the formation of a flexible knowledge base through evolving fuzzy systems that describes a system from interpretable representations in terms of semantically soundly local fuzzy models for closed-loop model learning. In summary, the main contributions of this work are:

- a new online learning ETC algorithm based on evolving fuzzy information granules for NCSs with nonlinear time-varying plants with unknown dynamics;
- an improved evolving fuzzy modeling algorithm called EEFIG-ARX which provides proper T-S fuzzy ARX models for representing nonlinear time-varying plants with unknown dynamics;

- the estimation of the consequent parameters of the fuzzy information granules by using a novel fuzzy weighted recursive least squares with variable-direction forgetting (FWRLS-VDF). This estimator is the recursive least squares with variable-direction forgetting (RLS-VDF)[34] weighted by fuzzy membership degrees which it is suitable for evolving control applications, since it prevents excitation persistence loss;
- a new event triggering learning mechanism which is proposed to avoid unnecessary controller updates while ensuring the minimum tolerated model accuracy.

The remaining of this article is organized as follows. Section 2 presents the problem formulation concerning simultaneous ETC and evolving modeling of nonlinear systems with unknown dynamics. Section 3 introduces the EEFIG-ARX algorithm for model learning and consequent parameterization through FWRLS-VDF. Section 4 presents the ETC design based on evolving fuzzy models and provides an overview of the integrated learning control algorithm. Section 5 presents numerical simulation examples for assessing the proposed learning ETC approach. Finally, Section 6 points out the achievements, drawbacks, and future perspectives of this work.

Notation: $\mathbb{N}$ denotes the set of natural numbers, $\mathbb{N}_{\leq p}$ denotes the set of natural numbers less than or equal to $p \in \mathbb{N}$, $\mathbb{R}$ denotes the field of real numbers, and $\mathbb{R}_{\geq 0}$ ($\mathbb{R}_{>0}$) denotes the set of all non-negative (positive) real numbers. The Euclidean norm is denoted by $\| \cdot \|$. In a symmetric matrix, the symbol "$\star$" denotes the term deduced by symmetry and $\text{diag}(X_1, \dots, X_n)$ denotes the block diagonal matrix of matrices $X_1, \dots, X_n$. For vectors $x_i \in \mathbb{R}^{n_i}, i \in \mathbb{N}_{\leq p}$ the vector $[x_1^\top, x_2^\top, \dots, x_p^\top]^\top \in \mathbb{R}^n$, with $n = \sum_{i=1}^{p} n_i$, is denoted by $(x_1, x_2, \dots, x_p)$.

2 | PRELIMINARIES

Consider the setup depicted in Figure 1, where the discrete-time plant $\mathcal{P}$, possibly nonlinear and time-varying, is connected to the controller $\mathcal{C}$ through a general-purpose network. It is assumed that there is no knowledge on the plant's dynamic model, only the input and output signals are available for measurement. The plant is described as

$$y_k = F\left(y_{k-1}, \dots, y_{k-\tau}, u_{k-1}, \dots, u_{k-\delta}, k\right), \tag{1}$$

where $y_k \in \mathbb{R}^n$ is the measured output, $u_k \in \mathbb{R}^m$ is the input vector, the map $F(\cdot)$ is unknown, and the variables τ and δ denotes the order of the ARX model with respect to the auto-regressive part and input, respectively, that is, τ denotes the maximum lag of y and δ denotes the maximum lag of u in the structure of the nonlinear map $F(\cdot)$.

Let $\mathcal{D} = (z, k)$ be a data stream, where z is the sequence of input u and output y samples and $k \in \mathbb{N}_0$ is the sequence of positive integers. For the sake of simplicity, the elements of $\mathcal{D}$ are denoted by $z_k = (u_k, y_k) \in \mathbb{R}^{n+m}$. Given the observations

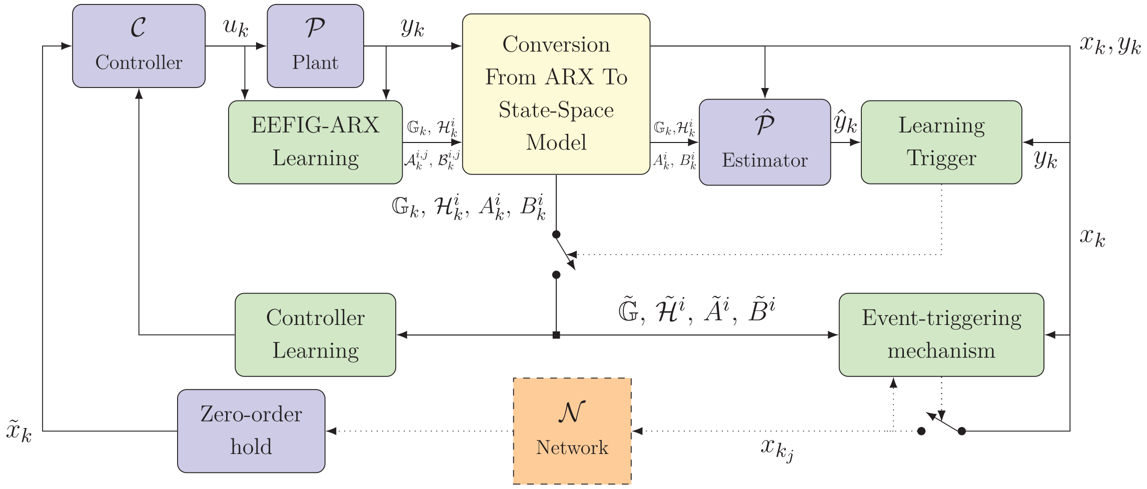
$\mathcal{C}$
Controller
u_k
$\mathcal{P}$
Plant
y_k
Conversion From ARX To State-Space Model
x_k, y_k
EEFIG-ARX Learning
$\mathbb{G}_k, \mathcal{H}_k^i$
$\mathcal{A}_k^{i,j}, \mathcal{B}_k^{i,j}$
$\mathbb{G}_k, \mathcal{H}_k^i$
A_k^i, B_k^i
$\hat{\mathcal{P}}$
Estimator
$\hat{y}_k$
Learning Trigger
y_k
$\mathbb{G}_k, \mathcal{H}_k^i, A_k^i, B_k^i$
x_k
Controller Learning
$\tilde{\mathbb{G}}, \tilde{\mathcal{H}}^i, \tilde{A}^i, \tilde{B}^i$
Event-triggering mechanism
$\tilde{x}_k$
Zero-order hold
$\mathcal{N}$
Network
x_{k_j}

FIGURE 1 Representation of an ETC control setup, where $\mathcal{P}$ is the discrete-time plant, $\mathcal{C}$ is the controller, $\mathcal{N}$ is the network communication channel, x_k is the continuous state measurement, x_{k_j} is the most recently transmitted state measurement, $\tilde{x}_k$ is x_{k_j} under the action of the ZOH, y_k is the measured output, $\hat{y}_k$ is the one-step-ahead output prediction, and u_k is the control input

of z_k, the EEFIG[35] is used to estimate an evolving fuzzy model. The EEFIG model is composed by a collection of granules $\mathbb{G}_k = \{\mathcal{G}_k^1, \dots, \mathcal{G}_k^N\}$ that spans the following rule-based fuzzy model

$$\begin{aligned} \textbf{Rule } i : \ & \text{IF } z_k \text{ is } \mathcal{G}_k^i, \\ & \text{THEN } \hat{y}_k^i = F_k^i\left(y_{k-1}, \dots, y_{k-\tau}, u_{k-1}, \dots, u_{k-\delta}\right), \end{aligned} \tag{2}$$

where $\mathcal{G}_k^i$ is the ith multidimensional fuzzy set, the EEFIG, at the kth time instant whose structure and normalized membership functions g_k^i at the kth time instant are to be specified later. Thus, using the center-of-gravity defuzzification, the estimate of (1) is obtained by means of (2) as

$$\hat{y}_k = \sum_{i=1}^{N} g_k^i(z_k)\hat{y}_k^i, \quad g_k^i(z_k) = \frac{\omega_k^i(z_k)}{\sum_{i=1}^{N} \omega_k^i(z_k)}, \tag{3}$$

where $\omega_k^i : \mathbb{R}^{n+m} \to [0, 1]$ is the activation rule of the ith EEFIG at the kth time instant.

The estimation error due to the model approximation is defined as follows

$$\epsilon_k = \hat{y}_k - y_k. \tag{4}$$

Although evolving fuzzy models have been successfully employed to drive learning control strategies[27] with continuously learned models, performing continuously the learning task requires significant computational cost due to the model and control updates. Thus, in this article, it is used an event-triggered learning strategy that does not allow the model learning until it deviates significantly from the expected value, that is, when ϵ_k is higher than a tolerated value. This idea can be formulated as follows:

$$n_0 = 0, \ n_{j+1} = \min\{k \in \mathbb{N} : k > n_j \wedge \|\epsilon_k\| < \sigma\|y_k\|\}, \quad \forall j \in \mathbb{N}, \tag{5}$$

where $\sigma \in \mathbb{R}_{\geq 0}$ is a given constant.

As a result, the proposed learning control approach is not based on the model (2), but it uses a model that is only updated after a learning event, as described in (5):

$$\begin{aligned} \textbf{Rule } i : \ & \text{IF } z_k \text{ is } \tilde{\mathcal{G}}_k^i, \\ & \text{THEN } \hat{y}_k^i = \tilde{F}_k^i\left(y_{k-1}, \dots, y_{k-\tau}, u_{k-1}, \dots, u_{k-\delta}\right), \end{aligned} \tag{6}$$

where $\forall \mathcal{G}_k^i \in \mathbb{G}_k$ and $\forall k \in \{n_j, \dots, n_{j+1} - 1\}$:

$$\tilde{\mathbb{G}}_k = \left\{\tilde{\mathcal{G}}_k^1, \dots, \tilde{\mathcal{G}}_k^N\right\}, \quad \tilde{\mathcal{G}}_k^i = \mathcal{G}_{n_j}^i, \quad \tilde{F}_k^i\left(y_{k-1}, \dots, y_{k-\tau}, u_{k-1}, \dots, u_{k-\delta}\right) = F_{n_j}^i\left(y_{k-1}, \dots, y_{k-\tau}, u_{k-1}, \dots, u_{k-\delta}\right).$$

The notation $\tilde{\mathbb{G}}$ denotes the granular base which was learned at the last learning event n_j. The granular base $\tilde{\mathbb{G}}$ is composed by the information granules $\tilde{\mathcal{G}}^i$, for $i \in \mathbb{N}_{\leq N}$, whose consequents' parameters are denoted by $\tilde{F}^i$. Based on the evolving input–output model (6), the following state-space representation is obtained:

$$x_{k+1} = f(x_k, u_k), \quad y_k = h(x_k), \tag{7}$$

where $x_k \in \mathbb{R}^{n_x}$ is the state vector. This state-space model obtained from data is thus available for the ETC design. The event-triggering mechanism (ETM) determines the sample k_j to transmit the state x_{k_j} to the controller and the state information is available to the controller as $\tilde{x}_k$. Since $\{k_j\}_{j\in\mathbb{Z}_{\geq 0}}$ is a sub-sequence of $k \in \mathbb{Z}_{\geq 0}$, the ETC system can lead to fewer transmissions than the traditional feedback control setup counterpart.

The following state-feedback controller C is considered to stabilize the plant $\mathcal{P}$:

$$u_k = \phi(\tilde{x}_k), \tag{8}$$

where $\phi : \mathbb{R}^{n_x} \to \mathbb{R}^m$ is a continuously differentiable function and $\tilde{x}_k$ is the latest state information available to the controller. The following closed-loop system is obtained substituting the control law (8) into the plant dynamics (7):

$$x_{k+1} = f(x_k, \phi(\tilde{x}_k), k). \tag{9}$$

By assuming an ideal communication network, the premise variable and state measurements available to the controller are, respectively,

$$\tilde{z}_k = z_{k_j}, \quad \tilde{x}_k = x_{k_j}, \quad \forall k \in \{k_j, \dots, k_{j+1} - 1\}, \tag{10}$$

and the following transmission error is induced by the event-based sampling:

$$e_k = \tilde{x}_k - x_k, \quad \forall k \in \{k_j, \dots, k_{j+1} - 1\}. \tag{11}$$

To reduce the number of transmissions aiming to reduce the network resources' usage, a triggering mechanism is considered to determine the transmission instants as follows:

$$k_0 = 0, \; k_{j+1} = \min\{k \in \mathbb{N} : k > k_j \wedge \Gamma(x_k, e_k) \leq 0\}, \quad \forall j \in \mathbb{N}, \tag{12}$$

where $\Gamma(x_k, e_k)$ is a trigger function specified later.

Problem statement. Given the evolving model (2) learned under the policy in (5), design the state-feedback control law (8) and the ETM (12) that stabilizes the system (1).

3 | EVOLVING FUZZY GRANULAR MODELS

Evolving fuzzy models are self-learning fuzzy rule-based models which are commonly associated to data-stream-driven approaches and present useful properties in terms of incremental learning, such as flexibility regarding the rule base, and adaptability regarding the model parameters update performed along the real-time data processing.[36]

The evolving fuzzy modeling is able to provide fuzzy models that comprise nonlinear and also time-varying dynamics. In the case of control applications, evolving fuzzy models are useful for regulation of systems with unknown and time-varying dynamics as proposed in References 27 and 37, since the model learns to represent the nonlinearities and adapts itself to the time-varying behavior, therefore it is suitable for learning-based control design. The procedure for obtaining the evolving fuzzy models is summarized as follows: it is performed a granulation process on the data stream, which updates both antecedent and consequent of the fuzzy rule in the sense that a sequence of recursive operations is carried out for checking similarity on data supporting granules (rules) creation, and the adaptation on the granules' prototype and parameters.

In Reference 35, it is introduced an online data processing with the employment of evolving fuzzy information granules. Based on the parametric principle of justifiable granularity,[38] a new granularity allocation procedure for knowledge base formation from data streams is developed. This recent strategy is considered in this work to obtain evolving fuzzy models of nonlinear dynamical systems with time-varying and uncertain dynamics. Details about this strategy are provided in Sections 3.1 and 3.2.

3.1 | Evolving ellipsoidal fuzzy information granules

An EEFIG is a fuzzy set $\mathcal{G}_k^i = \left(\mathbb{R}^{n+m}, \omega_k^i\right)$, where $\omega_k^i : \mathbb{R}^{n+m} \to [0, 1]$ is the membership function of the EEFIG $\mathcal{G}_k^i$. The membership function ω_k^i is parameterized by the granular prototype $\mathcal{H}_k^i$ of the ith granule at the time instant k, which is also a numerical evidence basis for the granulation process. The granule prototype is defined as follows:

$$\mathcal{H}_k^i = \left(\underline{\mu}_k^i, \mu_k^i, \overline{\mu}_k^i, {\Sigma_k^i}^{-1}\right), \tag{13}$$

where $\underline{\mu}_k^i$, μ_k^i, and $\overline{\mu}_k^i$ are the lower, mean, and upper bound vectors of the ith EEFIG at k and $(\Sigma_k^i)^{-1}$ is the inverse of its covariance matrix. Given the granule prototype $\boldsymbol{\mathcal{P}}_k^i$, the membership function of an EEFIG is parameterized as

$$\omega_k^i(z_k) = \exp\left\{-\left[(z_k - \mu_k^i)^\top \left(\Delta_k^i\right)^{-1}(z_k - \mu_k^i)\right]^{1/2}\right\}, \tag{14}$$

where $\Delta_k^i = \text{diag}\left\{\left(\frac{\overline{\mu}_{k,1}^i - \underline{\mu}_{k,1}^i}{2}\right)^2, \dots, \left(\frac{\overline{\mu}_{k,p}^i - \underline{\mu}_{k,p}^i}{2}\right)^2\right\}$, $p \in \mathbb{N}_{\leq n+m}$, being $\overline{\mu}_k^i$ and $\underline{\mu}_k^i$ the semi-axes of the ith EEFIG prototype such that $\underline{\mu}_k^i < \mu_k^i < \overline{\mu}_k^i$.[39] Moreover, the distance of a given data sample z_k to the ith EEFIG is given by

$$d(z_k, \mu_k^i) = (z_k - \mu_k^i)^\top \Sigma_k^{i\,-1}(z_k - \mu_k^i), \tag{15}$$

which can be interpreted as the square of Mahalanobis distance.

The granulation process also consists in the formation of an evolving fuzzy rule base from the data stream. For this purpose, it is used the concept of data sample admissibility. A data sample z_k is said to be admitted by a given granule prototype $\mathcal{H}_k^i$ if it is used to update the granule prototype parameters. In this sense, two criteria are used to evaluate the data sample admissibility.

3.1.1 | Similarity

Based on the distance (d) of z_k to the EEFIG center μ_k^i and conditioning it to a threshold, the first level of admissibility (L_1) is reached:

$$L_1 \equiv d(z_k, \mu_k^i) < \nu, \tag{16}$$

where $d(z_k, \mu_k^i)$ is defined in (15) and $\nu = \left(\chi^2\right)^{-1}(\gamma, n+m)$ is a threshold parameterized by the inverse of chi-squared statistic with $n+m$ degrees of freedom, leading EEFIG prototype to cover around $100\gamma\%$ of the stream sample. In parallel, as the data samples are available and evaluated, it is established a structure named *Tracker* whose objective is to follow the data stream dynamics to indicate change points.

The *Tracker* is parameterized by a mean vector μ_k^{tr} and an inverse covariance matrix Σ_k^{tr}, which are recursively updated.[40] The tracker separation verification is made considering the c-separation condition.[35] In this sense, if the *Tracker* is c-separated from all the existing granule prototypes, a new granule is created considering data samples that do not reach **L1**, during a certain discrete time interval (T_s). The condition is verified as follows

$$c_{\text{s}} \equiv \left\|\mu_k^{\text{tr}} - \mu_k^i\right\| \geq c\sqrt{(n+m)\max(\xi_{\max}(\Sigma_k^{\text{tr}}), \xi_{\max}(\Sigma_k^i))}, \quad \forall \mathcal{G}_k^i \in \mathbb{G}_k, \tag{17}$$

where $\xi_{\max}$ is the largest eigenvalue of Σ_k and, $c \in [0, \infty)$ specifies the separation level. Here, c is assumed as 2.

3.1.2 | Effect on the granules' performance index ($\mathcal{Q}_k^i$)

The available data samples are used to update the granule prototype and its semantics. According to a granularity allocation procedure, z_k can be used to increase granules' coverage, updating its prototype. When it comes to semantics, the update of $\mathcal{H}_k^i$ directly implies on the antecedent and eventually on the consequent of each granular rule described in (2). As data are processed and grouped, the granularity allocation procedure is performed in terms of the existing granules, as summarized as follows:

Data sample contribution index: Each data sample $z_k \in \mathbb{R}^{n+m}$ is evaluated for each information granule $\mathcal{G}_k^i$ in terms of its distance to the granule prototype center, μ_k^i, and the effect on the parameterization of the granule membership function, which is posed in terms of the prototype $\mathcal{H}_k^i$. The data sample contribution index $\overline{\mathcal{Q}}_k^i$ is defined as

$$\overline{\mathcal{Q}}_k^i(z_k) = d(z_k, \mu_k^i)|\mathcal{G}_k^i|, \tag{18}$$

where $d(z_k, \mu_k^i)$ is defined in (15) and $| \cdot |$ is the fuzzy cardinality operator of the ith EEFIG.

Fuzzy cardinality variation analysis: To check the second admission level of a new data sample z_k by a granule prototype, it is necessary to update the fuzzy cardinality as following

$$|\mathcal{G}_k^i| = |\mathcal{G}_{k-1}^i| + g_k^i(z_k) - \frac{\partial g_k^i(z_k)}{\partial \mathcal{H}_k^i}, \tag{19}$$

where

$$\frac{\partial g_k^i(z_k)}{\partial \mathcal{H}_k^i} = \frac{\partial g_k^i(z_k)}{\partial \underline{\mu}_k^i}\Delta\underline{\mu}_k^{i\top} + \frac{\partial g_k^i(z_k)}{\partial \mu_k^i}\Delta\boldsymbol{\mu}_k^{i\top} + \frac{\partial g_k^i(z_k)}{\partial \overline{\mu}_k^i}\Delta\overline{\mu}_k^{i\top},$$

with

$$\frac{\partial g_k^i(z_k)}{\partial \underline{\mu}_k^i} = \left[\frac{\partial g_k^i(z_k)}{\partial \underline{\mu}_{k,1}^i} \cdots \frac{\partial g_k^i(z_k)}{\partial \underline{\mu}_{k,p}^i}\right], \ \frac{\partial g_k^i(z_k)}{\partial \mu_k^i} = \left[\frac{\partial g_k^i(z_k)}{\partial \mu_{k,1}^i} \cdots \frac{\partial g_k^i(z_k)}{\partial \mu_{k,p}^i}\right], \ \frac{\partial g_k^i(z_k)}{\partial \overline{\mu}_k^i} = \left[\frac{\partial g_k^i(z_k)}{\partial \overline{\mu}_{k,1}^i} \cdots \frac{\partial g_k^i(z_k)}{\partial \overline{\mu}_{k,p}^i}\right],$$
$$\Delta\underline{\mu}_k^i = \left[\underline{\mu}_{k,1}^i - \underline{\mu}_{k-1,1}^i \ \cdots \ \underline{\mu}_{k,p}^i - \underline{\mu}_{k-1,p}^i\right], \ \Delta\mu_k^i = \left[\mu_{k,1}^i - \mu_{k-1,1}^i \ \cdots \ \mu_{k,p}^i - \mu_{k-1,p}^i\right],$$
$$\Delta\overline{\mu}_k^i = \left[\overline{\mu}_{k,1}^i - \overline{\mu}_{k-1,1}^i \ \cdots \ \overline{\mu}_{k,p}^i - \overline{\mu}_{k-1,p}^i\right],$$

where each partial derivative is given by

$$\frac{\partial g_k^i(z_k)}{\partial \underline{\mu}_{k,j}^i} = 2g_k^i(z_k)\left(g_k^i(z_k) - 1\right)\frac{(z_{k,j} - \mu_{k,j}^i)^2}{(\Omega_k^i)^{12}(\overline{\mu}_{k,j}^i - \underline{\mu}_{k,j}^i)^3},$$
$$\frac{\partial g_k^i(z_k)}{\partial \mu_{k,j}^i} = -2g_k^i(z_k)\left(g_k^i(z_k) - 1\right)\frac{z_{k,j} - \mu_{k,j}^i}{(\Omega_k^i)^{12}(\overline{\mu}_{k,j}^i - \underline{\mu}_{k,j}^i)^2},$$
$$\frac{\partial g_k^i(z_k)}{\partial \overline{\mu}_{k,j}^i} = -2g_k^i(z_k)\left(g_k^i(z_k) - 1\right)\frac{(z_{k,j} - \mu_{k,j}^i)^2}{(\Omega_k^i)^{12}(\overline{\mu}_{k,j}^i - \underline{\mu}_{k,j}^i)^3},$$

with $\Omega_k^i = \sum_{j=1}^{p}\left(\frac{z_{k,j}-\mu_{k,j}^i}{b_{k,j}^i-a_{k,j}^i}\right)^2, j \in \mathbb{N}_{\leq p}$. Notice that the derivative of $\mathcal{H}_k^i$ with respect to $\Sigma_k^{i\,-1}$ is zero, as it is not directly related to the formulation of the membership function ω_k^i. The derivative term in (19) refers to the variation of z_k on the granules' membership function parameters. The current fuzzy cardinality can be kept or increased according to the incoming data evaluation. By this formulation, one can compute the data sample contribution index, which is intimately linked with the granules performance index as described in the sequel.

Granule's performance index: This is computed by the sum of the data sample contribution index of an EEFIG:

$$\mathcal{Q}_k^i = \sum_{j=1}^{k}\overline{\mathcal{Q}}_j^i(z_j). \tag{20}$$

The granularity allocation considering (20) is analyzed by the comparison between the performance index value at the kth instant with the previous one, that is, if $\mathcal{Q}_k^i > \mathcal{Q}_{k-1}^i$, then, all the parameters of the information granule prototype are updated. It is noteworthy the role of $|\mathcal{G}_k^i|$ in the update condition evaluation. Insofar as a data sample can change the granule membership function parameterization, the fuzzy cardinality is held or updated (increased) depending on its variation. Based on it, the second level of admissibility (L_2) is reached:

$$L_2 \equiv Q_k^i > Q_{k-1}^i. \tag{21}$$

Considering the evolving aspect and, if L_1 in (16) and L_2 in (21) are satisfied, the prototype parameters are recursively updated according to the data stream processing and a justifiable granularity allocation procedure

$$\mu_k^i = \mu_{k-1}^i + \frac{\omega_k^i(z_k)}{\sum_{i=1}^N \omega_k^i(z_k)}(z_k - \mu_k^i), \tag{22a}$$

$$\underline{\mu}_{k,p} = \max\{\mu_{k,p} - sL_{k,p}, 0\}, \tag{22b}$$

$$\overline{\mu}_{k,p} = \min\{\mu_{k,p} + sR_{k,p}, +\infty\}, \tag{22c}$$

$$\Sigma_k^{i^{-1}} = \Pi_{1_k^i}\left[\Sigma_k^{i^{-1}} - \frac{\Sigma_k^{i\ -1}(z_{k+1} - \mu_k^i)(z_{k+1} - \mu_k^i)^\top \Sigma_k^{i^{-1}}}{\Pi_{2_k^i} + (z_{k+1} - \mu_k^i)^\top \Sigma_k^{i^{-1}}(z_{k+1} - \mu_k^i)}\right], \tag{22d}$$

where $s \in [2, 4]$ as in Reference 39, and

$$L_{k,p}^i = \frac{L_{k-1,p}^i \cdot l_{k-1,p} + \mu_k^i - z_{k,p}}{l_{k,p}}, \quad R_{k,p}^i = \frac{R_{k-1,p}^i \cdot r_{k-1,p} + z_{k,p} - \mu_k^i}{r_{k,p}},$$

are the left-side mean and the right-side mean, respectively, with $l_{k,p}$ and $r_{k,p}$, $p \in \mathbb{N}_{\leq n+m}$ being the cardinality of the subsets $z_{k,p}^- = \{z_{k,p} \mid z_{k,p} < \mu_{k,p},\ \forall k \in \mathbb{N}_{>0}\}$ and $z_{k,p}^+ = \{z_{k,p} \mid z_{k,p} \geq \mu_{k,p},\ \forall k \in \mathbb{N}_{>0}\}$, and dimension of (z_k), respectively. Complementary,

$$\Pi_{1_k^i} = \frac{\pi_{2_k^i}\left(\left(\pi_{2_{k+1}^i}\right)^2 - \pi_{1_{k+1}^i}\right)}{\pi_{2_{k+1}^i}\left(\left(\pi_{2_k^i}\right)^2 - \pi_{1_{k+1}^i}\right)}, \quad \Pi_{2_k^i} = \frac{\pi_{2_{k+1}^i}\left(\left(\pi_{2_k^i}\right)^2 - \pi_{1_k^i}\right)}{\pi_{2_k^i}\omega_{k+1}^i\left(\pi_{2_{k+1}^i} + \omega_{k+1}^i - 2\right)},$$

and $\pi_{1_k}^i = \sum_{j=1}^{k}\left(\omega_j^i(z_j)\right)^2$, $\pi_{2_k}^i = \sum_{j=1}^{k}\omega_j^i(z_j)$, as described in Reference 40.

3.2 | EEFIG-ARX learning

For the online identification, it is employed an ARX model to obtain the parameters for (2). The foundation to be noted here is related to how the parameters are computed and updated considering the aspects of the evolving granular modeling.

Let $\psi_k = (Y_k, U_k) \in \mathbb{R}^{n\tau + m\delta}$ be the regressors vector, where $Y_k = (y_{k-1}, \ldots, y_{k-\tau})$ and $U_k = (u_{k-1}, \ldots, u_{k-\delta})$. Regarding the T-S rule base consequent structure, the autoregressive model identification is performed considering the ARX as structure. This formulation is expressed for the ith EEFIG as

$$\mathcal{A}_k^{i,j}\left(q^{-1}, k\right) y_k = \mathcal{B}_k^{i,j}\left(q^{-1}, k\right) u_k, \quad \forall j \in \mathbb{N}_{\leq n}, \tag{23}$$

where q^{-1} is the standard time shift operator, and $\mathcal{A}_k^{i,j}\left(q^{-1}, k\right)$, $\mathcal{B}_k^{i,j}\left(q^{-1}, k\right)$ are the following polynomials:

$$\mathcal{A}_k^{i,j}\left(q^{-1}, k\right) = 1 - a_{k,1}^{i,j}q^{-1} - \cdots - a_{k,\tau}^{i,j}q^{-\tau}, \tag{24}$$

$$\mathcal{B}_k^{i,j}\left(q^{-1}, k\right) = b_{k,1}^{i,j}q^{-1} + \cdots + b_{k,\delta}^{i,j}q^{-\delta}, \tag{25}$$

such that $a_{k,i_1}^{i,j} \in \mathbb{R}$, $i_1 \in \mathbb{N}_{\leq \tau}$ and $b_{k,i_2}^{i,j} \in \mathbb{R}$, $i_2 \in \mathbb{N}_{\leq \delta}$ are the output and input estimated parameters, respectively.

3.3 | Consequent update with recursive least squares with variable-direction forgetting

When a new granule is created, it is needed to compute its parameters which are initialized considering a window $\Psi_k = [\psi_k^\top, \dots, \psi_{k-n_w+1}^\top]^\top$ containing n_w with samples before the creation instant. The parameter estimation is performed using recursive least squares with variable-direction forgetting (RLS-VDF).[34] This method is applied to avoid bursting of least square parameters in absence of persistent excitation in data processing.

The RLS-VDF algorithm is described in Reference 34 and it can be applied to each consequent of the granule $\mathcal{G}_k^i$ by using the following equations

$$\overline{S}_k^{i,j} = \Lambda_k^{i,j^{-1}} S_k^{i,j} \Lambda_k^{i,j^{-1}}, \tag{26a}$$

$$\Lambda_k^{i,j} = U_k^{i,j} \overline{\Lambda}_k U_k^{i,j^\top}, \ \overline{\Lambda}_k = \left[\overline{\lambda}_k(k_1, k_2)\right]_{(n\tau+m\delta)\times(n\tau+m\delta)}, \ \overline{\lambda}_k(k_1, k_2) \triangleq \begin{cases} \sqrt{\lambda}, \text{ if } k_1 = k_2 \text{ and } \|\mathrm{col}_j(\Psi_k)\| > \upsilon, \\ 1, \text{ if } k_1 = k_2 \text{ and } \|\mathrm{col}_j(\Psi_k)\| \leq \upsilon, \\ 0, \text{ otherwise}, \end{cases} \tag{26b}$$

$$S_{k+1}^{i,j} = \overline{S}_k^{i,j} - \overline{S}_k^{i,j} \Psi_k \left(I + \Psi_k^\top \overline{S}_k^{i,j} \Psi_k\right)^{-1} \Psi_k^\top \overline{S}_k^{i,j}, \tag{26c}$$

$$T_{k+1}^{i,j} = T_k^{i,j} + S_{k+1}^{i,j} \Psi_k \left(y_k - \Psi_k^\top T_k^{i,j}\right), \tag{26d}$$

where $S_k^{i,j}$, $\Lambda_k^{i,j}$, $U_k^{i,j}$, $I \in \mathbb{R}^{(n\tau+m\delta)\times(n\tau+m\delta)}$ are, respectively, the parameter covariance matrix, the data-dependent forgetting factor matrix, orthonormal matrix which columns are the singular vectors of $S_{k-1}^{i,j^{-1}}$ and, identity matrix. $\lambda \in (0, 1]$ is the forgetting factor, $\upsilon \in \mathbb{R}$ is a very small positive value, and $T_k^{i,j} \in \mathbb{R}^{n\tau+m\delta}$ is the estimated parameters matrix. In the RLS-VDF algorithm, Λ_k is chosen to apply the forgetting factor λ only in the most relevant information directions. Finally, in terms of the output and input estimated parameters, $T_k^{i,j}$ is posed as follows

$$T_k^{i,j} = \begin{bmatrix} -a_{1,k}^{i,j} & \cdots & -a_{\tau,k}^{i,j} & b_{1,k}^{i,j} & \cdots & b_{\delta,k}^{i,j} \end{bmatrix}, \quad \forall j \in \mathbb{N}_{\leq n}. \tag{27}$$

However, the formulation (26a)–(26d) does not consider the normalized membership degrees of each granule during the model update. In Reference 41, it is proposed the use of weighted recursive least squares (WRLS) for updating evolving fuzzy models by considering the normalized membership degrees. In this article, the normalized membership degree of each rule is assigned as the weight of the WRLS. In this article, the same approach is proposed to improve the RLS-VDF for fuzzy models update. This new version of the estimator is denominated fuzzy weighted least squares with variable-direction forgetting (FWRLS-VDF). In this method, granules' parameters on the rule consequent are weighted by the normalized membership degree g_k^i. In this sense, (26c) and (26d) are rewritten as follows

$$S_{k+1}^{i,j} = \overline{S}_k^{i,j} - \overline{S}_k^{i,j} \Psi_k \left(I + g_k^i \Psi_k^\top \overline{S}_k^{i,j} \Psi_k\right)^{-1} \Psi_k^\top \overline{S}_k^{i,j}, \tag{28a}$$

$$T_{k+1}^{i,j} = T_k^{i,j} + g_k^i \overline{S}_{k+1}^{i,j} \Psi_k \left(y_k - \Psi_k^\top T_k^{i,j}\right). \tag{28b}$$

However, it is still necessary to revert the order of (28a) and (28b) to make the parameters update $T_{k+1}^{i,j}$ independent of the covariance update $S_{k+1}^{i,j}$. For this purpose, the following lemma is borrowed from Reference 34.

Lemma 1. *Let $X \in \mathbb{R}^{n\times n}$ be a positive semidefinite matrix, $Z \in \mathbb{R}^{n\times n}$ be positive definite and $Y \in \mathbb{R}^{n\times m}$. Then,*

$$\left[I - XY^\top \left(Z + YXY^\top\right)^{-1} Y\right] XY^\top = XY^\top \left(Z + YXY^\top\right)^{-1} Z. \tag{29}$$

Proof. The proof available in Reference 34. ■

Substituting (28a) in (28b), it follows that:

$$T^{i,j}_{k+1} = T^{i,j}_k + \left[I - g^i_k \overline{S}^{i,j}_k \Psi_k \left(I + \Psi_k^\top g^i_k \overline{S}^{i,j}_k \Psi_k\right)^{-1} \Psi_k^\top\right] \left(y_k - \Psi_k^\top T^{i,j}_k\right). \tag{30}$$

Using Lemma 1, the following FWRLS-VDF equations are obtained:

$$T^{i,j}_{k+1} = T^{i,j}_k + g^i_k \overline{S}^{i,j}_k \Psi_k \left(I + g^i_k \Psi_k^\top \overline{S}^{i,j}_k \Psi_k\right)^{-1} \left(y_k - \Psi_k^\top T^{i,j}_k\right), \tag{31a}$$

$$S^{i,j}_{k+1} = \overline{S}^{i,j}_k - \overline{S}^{i,j}_k \Psi_k \left(I + \Psi_k^\top \overline{S}^{i,j}_k \Psi_k\right)^{-1} \Psi_k^\top \overline{S}^{i,j}_k, \tag{31b}$$

where $\overline{S}^{i,j}_k$ and $\Lambda^{i,j}_k$ are defined in (26a) and (26b).

3.4 | State-space representation of EEFIG-ARX

From (23)–(25), the following extended state-space T-S fuzzy model is obtained:

$$\begin{aligned} \textbf{Rule } i: \quad & \text{IF } z_k \text{ is } \mathcal{G}^i_k, \\ & \text{THEN } x_{k+1} = A^i_k x_k + B^i_k u_k, \\ & \qquad\quad\ \ y_k = C^i_k x_k, \end{aligned} \tag{32}$$

where $x_k = (y_k, \overline{Y}_k, y_{k-\tau+1}, \bar{U}_k) \in \mathbb{R}^{n_x}$, with $n_x = n\tau + m(\delta - 1)$, $\overline{Y}_k = (y_{k-1}, \dots, y_{k-\tau+2})$, $\bar{U}_k = (u_{k-1}, \dots, u_{k-\delta+1})$, and

$$A^i_k = \begin{bmatrix} a^i_{k,1} & \alpha^i_k & a^i_{k,\tau} & \beta^i_k \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix}, \; B^i_k = \begin{bmatrix} b^i_{k,1} \\ 0 \\ 0 \\ I \\ 0 \end{bmatrix}, \; \alpha^i_k = \begin{bmatrix} a^i_{k,2} & \cdots & a^i_{k,\tau-1} \end{bmatrix}, \; \beta^i_k = \begin{bmatrix} b^i_{k,2} & \cdots & b^i_{k,\delta} \end{bmatrix}, \; C^i_k = \begin{bmatrix} I & 0 & 0 & 0 \end{bmatrix}. \tag{33}$$

Using the center-of-gravity defuzzification, the T-S fuzzy model can be represented in the following form:

$$\begin{aligned} x_{k+1} &= A_k(z_k)x_k + B_k(z_k)u_k, \\ y_k &= C_k(z_k)x_k, \end{aligned} \tag{34}$$

where $[A_k(z_k)\ B_k(z_k)\ C_k(z_k)] = \sum_{i=1}^N g^i_k(z_k) \begin{bmatrix} A^i_k & B^i_k & C^i_k \end{bmatrix}$. Notice that the normalized membership functions satisfy, by definition, the convexity property, that is:

$$\sum_{i=1}^N g^i_k(z_k) = 1, \quad 0 \leq g^i_k(z_k) \leq 1, \quad \forall k \in \mathbb{N}, i \in \mathbb{N}_{\leq N}. \tag{35}$$

Finally, considering the proposed learning triggering mechanism (5), the model used between two consecutive learning activation (cf. (6)), is described as follows:

$$\begin{aligned} \textbf{Rule } i: \quad & \text{IF } z_k \text{ is } \tilde{\mathcal{G}}^i_k, \\ & \text{THEN } x^i_{k+1} = \tilde{A}^i_k x_k + \tilde{B}^i_k u_k, \end{aligned} \tag{36}$$

where $\tilde{A}_k^i = A_{n_j}^i, \tilde{B}_k^i = B_{n_j}^i, \tilde{\mathcal{H}}_k^i = \mathcal{H}_{n_j}^i, \tilde{\mathcal{G}}_k^i = \mathcal{G}_{n_j}^i, \forall k \in \left\{n_j, \dots, n_{j+1} - 1\right\}$. Then, the inferred fuzzy model is

$$x_{k+1} = \tilde{A}_k(z_k)x_k + \tilde{B}_k(z_k)u_k, \tag{37}$$

where $\left[\tilde{A}_k(z_k)\ \tilde{B}_k(z_k)\right] = \sum_{i=1}^N \tilde{g}_k^i(z_k)\left[\tilde{A}_k^i\ \tilde{B}_k^i\right]$ and $\tilde{g}_k^i = g_{n_j}^i, \forall k \in \left\{n_j, \dots, n_{j+1} - 1\right\}$ which also meet the convexity property.

The algorithmic realization of the proposed EEFIG-ARX for the evolving T-S fuzzy modeling is summarized in Algorithm 1.

Algorithm 1. EEFIG-ARX

```
Input:   $z_k$ (sample from the data stream), $\mathbb{G}_{k-1}$ (information granules)
Output:  $\mathbb{G}_k, A_k^i, B_k^i, \quad \forall i \in \mathbb{N}_{\leq N}$ (estimated parameters).
for $i \leftarrow 1$ to $N$ do
    compute $g_k^i(z_k)$ using (3) and (14);
    if $L_1 = TRUE$ and $L_2 = TRUE$ then
        update $\mathcal{H}_k^i$ using (22a), (22b), (22c) and (22d);
    end
    if $c_s = TRUE$ then
        create a new granule;
        $N \leftarrow N + 1$;                               // updating the number of granules
        compute $\Psi_k$ for a window $\mathcal{W}$;                  // regressors matrix
        compute $S_k^N, T_k^N$ from (26c) and (26d);  // initial parameters for the new EEFIG using
                                                                        RLS-VDF
        build $A_k^N$ and $B_k^N$ from $T_k^N$ using (27) and (33);
    end
    compute $g_k^i(z_k)$ considering the update on the number of granules;
    compute $\Psi_k$ for a window $\mathcal{W}$;
    compute $T_k^i, S_k^i$ using (31a)–(31b);                        // using WRLS-VDF
    build $A_k^i$ and $B_k^i$ from $T_k^i$ using (27) and (33);
end
```

4 | ETC BASED ON EVOLVING FUZZY MODELS

In this section, all developments are performed for a fixed time step $k \in \{k_j, \dots, k_{j+1} - 1\}$. Therefore, for simplicity, the time sample is omitted here. The updating rule of the controller and the triggering parameters are discussed in detail in Section 4.2. Consider the following evolving fuzzy control law:

$$u_k = K(\tilde{z}_k)\tilde{x}_k = \sum_{j=1}^N \tilde{g}^j(\tilde{z}_k)K^j\tilde{x}_k, \quad \forall k \in \{k_j, \dots, k_{j+1} - 1\}, \tag{38}$$

where $\tilde{x}_k$ is defined in (8) and $K_k^j \in \mathbb{R}^{m \times n_x}, j \in \mathbb{N}_{\leq N}$ are the local control gains. Based on the transmission error defined in (11), the closed-loop system (37) with the control law (38) can be written as follows:

$$x_{k+1} = \left(\tilde{A}(z_k) + \tilde{B}(z_k)K(\tilde{z}_k)\right)x_k + \tilde{B}(z_k)K(\tilde{z}_k)e_k. \tag{39}$$

It is clear that there is a mismatching between the controller's and plant's premise variables induced by the event-based sampling. This phenomenon is one of the sources of conservativeness in ETC design for T-S fuzzy models. To cope with this phenomenon, an appropriate triggering strategy is proposed as follows:

$$k_0 = 0,\ k_{j+1} = \min\{k \in \mathbb{N} : k > k_j \wedge \Gamma(x_k, e_k) \leq 0\}, \quad \forall j \in \mathbb{N}, \tag{40}$$

with the following trigger function:

$$\Gamma(x_k, e_k) := x_k^\top \Theta x_k - e_k^\top \Xi e_k - \zeta(x_k, e_k), \tag{41}$$

where

$$\zeta(x_k, e_k) := 2\zeta_1^\top(x_k, e_k) P \zeta_2(x_k, e_k) + \zeta_2^\top(x_k, e_k) P \zeta_2(x_k, e_k), \tag{42}$$

$$\zeta_1(x_k, e_k) := \left(\tilde{A}(z_k) + B(z_k)K(z_k)\right) x_k + \tilde{B}(z_k)K(z_k)e_k, \tag{43}$$

$$\zeta_2(x_k, e_k) := \tilde{B}(z_k)\left(K(\tilde{z}_k) - K(z_k)\right)(x_k + e_k), \tag{44}$$

with $P, \Theta, \Xi \in \mathbb{R}^{n_x \times n_x}$ being symmetric positive definite matrices.

4.1 | LMI-based ETC design

The design condition provided in this article is based on the following Lyapunov function candidate $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ for the system (39) equipped with the ETM in (40):

$$V(x_k) = x_k^\top P x_k, \tag{45}$$

where $P \in \mathbb{R}^{n_x \times n_x}$ is a symmetric positive definite matrix. Based on the Lyapunov function candidate $V(x_k)$ in (45) and the ETM with cancellation structure of asynchronous parameters in (41), an LMI-based co-design condition is stated as follows.

Theorem 1. *For a given $x_k \in \mathbb{R}^{n_x}$, if there exist symmetric positive definite matrices $\tilde{Q}, \tilde{\Theta}, \tilde{\Xi}, \tilde{Q}_0 \in \mathbb{R}^{n_x \times n_x}$ and matrices $\tilde{X} \in \mathbb{R}^{n_x \times n_x}$ and $\tilde{K}^j \in \mathbb{R}^{m \times n_x}$, $j \in \mathbb{N}_{\leq r}$, such that the following optimization problem is feasible:*

$$\min_{\tilde{Q}, \tilde{X}, \tilde{\Xi}, \tilde{\Psi}, \tilde{K}^j} \quad \operatorname{tr}(\tilde{\Xi} + \tilde{\Psi} + \tilde{Q}_0), \tag{46}$$

$$\text{subject to} \quad \begin{cases} \Upsilon^{ii} < 0, \\ \Upsilon^{ij} + \Upsilon^{ji} < 0, \quad i < j, \end{cases} \qquad \forall\ i,j \in \mathbb{N}_{\leq N}, \tag{47}$$

$$\begin{bmatrix} -\tilde{Q} & I \\ I & -\tilde{Q}_0 \end{bmatrix} < 0, \tag{48}$$

$$\begin{bmatrix} 1 & x_k^\top \\ x_k & \tilde{Q} \end{bmatrix} \geq 0, \tag{49}$$

where

$$\Upsilon^{ij} = \begin{bmatrix} -\tilde{X} - \tilde{X}^\top + \tilde{Q} & \star & \star & \star \\ 0 & -\tilde{\Xi} & \star & \star \\ \tilde{A}^i \tilde{X} + B^i \tilde{K}^j & \tilde{B}^i \tilde{K}^j & -\tilde{Q} & \star \\ \tilde{X} & 0 & 0 & -\tilde{\Theta} \end{bmatrix},$$

then the origin of the closed-loop system (39) equipped with the ETM (40) is asymptotically stable with control gains $K^j = \tilde{K}^j \tilde{X}^{-1}$, $j \in \mathbb{N}_{\leq N}$, triggering matrices $P = \tilde{Q}^{-1}$, $\Theta = \tilde{\Theta}^{-1}$, $\Xi = \tilde{X}^{-\top} \tilde{\Xi} \tilde{X}^{-1}$, and $V(x_k)$ in (45) is a Lyapunov function.

Proof. Assume the optimization problem (46), subject to the constraints (47)–(49), is feasible. Since

$$\Upsilon(z_k) = \sum_{i=1}^{N}\sum_{j=1}^{N}\tilde{g}^i(z_k)\tilde{g}^j(z_k)\Upsilon^{ij} = \sum_{i=1}^{N}\left(\tilde{g}^i(z_k)\right)^2\Upsilon^{ii} + \sum_{i=1}^{N}\sum_{j=i+1}^{N}\tilde{g}^i(z_k)\tilde{g}^j(z_k)\left(\Upsilon^{ij}+\Upsilon^{ji}\right) \tag{50}$$

from convexity property of the normalized membership functions in (35), inequalities in (47) implies that

$$\Upsilon(z_k) = \begin{bmatrix} -\tilde{X}-\tilde{X}^{\top}+\tilde{Q} & \star & \star & \star \\ 0 & -\tilde{\Xi} & \star & \star \\ \tilde{A}(z_k)\tilde{X}+\tilde{B}(z_k)\tilde{K}(z_k) & \tilde{B}(z_k)\tilde{K}(z_k) & -\tilde{Q} & \star \\ \tilde{X} & 0 & 0 & -\tilde{\Theta} \end{bmatrix} < 0. \tag{51}$$

Then define $K^j = \tilde{K}^j\tilde{X}^{-1}, j \in \mathbb{N}_{\leq N}$, and substitute $K(z_k) = \tilde{K}(z_k)\tilde{X}^{-1}$ into (51) to obtain

$$\begin{bmatrix} -\tilde{X}-\tilde{X}^{\top}+\tilde{Q} & \star & \star & \star \\ 0 & -\tilde{\Xi} & \star & \star \\ \tilde{A}(z_k)\tilde{X}+\tilde{B}(z_k)K(z_k)\tilde{X} & \tilde{B}(z_k)K(z_k)\tilde{X} & -\tilde{Q} & \star \\ \tilde{X} & 0 & 0 & -\tilde{\Theta} \end{bmatrix} < 0. \tag{52}$$

Also, as $\tilde{X}+\tilde{X}^{\top} > \tilde{Q} > 0$ then $\tilde{X}$ is nonsingular and $-\tilde{X}^{\top}\tilde{Q}^{-1}\tilde{X} \leq -\tilde{X}-\tilde{X}^{\top}+\tilde{Q}$, then it follows that

$$\begin{bmatrix} -\tilde{X}^{\top}\tilde{Q}^{-1}\tilde{X} & \star & \star & \star \\ 0 & -\tilde{\Xi} & \star & \star \\ \tilde{A}(z_k)\tilde{X}+\tilde{B}(z_k)K(z_k)\tilde{X} & \tilde{B}(z_k)K(z_k)\tilde{X} & -\tilde{Q} & \star \\ \tilde{X} & 0 & 0 & -\tilde{\Theta} \end{bmatrix} < 0. \tag{53}$$

By pre-multiplying inequality (53) by diag $\left(\tilde{X}^{-\top},\tilde{X}^{-\top},\tilde{Q}^{-1},I\right)$ and post-multiplying it by diag $\left(\tilde{X}^{-1},\tilde{X}^{-1},\tilde{Q}^{-1},I\right)$, it yields

$$\begin{bmatrix} -\tilde{Q}^{-1} & \star & \star & \star \\ 0 & -\Xi & \star & \star \\ \tilde{Q}^{-1}\left(\tilde{A}(z_k)+\tilde{B}(z_k)K(z_k)\right) & \tilde{Q}^{-1}\tilde{B}(z_k)K(z_k) & -\tilde{Q}^{-1} & \star \\ I & 0 & 0 & -\tilde{\Theta} \end{bmatrix} < 0, \tag{54}$$

where $\Xi = \tilde{X}^{-\top}\tilde{\Xi}\tilde{X}^{-1}$. Since $\tilde{Q}$ and $\tilde{\Theta}$ are nonsingular matrices, define $P = \tilde{Q}^{-1}$ and $\Theta = \tilde{\Theta}^{-1}$ to rewrite (54) as

$$\begin{bmatrix} -P & \star & \star & \star \\ 0 & -\Xi & \star & \star \\ P\left(\tilde{A}(z_k)+\tilde{B}(z_k)K(z_k)\right) & P\tilde{B}(z_k)K(z_k) & -P & \star \\ I & 0 & 0 & -\Theta^{-1} \end{bmatrix} < 0. \tag{55}$$

From Schur complement lemma, it implies

$$\begin{bmatrix} -P+\Theta+A_{cl}^{\top}(z_k)PA_{cl}(z_k) & \star \\ B_{cl}^{\top}(z_k)PA_{cl}(z_k) & -\Xi+B_{cl}^{\top}(z_k)PB_{cl}(z_k) \end{bmatrix} < 0, \tag{56}$$

where $A_{cl}(z_k) := \tilde{A}(z_k) + \tilde{B}(z_k)K(z_k)$, $B_{cl}(z_k) := \tilde{B}(z_k)K(z_k)$. Now, pre-multiplying (56) by $\begin{bmatrix} x_k^\top & e_k^\top \end{bmatrix}$ and post-multiplying it by $\begin{bmatrix} x_k^\top & e_k^\top \end{bmatrix}^\top$, results

$$(A_{cl}(z_k)x_k + B_{cl}(z_k)e_k)^\top P\,(A_{cl}(z_k)x_k + B_{cl}(z_k)e_k) - x_k^\top P x_k + x_k^{\top\Theta} x_k - e_k^{\top\Xi} e_k + \zeta(x_k, e_k) - \zeta(x_k, e_k) < 0, \tag{57}$$

which ensures that

$$V(x_{k+1}) - V(x_k) + \Gamma(x_k, e_k) < 0, \tag{58}$$

where $\Gamma(x_k, e_k)$ is defined in (41) and $V(x_k)$ in (45). Since the triggering mechanism in (40) enforces that $\Gamma(x_k, e_k) > 0$, then it follows from (58) that $V(x_{k+1}) - V(x_k) < 0$, $\forall k \in \{k_j, \dots, k_{j+1} - 1\}$. Therefore, the origin of the closed-loop system (39) under the ETM (40) is asymptotically stable. From Schur complement lemma, constraint (48) is equivalent to $\tilde{Q}^{-1} < \tilde{Q}_0$. Then the minimization of $\mathrm{tr}(\tilde{Q}_0)$ tends to minimize the eigenvalues of $\tilde{Q}^{-1} = P$. From Schur complement lemma, constraint (49) implies $x_k^\top P x_k \leq 1$, which means that the unitary level set of $V(x_k)$ is maximized. As discussed in Reference 11, the minimization of $\mathrm{tr}(\tilde{\Xi} + \tilde{\Psi})$ tends to induce larger inter-event times and, consequently, to reduce the number of events. This concludes the proof. ■

4.2 | Overview of the learning-based ETC algorithm

Theorem 1 provides a sufficient condition for the stabilization of the T-S fuzzy model by means of an ETC law. However, when the controller gains and the ETM are designed through Theorem 1 for a given realization of the EEFIG model $(A_k(z_k), B_k(z_k))$, the stability is not guaranteed if the next realization $(A_{k+1}(z_{k+1}), B_{k+1}(z_{k+1}))$ diverges from the previous one. To tackle this issue, it is proposed an algorithm that redesigns the matrices K, Ξ, and Θ, when the system parameters evolve. Indeed if the learning is not triggered at instant $k+1$, the same K, Ξ, and Θ can be used. In this sense, for the sake of consistency, those matrices will be denoted as time-varying in this section, namely K_k, Ξ_k, and Θ_k.

The learning trigger mechanism described in (5) ensures that the model defined by $\tilde{A}_k$ and $\tilde{B}_k$ holds until the next learning trigger event. During this time, the ETC solution is maintained, that is, $K_k = K_{k-1}$, $\Xi_k = \Xi_{k-1}$, and $\Theta_k = \Theta_{k-1}$. When the learning trigger is activated, $\tilde{A}_k$ and $\tilde{B}_k$ are updated and the matrices K_k, Ξ_k, and Θ_k are computed again by solving (46) with the matrix $\tilde{Q}_{k-1}$ obtained in the last iteration. Since the solution of the optimization problem (46)–(49) ensures, from Theorem 1, that $V(x_{k+1}) < V(x_k) \leq 1$ with $P = \tilde{Q}_{k-1}^{-1}$, the idea of using the same matrix $\tilde{Q}_{k-1}$ is to ensure that this inclusion condition is fulfilled at the next iteration. However, this may lead to a more conservative solution and the feasibility is ensured in general for sufficiently smooth variations on the model parameters.

When the fuzzy model is updated to describe the nonlinear system behavior and the learning trigger mechanism transmits new model parameters to the event-triggered fuzzy controller design, the optimization problem solved with the new model parameters may become infeasible. In this case, the optimization problem (46) is solved again with $\tilde{Q}_k$ as the decision variable and a new $\tilde{Q}_k$ is obtained. Then the process is repeated in order to ensure the inclusion for the new computed Lyapunov function. The learning-based ETC algorithm that is employed to update $\tilde{Q}_k$, K_k, Ξ_k, and Θ_k is summarized in Algorithm 2.

4.3 | Initialization and issues of validity

4.3.1 | Kickoff in the evolving granular model

The initialization of the proposed evolving modeling approach (EEFIG-ARX) relies on the existence of some input–output data from the system. Those data are used to create the first granule, compute its prototype, and estimate its parameters by using the RLS-VDF. The minimum number of samples necessary to initialize the EEFIG-ARX is $n_w + 1$. It is not required stable operation of the system when this data is acquired, that is, it can be obtained in open-loop even for an unstable system. The Learning-based ETC algorithm (Algorithm 2) should be executed based on a granular base $\mathbb{G}_k$ that contains at least one granule, but it may be initialized with more granules if there is enough input–output data. In all the experiments presented in this article, it is used only the minimum data sample to initialize the EEFIG-ARX, and therefore, there is a single granule at the beginning of the simulations.

Algorithm 2. Learning-based ETC

```
Input:  $z_k, \mathbb{G}_{k-1}, \tilde{Q}_{k-1}, K_{k-1}, \Xi_{k-1}, \Theta_{k-1}, A^i_{k-1}, B^i_{k-1}, \quad \forall i \in \mathbb{N}_{\leq N}$
Output: $u_k, \tilde{Q}_k$.
update $\mathbb{G}_k, A_k(z_k)$, and $B_k(z_k)$ using Algorithm 1;
if $\|\epsilon_k\| \geq \sigma\|y_k\|$ then
    $\tilde{\mathbb{G}}_k \leftarrow \mathbb{G}_k, \tilde{A}_k(z_k) \leftarrow A_k(z_k)$, and $\tilde{B}_k(z_k) \leftarrow B_k(z_k)$;
    solve the problem (46)–(49) for a given $\tilde{Q}_{k-1}$;
    if (47)–(49) are INFEASIBLE for a given $\tilde{Q}_{k-1}$ then
        solve (46)–(49) with $\tilde{Q}_k$ as decision variable;
        if (46)–(49) are INFEASIBLE with $\tilde{Q}_k$ as decision variable then
            $\tilde{Q}_k \leftarrow \tilde{Q}_{k-1}, K^j_k \leftarrow K^j_{k-1}, \Xi_k \leftarrow \Xi_{k-1}$, and $\Theta_k \leftarrow \Theta_{k-1}$;
        else
            $K^j_k \leftarrow \tilde{K}^j\tilde{X}^{-1}, \Xi_k \leftarrow \tilde{X}^{-\top}\tilde{\Xi}\tilde{X}^{-1}$, and $\Theta_k \leftarrow \tilde{\Theta}^{-1}$;
        end
    else
        $\tilde{Q}_k \leftarrow \tilde{Q}_{k-1}, K^j_k \leftarrow \tilde{K}^j\tilde{X}^{-1}, \Xi_k \leftarrow \tilde{X}^{-\top}\tilde{\Xi}\tilde{X}^{-1}$, and $\Theta_k \leftarrow \tilde{\Theta}^{-1}$;
    end
else
    $\tilde{Q}_k \leftarrow \tilde{Q}_{k-1}, K^j_k \leftarrow K^j_{k-1}, \Xi_k \leftarrow \Xi_{k-1}$, and $\Theta_k \leftarrow \Theta_{k-1}$;
end
compute $\Gamma(x_k, e_k)$ using (41)–(44);
compute $\tilde{u} = K_k\,(z_k)\,x_k$;
if $\Gamma(x_k, e_k) \leq 0$ then
    $u_k \leftarrow \tilde{u}$;
else
    $u_k \leftarrow u_{k-1}$;
end
```

4.3.2 | Recursive feasibility and robustness

The problem of recursive feasibility is not totally addressed here, although condition (49) helps to maintain the system trajectories within the origin's domain of attraction. In general, it is enough to provide stabilization and recursive feasibility of the algorithm. However, there are situations that may lead to instability or infeasibility: (i) when the model updates too fast, it is possible that new granules with inaccurate initial parameters are created making the optimization problem infeasible or producing controllers based on immature models; (ii) disturbances may instantaneously lead the trajectories to a region of the state space that is too far from the already covered region and, therefore, out of the domain of attraction ensured by the current controller. Both situations may be avoided by using initialization data with enough spectral distribution. In addition, the parameters n_w and η may be also increased to make the model more resistant to novel information.

5 | SIMULATION EXAMPLES

The effectiveness of the proposed EEFIG-ARX learning and the evolving ETC is illustrated by two examples: the van der Pol oscillator and the magnetic levitation system. The parameters of the controller and ETM are computed by solving the optimization problem (46) subject to the LMI constraints (47)–(49) using the Yalmip parser and MOSEK solver in the MATLAB environment.

5.1 van der Pol oscillator

Consider the discrete van der Pol oscillator equations:[42]

$$\begin{bmatrix} y_{1,k+1} \\ y_{2,k+1} \end{bmatrix} = \begin{bmatrix} y_{1,k} + hy_{2,k} \\ -hy_{1,k} + y_{2,k} + \rho_k h(0.5 - y_{1,k}^2)y_{2,k} \end{bmatrix} + \begin{bmatrix} 0 \\ h \end{bmatrix} u_k, \tag{59}$$

with parameters ρ_k and $h = 0.01$ s. The time-varying parameter ρ_k is described as follows:

$$\rho_k = \begin{cases} 1, & 0 \leq k \leq 500, \\ 2, & 500 < k \leq 1000, \\ 3, & 1000 < k \leq 1500, \\ 1 + 1.1\cos(0.1hk), & k > 1500. \end{cases} \tag{60}$$

This nonlinear time-varying model is only considered to generate the data stream.

Based only on the generated data stream, and without any prior knowledge about system dynamics, the evolving fuzzy model and the event-triggered fuzzy controller are updated according to Algorithm 1. Two distinct experiments are performed to evaluate the effect of the forgetting factor and the window width on the granulation process and, consequently, in the control per se. Each experiment has been conducted considering the following parameters:

- Experiment 1: $\eta = 0.99$, $\sigma = 0.001$, $n_w = 5$.
- Experiment 2: $\eta = 0.975$, $\sigma = 0.001$, $n_w = 20$.

In all experiments, the selected model structure presents the same parameters $\tau = 1$ and $\delta = 2$, that is, the state vector is given by $x_k = (y_k, u_{k-1})$. Furthermore, the same threshold σ is considered for the learning trigger rule.

The difference between them is in the selected forgetting factor η and the window width n_w. The forgetting factor η is related to the weight given to more recent data, while n_w is related to the model granulation, such that larger values of n_w produce less variations in the evolving model. In addition, for the initialization of both experiments it has been considered a random sequence input data with $n_w + 1$ and the system's response when applying the input data. The same sequence has been used in both experiments. However, it has been used only the minimum amount of data for initializing the algorithm, that is, $n_w + 1$, and the initial output values y_0 are assumed to be the last output of the initialization data. For this reason, the initial outputs of experiments are different: in Experiment 1, it is $y_0 = [-0.8352\ 0.4332]^\top$; and in Experiment 2, it is $y_0 = [-0.0087\ 0.1332]^\top$.

The results of Experiment 1 are depicted in Figure 2. The data measured from the plant are shown in Figure 2A, the control signal generated under Algorithm 2 is depicted in Figure 2C, the transmission instants of the learning trigger and the learning ETC are depicted in Figure 2B and the prediction error is depicted in Figure 2D. In this case, three granules have been created during the execution and the instants at which new granules and, consequently, new fuzzy rules have been created are illustrated as dotted vertical lines. It can be noticed that the creation of new fuzzy rules coincides with the instants at which the prediction error increases, indicating that new granules are created when the current existing knowledge-base is not able to properly describe the system's behavior. It can also be noticed that the learning trigger avoids updating the controller structure at every time instants, the updating occurs only when the trigger rule is violated. Thus, the instants at which the controller structure is updated are illustrated in Figure 2B when the learning trigger signal is 1. From Figures 2A,B, it can be noticed that the control signal is updated only when new events of the ETM are transmitted, which implies saving network communication resources.

The results of Experiment 2 are depicted in Figure 3. The data measured from the plant are shown in Figure 3A, the control signal is depicted in Figure 3C, the transmission instants of the learning trigger and the

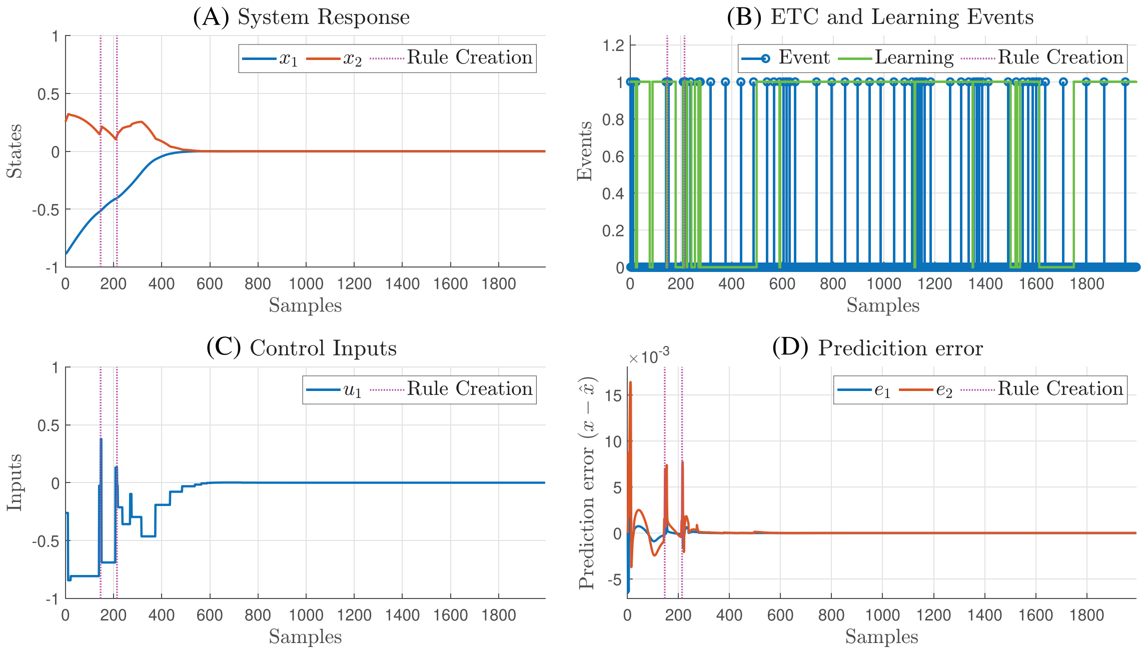

FIGURE 2 Simulation results of Experiment 1 for the van der Pol oscillator system

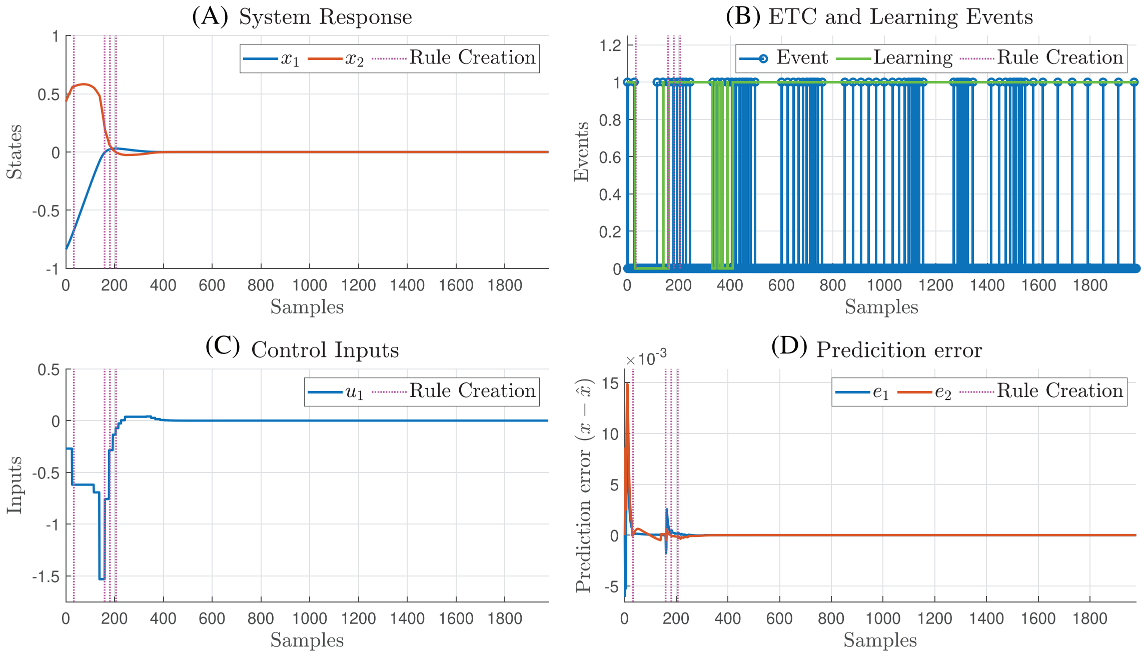

FIGURE 3 Simulation results of Experiment 2 for the van der Pol oscillator system

ETC are depicted in Figure 3B as well as the prediction error is depicted in Figure 3D. In this case, five granules have been created and, similarly to Experiment 1, the creation of new fuzzy rules coincides with the variations of the prediction error. In contrast to the first experiment, the smaller forgetting factor η prioritizes the use of recent data for the construction of the data-driven fuzzy model, which is related to the higher number of created granules. It is also evident the effect of the learning trigger in reducing the controller structure updating as well as the learning ETC scheme in reducing the amount of data transmitted through the network communication.

5.2 | Magnetic levitation

Consider the magnetic levitation system[43] depicted in Figure 4 and described by the following nonlinear dynamical equations:

$$\begin{bmatrix} y_{1,k+1} \\ y_{2,k+1} \end{bmatrix} = \begin{bmatrix} y_{1,k} + hy_{2,k} \\ f_1(y_k)y_{1,k} + \left(1 - \frac{hK_m}{m}\right) x_{2,k} + f_2(y_k)\left(u_k + d_k\right) \end{bmatrix}, \tag{61}$$

where $x_{1,k}$ is the ball's height around the reference system at $y^o = 0.05$ m, $x_{2,k}$ is the ball's vertical velocity,

$$f_1(y_k) = \frac{hg\mu\left(\mu y_{1,k} + 2\mu y^o + 2\right) y_{1,k}}{\left(1 + \mu\left(y_{1,k} + y^o\right)\right)^2}, \quad f_2(y_k) = \frac{hL_k\mu}{2m\left(1 + \mu\left(y_{1,k} + y^o\right)\right)^2}, \tag{62}$$

and the parameters are: the ball's mass $m = 0.068$ kg, gravitational acceleration $g = 9.8$ m/s^2, viscous friction coefficient $K_m = 0.001$ N$\,$(m/s)$^{-1}$, coefficient of inductance variation $\mu = 2$ m^{-1}, and sampling period $h = 0.01$ s. The time-varying inductance is

$$L_k = \begin{cases} 0.46, & 0 \leq k \leq 150, \\ 0.368, & 150 \leq k \leq 750, \\ 0.575, & 750 < k \leq 1400, \\ 0.46 + 0.23\sin(0.05hk), & k > 1400, \end{cases} \tag{63}$$

and the actuator disturbance d_k is

$$d_k = \begin{cases} 0, & 0 \leq k \leq 750, \\ -0.015, & k > 750. \end{cases} \tag{64}$$

It is worth to mention that (61) presents an open-loop unstable equilibrium point in the origin, and fast dynamics, which makes the stabilization based on data-driven models a challenging task. Three experiments have been performed using the structure with $\tau = 2$ and $\delta = 2$, as the maximum lags of output and input, respectively. The parameters used in each experiment are listed as follows:

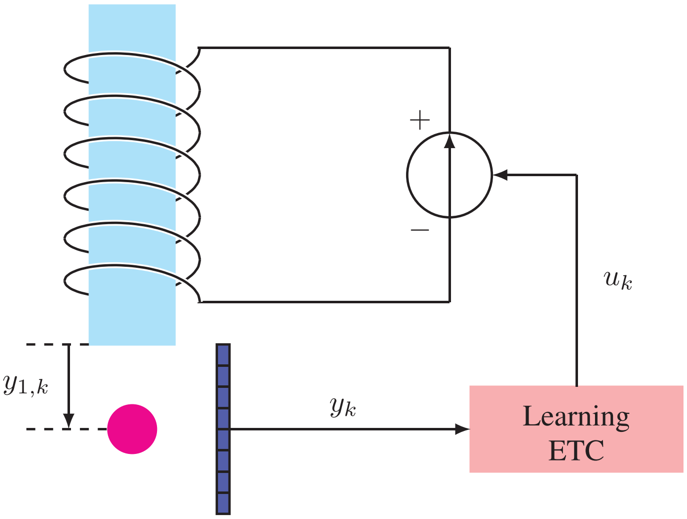

FIGURE 4 Diagram of the magnetic suspension system

- Experiment 1: $\eta = 0.99$, $\sigma = 0.001$, $n_w = 30$.
- Experiment 2: $\eta = 0.997$, $\sigma = 0.001$, $n_w = 25$.
- Experiment 3: $\eta = 0.997$, $\sigma = 0.00001$, $n_w = 25$.

The results of Experiments 1–3 are depicted, respectively, in Figures 5–7. As in the van der Pol experiment, for the initialization of the three experiments it has been considered a random sequence input data with $n_w + 1$

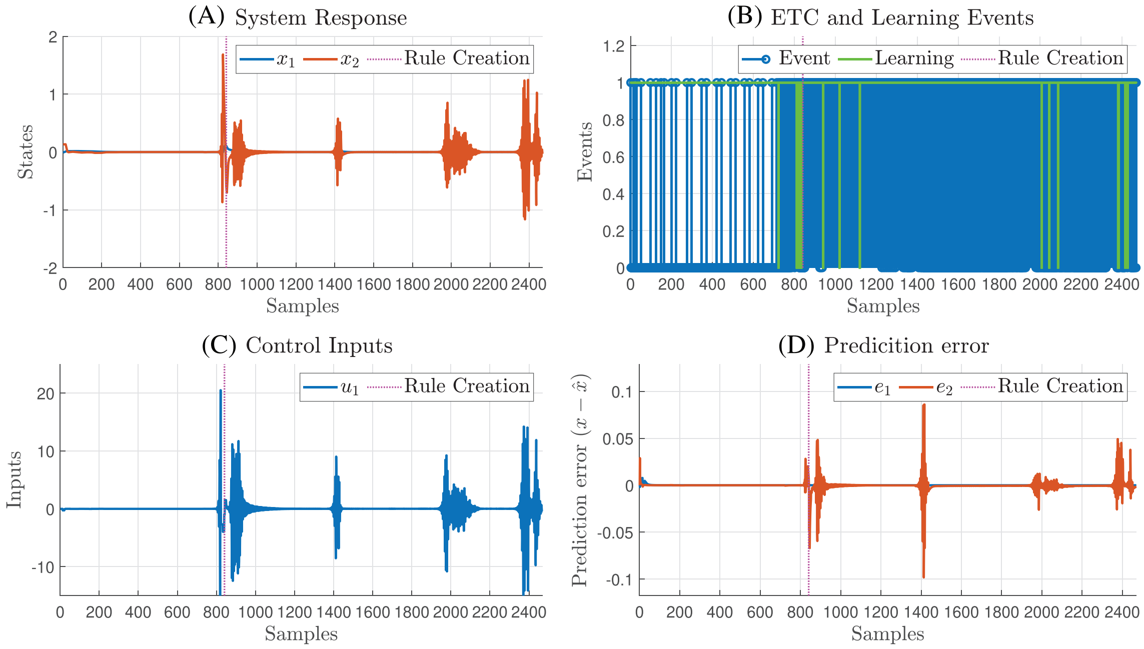

FIGURE 5 Simulation results of Experiment 1 for the magnetic levitation system

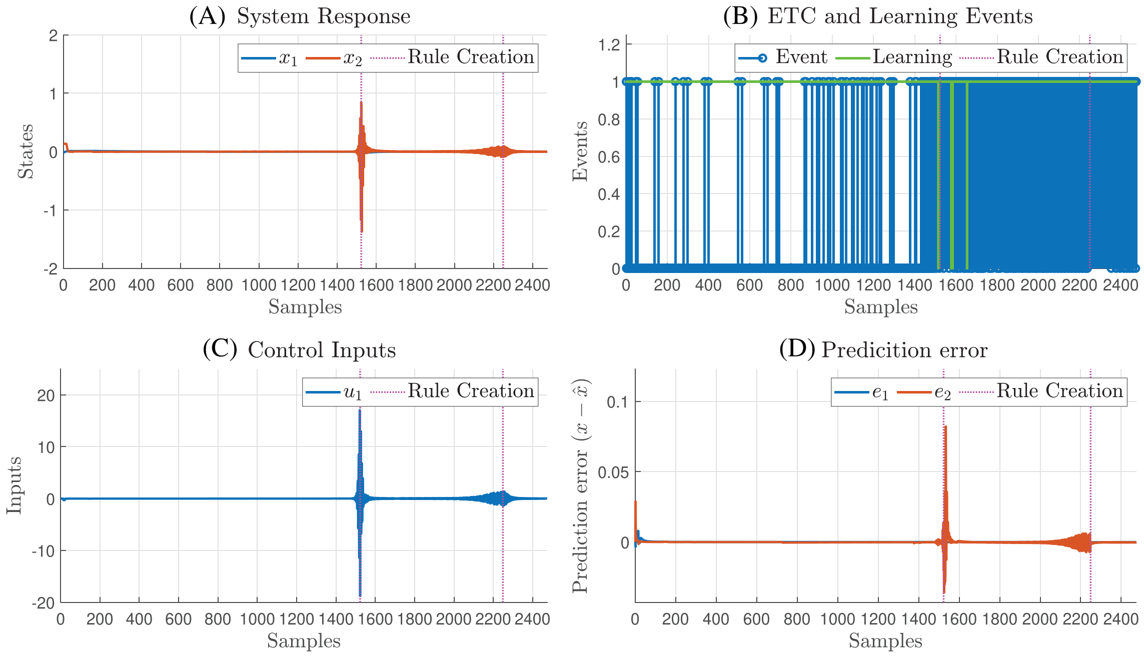

FIGURE 6 Simulation results of Experiment 2 for the magnetic levitation system

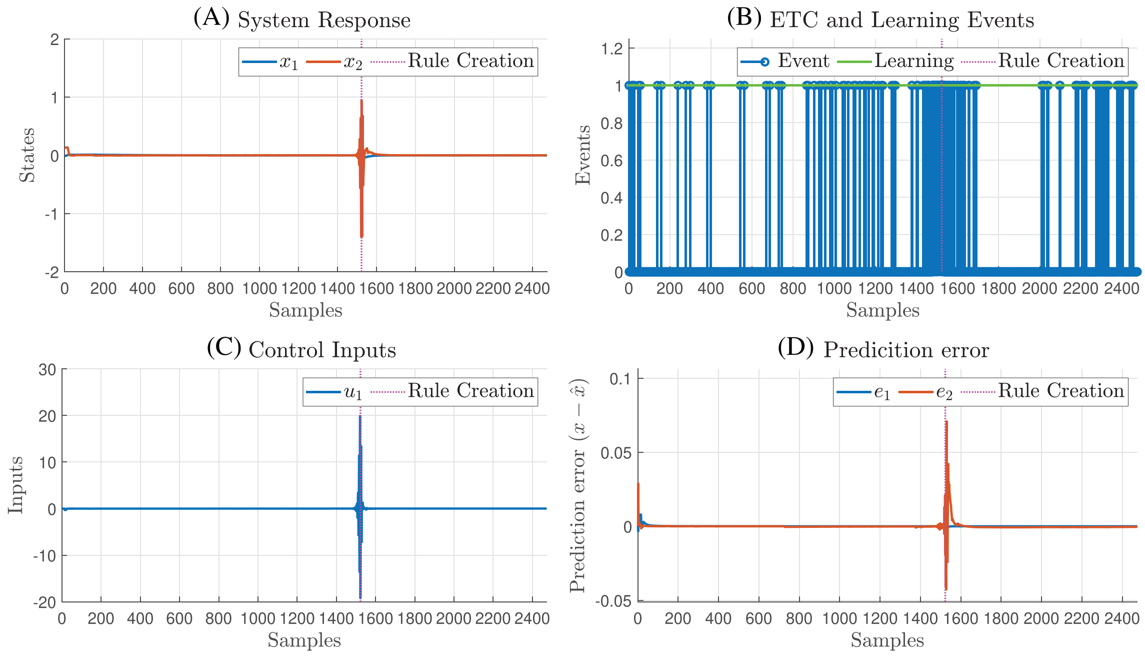

FIGURE 7 Simulation results of Experiment 3 for the magnetic levitation system

and the system's response when applying the input data. The same sequence has been used in all the experiments. However, it has been only used the minimum amount of data for initializing the algorithm, that is, $n_w + 1$, and the initial output values y_0 are assumed to be the last output of the initialization data. For this reason, the initial outputs of Experiment 1 is $y_0 = [-0.0087\ 0.1332]^\mathsf{T}$, and the initial outputs of Experiments 2 and 3 are $y_0 = [-0.0154\ 0.1331]^\mathsf{T}$.

The results in all experiments indicated that the proposed learning ETC algorithm is able to stabilize the magnetic levitation system even under variation of the inductance and presence of disturbances. However, it is clear that the unstable dynamics required more effort from the control and the model learning algorithm to ensure the stability. Furthermore, since the system was very close to the origin in $k = 0$ and the stabilization is reached rapidly, the model was not able to learn the dynamics outside of the neighborhood of the origin. For this reason, in absence of disturbances, a linear model was enough to stabilize the system. In addition, after the injection of the disturbance and the variation of the inductance value, it was necessary to create new granules to better represent the dynamics.

In Experiment 1, the injection of the additive actuator disturbance d_k at $k = 750$ increases prediction error and, consequently, the system's output and the control input increase as well. As the system evolves, a new granule has been created and the effects of the disturbance have been attenuated owe to the granule maturation. The system's trajectories and the prediction error increased again after the $k = 1400$ due to the sinusoidal behavior of the inductance. The system reached the equilibrium again after some samples due to the learning ability of the model, but new disturbances due to sinusoidal parameter variation appear periodically. However, it is clear that the system with time-varying dynamics and disturbances required more learning and control events to ensure the system stability.

In Experiment 2, a similar behavior was observed and the number of events was increased after $k = 750$. However, with a higher forgetting factor $\eta = 0.997$, it was less sensitive to the first disturbance injection at $k = 750$, although the injection of the sinusoidal inductance variation at $k = 1400$ increased the prediction error and displaced the ball, leading to the creation of a new granule to describe the dynamics far away from the equilibrium point. The system trajectories and the prediction error are disturbed again at $k = 1800$ when the sinusoidal inductance is approaching to its peak, which increases the prediction error. The stabilization was ensured again after the creation of a third granule. The comparison between Experiments 2 and 3 allows to understand the effect of the learning threshold σ. In Experiment 3, the use of a lower learning threshold σ increases the number learning events. In this case, the model learning have been active during the whole simulation time. As a consequence, the system's states were less affected by the disturbances, less control events were necessary, and only two granules were used to represent the system's dynamics.

However, the computational cost of designing the control parameters at each learning event is greater than in the other experiments.

6 | CONCLUSION

This article presented a novel learning-based ETC strategy that relies on the data-driven evolving fuzzy modeling EEFIG-ARX. The proposed learning-based control approach addresses the networked control of nonlinear and time-varying systems with unknown dynamics. The EEFIG-ARX is based on the principle of justifiable granularity and uses a parameter estimation algorithm (FWRLS-VD) that eases the use of this kind of model for online learning control strategies, since it is able to deal with the loss of persistence of excitation. The ETC gains and the ETM parameters are designed through a convex optimization procedure based on LMI constraints which reduces the consumption of communication resources. That optimization problem is solved every time the model is updated. In order to save computational efforts, the model learning is controlled by means of an event-triggering learning mechanism that avoids control redesigns while the model accuracy is admissible. Numerical simulations of nonlinear systems with time-varying dynamics and disturbances are used to illustrate the effectiveness of the proposed learning-based ETC strategy.

Although the proposed approach seems to be effective to control even unstable and time-varying nonlinear systems, there are still issues regarding the recursive feasibility that should be investigated in future works. In addition, further investigations should also include mechanisms to ensure the control input boundness in order to avoid instability due to the model adaptation.

ACKNOWLEDGMENTS

This work was supported by CNPq, Brazil, Grant/Award Numbers: 307933/2018-0 and 164692/2020-7; FAPEMIG, Brazil, Grant/Award Number: PPM-00053-17; PROPG-CAPES/FAPEAM Scholarship Program: 88887.217045/2018-00; FAPEAM, Brazil; and CAPES - Finance Code 001, Brazil.

CONFLICT OF INTEREST

Authors have no conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Luiz A. Q. Cordovil Jr https://orcid.org/0000-0001-9503-856X
Pedro H. S. Coutinho https://orcid.org/0000-0002-6043-7103
Iury Bessa https://orcid.org/0000-0002-6603-3476
Márcia L. C. Peixoto https://orcid.org/0000-0001-7369-6297
Reinaldo Martínez Palhares https://orcid.org/0000-0003-2470-4240

REFERENCES

1. Hespanha JP, Naghshtabrizi P, Xu Y. A survey of recent results in networked control systems. *Proc IEEE*. 2007;95(1):138-162.
2. Dolk VS, Borgers DP, WPMH H. Output-based and decentralized dynamic event-triggered control with guaranteed $\mathcal{L}_p$-gain performance and zeno-freeness. *IEEE Trans Automat Contr*. 2017;62(1):34-49.
3. Johansson KH, Törngren M, Nielsen L. Vehicle applications of controller area network. *Handbook of Networked and Embedded Control Systems*. Springer; 2005:741-765.
4. Mahmoud MS, Hussain SA, Abido MA. Modeling and control of microgrid: an overview. *J Franklin Inst*. 2014;351(5):2822-2859.
5. Wang YL, Han QL. Network-based modelling and dynamic output feedback control for unmanned marine vehicles in network environments. *Automatica*. 2018;91:43-53.
6. Zhang XM, Han QL, Ge X, et al. Networked control systems: a survey of trends and techniques. *IEEE/CAA J Automat Sin*. 2019;7(1):1-17.
7. Brunner FD, Heemels WPMH, Allgöwer F. Event-triggered and self-triggered control for linear systems based on reachable sets. *Automatica*. 2019;101:15-26.
8. Tabuada P. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Trans Automat Contr*. 2007;52(9):1680-1685.

9. Girard A. Dynamic triggering mechanisms for event-triggered control. *IEEE Trans Automat Contr*. 2015;60(7):1992-1997.
10. Coutinho PHS, Palhares RM. Dynamic periodic event-triggered gain-scheduling control co-design for quasi-LPV systems. *Nonlinear Anal Hybrid Syst*. 2021;41:101044.
11. Coutinho PHS, Palhares RM. Co-design of dynamic event-triggered gain-scheduling control for a class of nonlinear systems. *IEEE Trans Automat Contr*. 2021;1-8. doi:10.1109/TAC.2021.3108498
12. Lei Y, Wang YW, Yang W, Liu ZW. Distributed control of heterogeneous multi-agent systems with unknown control directions via event/self-triggered communication. *J Frankl Inst*. 2020;357(17):12163-12179.
13. Wang YW, Lei Y, Bian T, Guan ZH. Distributed control of nonlinear multiagent systems with unknown and nonidentical control directions via event-triggered communication. *IEEE Trans Cybern*. 2020;50(5):1820-1832.
14. Nguyen AT, Taniguchi T, Eciolaza L, Campos V, Palhares R, Sugeno M. Fuzzy control systems: past, present and future. *IEEE Comput Intell Mag*. 2019;14(1):56-68.
15. Bessa I, Puig V, Palhares RM. TS fuzzy reconfiguration blocks for fault tolerant control of nonlinear systems. *J Franklin Inst*. 2020;357(8):4592-4623.
16. Araújo RF, Torres LAB, Palhares RM. Distributed control of networked nonlinear systems via interconnected Takagi-Sugeno fuzzy systems with nonlinear consequent. *IEEE Trans Syst Man Cybern Syst*. 2021;51(8):4858-4867.
17. Sun W, Wu YQ, Sun ZY. Command filter-based finite-time adaptive fuzzy control for uncertain nonlinear systems with prescribed performance. *IEEE Trans Fuzzy Syst*. 2020;28(12):3161-3170.
18. Zhu Z, Pan Y, Zhou Q, Lu C. Event-triggered adaptive fuzzy control for stochastic nonlinear systems with unmeasured states and unknown backlash-like hysteresis. *IEEE Trans Fuzzy Syst*. 2021;29(5):1273-1283.
19. Yin S, Li X, Gao H, Kaynak O. Data-based techniques focused on modern industry: an overview. *IEEE Trans Ind Electron*. 2015;62(1):657-667.
20. Maiworm M, Limon D, Findeisen R. Online learning-based model predictive control with Gaussian process models and stability guarantees. *Int J Robust Nonlinear Control*. 2021;31(18):8785-8812. doi:10.1002/rnc.5361
21. Chen J, Hua C, Guan X. Fast data-driven iterative event-triggered control for nonlinear networked discrete systems with data dropouts and sensor saturation. *J Franklin Inst*. 2020;357(13):8364-8382.
22. Liang Y, Zhang H, Duan J, Sun S. Event-triggered reinforcement learning H_∞ control design for constrained-input nonlinear systems subject to actuator failures. *Inf Sci*. 2021;543:273-295.
23. Yang X, He H. Decentralized event-triggered control for a class of nonlinear-interconnected systems using reinforcement learning. *IEEE Trans Cybern*. 2021;51(2):635-648.
24. Yang Y, Vamvoudakis KG, Ferraz H, Modares H. Dynamic intermittent Q-learning–based model-free suboptimal co-design of $\mathscr{L}_2$-stabilization. *Int J Robust Nonlinear Control*. 2019;29(9):2673-2694.
25. Lin N, Chi R, Huang B. Event-triggered model-free adaptive control. *IEEE Trans Syst Man Cybern Syst*. 2021;51(6):3358-3369.
26. Xue S, Luo B, Liu D. Event-triggered adaptive dynamic programming for zero-sum game of partially unknown continuous-time nonlinear systems. *IEEE Trans Syst Man Cybern Syst*. 2020;50(9):3189-3199.
27. Leite D, Palhares RM, Campos VC, Gomide F. Evolving granular fuzzy model-based control of nonlinear dynamic systems. *IEEE Trans Fuzzy Syst*. 2015;23(4):923-938.
28. Camargos MO, Bessa I, D'Angelo MFSV, Cosme LB, Palhares RM. Data-driven prognostics of rolling element bearings using a novel error based evolving Takagi-Sugeno fuzzy model. *Appl Soft Comput*. 2020;96:106628.
29. Luo C, Wang H. Fuzzy forecasting for long-term time series based on time-variant fuzzy information granules. *Appl Soft Comput*. 2020;88:106046.
30. Shahparast H, Mansoori EG. Developing an online general type-2 fuzzy classifier using evolving type-1 rules. *Int J Approx Reason*. 2019;113:336-353.
31. Bezerra CG, Costa BSJ, Guedes LA, Angelov PP. An evolving approach to data streams clustering based on typicality and eccentricity data analytics. *Inf Sci*. 2020;518:13-28.
32. Škrjanc I. Cluster-volume-based merging approach for incrementally evolving fuzzy Gaussian clustering–eGAUSS+. *IEEE Trans Fuzzy Syst*. 2020;28(9):2222-2231.
33. Blažič S, Škrjanc I. Hybrid system identification by incremental fuzzy C-regression Clustering. Proceedings of the 2020 IEEE International Conference on Fuzzy Systems; 2020:1-7; Glasgow, UK.
34. Goel A, Bruce AL, Bernstein DS. Recursive least squares with variable-direction forgetting: compensating for the loss of persistency [Lecture notes]. *IEEE Control Syst Mag*. 2020;40(4):80-102.
35. Cordovil LAQ, Coutinho PHS, Bessa I, D'Angelo MFSV, Palhares RM. Uncertain data modeling based on evolving ellipsoidal fuzzy information granules. *IEEE Trans Fuzzy Syst*. 2020;28(10):2427-2436.
36. Lughofer E. *Evolving Fuzzy Systems-Methodologies, Advanced Concepts and Applications*. Vol 53. Springer; 2011.
37. Bento A, Oliveira L, Scola IR, Leite VJ, Gomide F. Evolving granular control with high-gain observers for feedback linearizable nonlinear systems. *Evol Syst*. 2021;12:935-948.
38. Pedrycz W, Wang X. Designing fuzzy sets with the use of the parametric principle of justifiable granularity. *IEEE Trans Fuzzy Syst*. 2016;24(2):489-496.
39. Wang G, Shi P, Wang B, Zhang J. Fuzzy *n*-ellipsoid numbers and representations of uncertain multichannel digital information. *IEEE Trans Fuzzy Syst*. 2014;22(5):1113-1126.

40. Moshtaghi M, Leckie C, Bezdek JC. Online clustering of multivariate time-series. Proceedings of the 2016 SIAM International Conference on Data Mining; 2016:360-368.
41. Angelov PP, Filev DP. An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Trans Syst Man Cybern B Cybern*. 2004;34(1):484-498.
42. Arana-Daniel N, Lopez-Franco C, Alanis AY. *Bio-inspired Algorithms for Engineering*. Elsevier Science; 2018.
43. Peixoto MLC, Braga MF, Palhares RM. Gain-scheduled control for discrete-time nonlinear parameter-varying systems with time-varying delays. *IET Control Theory Appl*. 2020;14(19):3217-3229.

How to cite this article: Cordovil LAQ, Coutinho PHS, Bessa I, Peixoto MLC, Palhares RM. Learning event-triggered control based on evolving data-driven fuzzy granular models. *Int J Robust Nonlinear Control*. 2022;32(5):2805-2827. doi: 10.1002/rnc.6024