

Evolving Granular Fuzzy Model-Based Control of Nonlinear Dynamic Systems

Daniel Leite, *Member, IEEE*, Reinaldo M. Palhares, Victor C. S. Campos,
and Fernando Gomide, *Senior Member, IEEE*

Abstract—Unknown nonstationary processes require modeling and control design to be done in real time using streams of data collected from the process. The purpose is to stabilize the closed-loop system under changes of the operating conditions and process parameters. This paper introduces a model-based evolving granular fuzzy control approach as a step toward the development of a general framework for online modeling and control of unknown nonstationary processes with no human intervention. An incremental learning algorithm is introduced to develop and adapt the structure and parameters of the process model and controller based on information extracted from uncertain data streams. State feedback control laws and closed-loop stability are obtained from the solution of relaxed linear matrix inequalities derived from a fuzzy Lyapunov function. Bounded control inputs are also taken into account in the control system design. We explain the role of fuzzy granular data and the use of parallel distributed compensation. Fuzzy granular computation provides a way to handle data uncertainty and facilitates incorporation of domain knowledge. Although the evolving granular approach is oriented to control systems whose dynamics is complex and unknown, for expositional clarity, we consider online modeling and stabilization of the well-known Lorenz chaos as an illustrative example.

Index Terms—Dynamic systems, evolving systems, fuzzy control, fuzzy modeling, granular computation.

I. INTRODUCTION

MOST real-world processes are characterized by nonlinear behavior and uncertain time-varying parameters, which significantly hampers the description of their dynamical behavior using differential equations derived from first principles. Online modeling and control of nonlinear dynamical processes require a sophisticated theoretical framework and methods for closed-loop stability analysis. In addition, constraints and performance indices should be satisfied under adverse conditions, including disturbances and uncertainty [1]–[3].

Manuscript received January 23, 2014; revised April 4, 2014; accepted May 27, 2014. Date of publication June 30, 2014; date of current version July 31, 2015. This work was supported by CNPq (the Brazilian National Research Council) under Grant 500616/2013-2, Grant 302834/2010-9, Grant 143919/2011-3, and Grant 304596/2009-4. The work of R. M. Palhares was supported by the Research Foundation of the State of Amazonas, FAPEAM.

D. Leite, R. M. Palhares, and V. C. S. Campos are with the Department of Electronics Engineering, Federal University of Minas Gerais, Belo Horizonte 31270-901, Brazil (e-mail: danfl7@dca.fee.unicamp.br; palhares@cpdee.ufmg.br; victor@cpdee.ufmg.br).

F. Gomide is with the School of Electrical and Computer Engineering, University of Campinas, Campinas 13083-852, Brazil (e-mail: gomide@dca.fee.unicamp.br).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TFUZZ.2014.2333774

Robust control [4] has long been recognized as a general and effective control approach provided that bounds on the uncertainty of the process parameters are known, and fairly accurate knowledge about the structure of the process model is available. There is no evidence, however, that a robust control system designed offline will perform properly if the process or the surrounding environment changes, especially if the uncertainty tolerance range is small [5], [6]. Among the reasons for the changes are aging, wearing, operation mode variation, faults, and seasonality [7]. A persuasive mechanism to deal with time-varying dynamics in modeling and control problems is adaptation.

Adaptive control consists of an array of techniques for automatic adjustment of controllers to achieve and keep a desired performance, even if the process parameters change over time [2]. Adaptation is performed based on input–output data collected from the process. Although the indirect method for adaptive control [8], [9] is well developed for linear models and controllers, it is not for the case of nonlinear, nonstationary, and uncertain systems [5]. Moreover, adaptive control usually assumes fixed model and controller structure and is mainly applied to parameters. When the process is subject to abrupt and rapid changes, the adaptation transients in classical adaptive control schemes are often unsatisfactory [6]. Cyclical changes, switches, and severe faults make adaptive control schemes prone to forgetfulness and performance deterioration during a period of time just after the change. For control systems to be able to adapt to a variety of unexpected changes, dynamic learning is essential.

Learning is a fundamental characteristic of intelligent control systems [1], [10]. Different from adaptive systems, intelligent systems can change its actions to perform a task more effectively due to increasing knowledge related to the task. Intelligent control schemes based on fuzzy systems [11]–[13], recursive neural networks [14], and hybrids [15], [16] are oriented to problems that are hardly formulated and analyzed in the differential equation mathematical framework owing to nonstationary parameters and uncertainties. The ability to learn from experience to improve their actions has made intelligent learning control an appealing control paradigm [1].

Fuzzy control was first introduced as a model-free offline control method [17], [18]. Besides accurate control inputs, we sometimes want models that carry meaningful information about the underlying process and, therefore, allow us to analyze its properties to design a controller oriented to attain certain performance requirements. A fuzzy model can provide prediction of future values of variables and, potentially, an interpretable rule base.

The rule base is helpful for process supervision and controller design. The possibility of analyzing stability, performance, and robustness has given rise to a keen interest in model-based fuzzy control [3], [11], [13], [19].

The fuzzy model of a time-varying dynamic process whose mathematical equations are unknown can only be obtained through the from-scratch evolution of its rule base and recursive adaptation of parameters. The rules of an associated fuzzy controller must be reviewed if the fuzzy model changes. However, complete redesign of the controller is unlikely due to tight time constraints of online environments. Complete redesign can be avoided if the model and controller rules own the same antecedent terms. Such a control scheme is referred to as parallel distributed compensation (PDC) [20]–[22]. In evolving PDC whenever a model rule is updated to assimilate a new behavior, only the corresponding local control law needs to be revised. A stable closed-loop control system can be achieved by means of, e.g., the direct Lyapunov method and using linear matrix inequalities (LMIs) [21]–[24].

LMIs are a useful tool to support the solution of a variety of control problems [23]. For example, stability, a basic need for any closed-loop system, is theoretically guaranteed if a feasible solution to a set of LMIs can be determined. A clear advantage of representing control problems using LMIs is the possibility to consider multiple control requirements (design specifications and constraints) by simply appending new LMIs [25]. An approach in addition to finding a feasible solution to a set of LMIs is to optimizing an objective function over the set of LMIs. If the feasibility or the optimization problem is convex, it can quite often be solved numerically in polynomial time using interior-point methods [13]. In online environment, whenever a local model changes, either the feasibility of some LMIs should be checked or an optimization problem should be solved. For this reason, PDC fuzzy control and a smaller number of less-conservative LMI constraints are topics of great importance [26], [27].

A method to design Takagi–Sugeno (T–S) fuzzy controllers [18] able to continuously evolve their rule bases was proposed in [5]. The method uses linear control laws generated from numerical data and is model-free, that is, it does not allow for any kind of *a priori* stability and robustness analysis of the resulting closed-loop system. Evolving granular fuzzy control is introduced in this paper as a twofold extension of evolving T–S (eTS) fuzzy control to consider model-based design and uncertain data. In evolving granular fuzzy control, a functional fuzzy model is developed from uncertain (granular) data stream and an incremental learning algorithm. In general, the data can be fuzzy intervals instead of pointwise numerical values subject to noise and errors. Simultaneous to model development, a fuzzy PDC controller is evolved and refined according to model parameter changes. The consequent parameters of the PDC controller are updated on a per sample basis from the solution of a relaxed LMI problem associated with active rules.

Granular data arise from expert knowledge, readings from unreliable sensors or summaries of numerical values over time periods [28]. In general, inaccurate measurements and perception-based information are granular by the very nature

of the measuring instruments and subjectivity of the natural language. Fuzzy set theory, as a practical framework of granular computing, capture our innate conception of transitional set belonging and uncertainty [29], [30]. For an account of earlier developments on fuzzy granular data stream modeling, see [31], [32]. Fusion of heterogeneous data based on generalized constraints is discussed in [33]–[35].

The remainder of this paper is structured as follows. Section II reviews the state of the art in evolving intelligent control systems emphasizing fuzzy model-based granular control and related approaches. A general view of a granular control system and its components is also provided. Section III describes an incremental algorithm to construct functional fuzzy models from fuzzy data. Section IV introduces the notion of evolving stable controller design within the PDC framework. Relaxed LMI conditions for stabilization of the closed-loop control system are obtained in agreement with the characteristics of the modeling algorithm. Section V contains an illustrative application considering stabilization of the Lorenz chaos at an equilibrium point. Conclusion and discussions are given in Section VI.

II. EVOLVING GRANULAR CONTROL

Evolving systems are self-adaptive structures with learning and summarization capabilities [36], [37]. These systems are equipped with incremental algorithms that build the structure of rule-based models and adapt antecedent and consequent parameters of rules. This learning paradigm mimics the evolution of humans during their life cycle in the sense that models of the world are progressively adapted to new needs, and learning is supported by past and recent experiences. Evolving systems generate new rules each time new data do not fit into the current model/understanding of the underlying process or phenomenon, but at the same time only when these new data are informative enough.

Evolving granular control aims to handle streams of numeric or fuzzy data and provide control inputs that should primarily stabilize a closed-loop system. Performance-oriented control design and amplitude-bounded signals can also be incorporated to the control system design. Real-time information is useful to improve the process model and controller to achieve and maintain a level of desired performance. For example, new granules and rules can be developed in response to changes in the data stream. Rules are simply retrieved from a rule base when needed so that performance requirements can be reached quicker under known circumstances.

The evolving fuzzy granular control approach is especially advantageous when the process equations are unknown and time varying. In particular, the research problem addressed in this paper is concerned with 1) the continuous, never ending, adaptation of functional fuzzy models from nonstationary and uncertain data streams and 2) the use of the resulting fuzzy models to support the design of state feedback controllers. The issue of system robustness is dealt with through the use of a learning algorithm able to handle uncertain data and fuzzy set-based representation of the data. The time variance issue is dealt

with by using a scattering approach to granulate the data, and state-space equations and control laws that can be adapted over time.

A. Related Work

This section summarizes recent research related to online fuzzy control of dynamic systems. We do not intend to give an exhaustive review of the literature. The purpose is to overview works closely related to that addressed in this paper.

Process control is a difficult task if all the information has to be estimated online. Two main classes of methods have been developed to solve the online process control problem. On one hand, results have been obtained where only the parameters of fuzzy local models (rule consequents) are assumed unknown [8], [38]–[40]. In this case, the number of fuzzy rules and parameters of membership functions are determined *a priori*. On the other hand, relatively few results exist in the field of evolving intelligence where all parameters and the structure of fuzzy controllers are estimated and gradually refined with no prior knowledge about the process [5], [12], [15]. In this case, control systems are model free and do not permit theoretical analysis of stability. Next, we emphasize the contributions of the proposed evolving granular control method over the methods mentioned previously.

A recursive least-squares (RLS) method was developed in [38] to adapt the parameters of fuzzy controllers. The method is based on T–S fuzzy models and requires information on the number of fuzzy rules and parameters of membership functions to be stipulated and fixed. An application example considering stabilization of an inverted pendulum on a cart shows that the method is robust to uncertainties such as unmodeled dynamics and bounded disturbances. Evolving granular control extends the method in [38] by allowing membership functions and the number of fuzzy rules to be time varying. Granular model and controller are fully self-adjustable to concept drifts and shifts and may provide significant performance improvements over robust controllers in dynamic environments.

Online self-evolving fuzzy and neurofuzzy controllers with global learning capabilities were introduced in [12] and [15], respectively. The structure of the proposed fuzzy and neurofuzzy controllers is built from scratch using past plant output and control input data. The controllers do not need information about the differential equations of the plant nor offline training. Therefore, self-evolving fuzzy and neurofuzzy controllers are somehow similar to evolving granular controllers. Although their learning algorithms are completely different, the essential issue is to provide a reduced number of interpretable rules. The advantages of granular evolving control over the approaches proposed in [12] and [15] concerns the possibility of 1) dealing with measurement uncertainty, 2) analyzing Lyapunov stability of the closed-loop system, and 3) handling bounded control inputs.

An adaptive model-based fuzzy controller was proposed in [40] where a quadratic Lyapunov function of the state estimation error is used to ensure closed-loop stability. Bounded control inputs are also guaranteed. Results on the control of a single-link

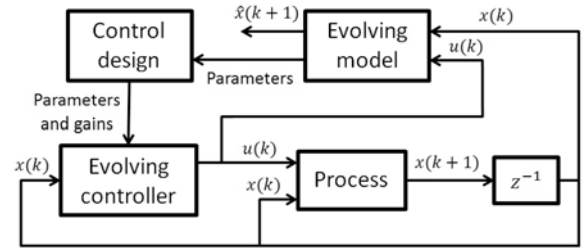


Fig. 1. Closed-loop evolving granular control system.

manipulator with flexible joint show that the method can deal with parameter changes. A drawback of the method is the assumption that for any input value, two membership functions have nonzero activation degree. More important, the structure of the controller is fixed, and prior information about the process is needed. For example, the membership functions of the system states are required to have their maximum values at certain linearization points. In evolving granular control, instead of using homogeneously distributed membership functions, a scattering method is used to granulate the data into membership functions. Structural adaptation of a model to capture abrupt changes is just as important as parametric adaptation to track gradual changes. Since evolving granular modeling in closed loop is made with effective control inputs and states, a more realistic process model is obtained. An accurate model is decisive for the stabilization of the original system. Finally, fuzzy Lyapunov functions, which tend to be less conservative than the quadratic functions adopted in [40], are employed to guarantee local stability. As a result, less-conservative LMI conditions associated with active rules should be satisfied.

B. Control System Overview

The block diagram of the evolving granular control system addressed in this paper is shown in Fig. 1. In the figure, $\mathbf{x}(k)$ and $\mathbf{u}(k)$ denote vectors of states and controls at step k , $\hat{\mathbf{x}}(k+1)$ is an estimation of the state at $k+1$, and z^{-1} is the delay operator. We assume uncertain or unreliable sensor measurements, thus the elements of $\mathbf{x}(k)$ are fuzzy data in general, being real-valued pointwise data a special case of fuzzy data. The mathematical definition of fuzzy datum is given in Section III-B. We shall make the assumption that all states $\mathbf{x}(k)$ are available for measurement. Observer-based output feedback control will not be discussed. The *process* is a general term that extends to objects, machines, and a variety of virtual and real-world systems that can be controlled. Mobile robots, spacecrafts, chemical processes, medical equipment, financial market, and video games are instances of processes.

The *evolving model* comprises a set of granules $\{\mathcal{M}^1, \dots, \mathcal{M}^c\}$, where each granule \mathcal{M}^i is assembled from fuzzy sets \mathcal{M}_{ψ}^i , $\psi = 1, \dots, \Psi$, and is governed by a rule R^i . The evolving model is supported by 1) a data-stream-oriented granulation algorithm that decides whether to fit new data into existing granules or to create new granules and 2) an RLS-like algorithm to update the parameters of local state equations based on granular data. The idea is to aggregate finer data granules into coarser,

yet more meaningful, granular models. A smaller number of larger granules is useful for data compression and for assisting controller design and decision making. For example, information in a granular model can be used for LMI-based controller synthesis, process supervision, and fault diagnosis. Moreover, granular models facilitate incorporation of expert knowledge and perceptions of the physical world. For example, experts can confront the information in a rule base with their knowledge, change parameters, or supply additional rules to extend the validity range of the model. The complete set of rules divide the data domain into a lattice of overlapping axis-parallel hyper-boxes (a set of granules). State equations (consequent terms of rules) are valid inside their corresponding hyperbox.

The *control design* encompasses a set of constraints in the form of LMIs and an interior-point algorithm. If a feasible solution to the LMI problem is found, then Lyapunov stability of the closed-loop system and bounded control inputs can be assured. As a result of the feasibility problem, a local feedback gain matrix K^i is determined. If no feasible solution is found at a step, then K^i is kept as calculated in the previous step. Finally, if this is not possible, then the gains of the closest granule/rule can be temporarily adopted. As the feasibility problem is required to be solved in a short period of time, relaxed conditions for stability are discussed in Section IV, where only LMIs related to active rules need to be satisfied. The design of control laws based on the online-approximated fuzzy model may not guarantee stability of the original system. This issue is alleviated through incremental learning in closed loop.

The *evolving granular fuzzy controller* consists of a set of fuzzy rules whose antecedent terms are the same as those of the *evolving model* of the process. Each rule consequent is composed of a time-varying linear state feedback control law. The main idea is to derive each controller rule in real time so as to compensate each rule of the model in a sense that the original process loses energy and the state trajectories in the phase plane [41] are asymptotically forced to an equilibrium state (stabilization of a fixed point). Feedback gain matrices K^i are adjusted in response to changes of rules R^i of the process model. Fuzzy model and controller are nonlinear in general due to the overlapping of fuzzy sets.

The control system dynamics works as follows. At each step k , the control system provides a control action $\mathbf{u}(k)$ and then performs a step of model adaptation and a step of controller adaptation. This way, model and controller learn and evolve their structures and parameters online. If the model predicts the system states $\mathbf{x}(k)$ with sufficient accuracy, and a feasible solution to the LMI problem is found, then it is expected that the control system will be stable and that performance requirements and constraints will be satisfied.

III. EVOLVING GRANULAR SYSTEM MODELING

A. Evolving Fuzzy Modeling

Functional fuzzy rule-based model built from a stream of data is attractive whenever the underlying process is unknown or change over time. In general, a finite number of past states $\mathbf{x}(k), \mathbf{x}(k-1), \dots, \mathbf{x}(k-m)$, control inputs

$\mathbf{u}(k), \mathbf{u}(k-1), \dots, \mathbf{u}(k-n)$, and other exogenous variables can be considered as elements of the antecedent part of fuzzy rules. Nevertheless, a common assumption in model-based fuzzy control is that rules are of the form:

$$\begin{aligned} R^i: & \text{IF } x_1(k) \text{ is } \mathcal{M}_1^i \text{ AND } \dots \text{ AND } x_\Psi(k) \text{ is } \mathcal{M}_\Psi^i \\ & \text{THEN } \mathbf{x}^i(k+1) = A^i \mathbf{x}(k) + B^i \mathbf{u}(k) \end{aligned}$$

where $\mathbf{x}(k) = [x_1(k) \dots x_\psi(k) \dots x_\Psi(k)]^T$, and $\mathbf{u}(k) = [u_1(k) \dots u_\phi(k) \dots u_\Phi(k)]^T$; $i = 1, \dots, c$ is the number of rules. In evolving modeling, A^i and B^i are matrices of appropriate dimensions with variable coefficients; \mathcal{M}_ψ^i , $\psi = 1, \dots, \Psi$ are membership functions built in light of the data being available. The number of rules R^i , $i = 1, \dots, c$ is also variable. Superscript i on the left-hand side of the consequent equation means a local estimation.

Consequent matrices and the state and control vectors can be extended to include affine terms as follows:

$$\tilde{A}^i = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{a}_0^i & A^i \end{bmatrix}, \tilde{B}^i = \begin{bmatrix} 0 \\ B^i \end{bmatrix}, \tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}, \tilde{\mathbf{u}} = \begin{bmatrix} 0 \\ \mathbf{u} \end{bmatrix}$$

where $\mathbf{a}_0^i = [a_{10}^i \dots a_{\psi 0}^i \dots a_{\Psi 0}^i]^T$. Rules R^i are rewritten as

$$\begin{aligned} R^i: & \text{IF } x_1(k) \text{ is } \mathcal{M}_1^i \text{ AND } \dots \text{ AND } x_\Psi(k) \text{ is } \mathcal{M}_\Psi^i \\ & \text{THEN } \tilde{\mathbf{x}}^i(k+1) = \tilde{A}^i \tilde{\mathbf{x}}(k) + \tilde{B}^i \tilde{\mathbf{u}}(k). \end{aligned}$$

For simplicity, in the rest of the paper, we omit the tilde from the notation and consider affine models. For the same reason, the index k is omitted from the time-varying membership functions \mathcal{M}_ψ^i and matrices A^i and B^i .

The state estimate from the functional fuzzy model is found as the weighted mean value:

$$\mathbf{x}(k+1) = \sum_{i=1}^c \mu^{ir} \mathbf{x}^i(k+1) \quad (1)$$

where μ^{ir} is the rescaled activation degree of the i th rule

$$\mu^{ir} = \frac{\mu^i}{\sum_{i=1}^c \mu^i} \text{ so that } \mu^{ir} \geq 0 \text{ and } \sum_{i=1}^c \mu^{ir} = 1. \quad (2)$$

Activation degrees μ^i are determined using any conjunctive aggregation operator, e.g., a t -norm [42], [43]. t -norms (T) are commutative, associative, and monotone operators on the unit hypercube $[0, 1]^n$ whose boundary conditions are $T(\omega, \omega, \dots, 0) = 0$ and $T(\omega, 1, \dots, 1) = \omega$, $\omega \in [0, 1]$. The neutral element of t -norms is $e = 1$. In this study, we shall adopt the product t -norm. Thus

$$\mu^i = \prod_{\psi=1}^{\Psi} \mu_{\psi}^i \quad (3)$$

where μ_{ψ}^i is the degree of membership of $x_{\psi}(k)$ in \mathcal{M}_{ψ}^i . While many works assume that the activation degree μ^i of at least one rule R^i is nonzero, this is not the case in evolving environment since no fuzzy set exists *a priori*. Fuzzy sets and rules are

created and developed gradually to cover the input data domain. The number of rules c increases by a unit if $\mu^i = 0 \forall i$. In this case, $\mu^{c+1} = 1$, that is, the fuzzy sets of the new rule match the input data. Online development of fuzzy sets and rules is addressed in the next sections.

B. Fuzzy Data and Models

Fuzzy data may arise from measurements from unreliable sensors, expert judgment, imprecision introduced due to pre-processing steps and summaries of numeric data over time periods. Fuzzy data modeling generalizes numeric data modeling by allowing fuzzy interval granulation [32], [44].

Trapezoidal fuzzy data and models are of concern to this study. A generic trapezoidal fuzzy set $\mathcal{N} = (l, \lambda, \Lambda, L)$ allows the modeling of a wide class of granular objects [45]. A triangular fuzzy set is a trapezoid where $\lambda = \Lambda$, an interval is a trapezoid where $l = \lambda$ and $\Lambda = L$, and a singleton is a trapezoid where $l = \lambda = \Lambda = L$. Additional features that make the trapezoidal representation attractive comprise the following: 1) ease of acquiring the necessary parameters: Only four parameters need to be captured. A trapezoidal model is formed straightforwardly from a trapezoidal datum; and 2) many operations on trapezoids can be performed using the endpoints of intervals, which are level sets of trapezoids. The piecewise linearity of the trapezoidal representation allows calculation of only two level sets (core and support) to obtain a complete implementation.

A fuzzy set $\mathcal{N} : X \rightarrow [0, 1]$ is upper semicontinuous if the set $\{x \in X | \mu(x) > \alpha\}$ is closed, that is, if the α -cuts of \mathcal{N} are closed intervals. If the universe X is the set of real numbers and \mathcal{N} is normal, $\mu(x) = 1 \forall x \in [\lambda, \Lambda]$, then \mathcal{N} is a model of a fuzzy interval, with monotone increasing function $\zeta_{\mathcal{N}} : [l, \lambda[\rightarrow [0, 1]$, monotone decreasing function $\iota_{\mathcal{N}} :]\Lambda, L] \rightarrow [0, 1]$, and zero otherwise [43]. A fuzzy interval \mathcal{N} has the following canonical form:

$$\mathcal{N} : x \rightarrow \mu(x) = \begin{cases} \zeta_{\mathcal{N}}, & x \in [l, \lambda[\\ 1, & x \in [\lambda, \Lambda] \\ \iota_{\mathcal{N}}, & x \in]\Lambda, L] \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where x is a real number in X . The fuzzy interval \mathcal{N} satisfies the conditions of normality ($\mu(x) = 1$ for at least one $x \in X$) and convexity $\mu(\kappa x^1 + (1 - \kappa)x^2) \geq \min\{\mu(x^1), \mu(x^2)\}$, $x^1, x^2 \in X$, $\kappa \in [0, 1]$. If

$$\zeta_{\mathcal{N}} = \frac{x - l}{\lambda - l} \quad \text{and} \quad \iota_{\mathcal{N}} = \frac{L - x}{L - \Lambda} \quad (5)$$

then the fuzzy interval (4) reduces to the model of a trapezoidal membership function. Moreover, when $\lambda = \Lambda$, $\mu(x) = 1$ for a single element x in X . In this case, the corresponding fuzzy entity is a fuzzy number.

Denote $x = (\underline{x}, \bar{x}, \bar{\bar{x}})$ as a trapezoidal datum. The membership degree of x in the fuzzy set \mathcal{N} can be obtained from (4) if x is degenerated into a singleton. Otherwise, if x is a symmetric object, i.e., if $\underline{x} - \bar{x} = \bar{\bar{x}} - \bar{x} \neq 0$, its membership degree in \mathcal{N}

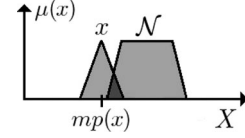


Fig. 2. Membership degree of the fuzzy datum x in the fuzzy model \mathcal{N} given by (6) is zero despite their similarity.

can be calculated using the midpoint of x :

$$\text{mp}(x) = \frac{\bar{x} + \underline{x}}{2}. \quad (6)$$

The center of gravity

$$\text{CoG}(x) = \frac{\underline{x} + 5\underline{x} + 5\bar{x} + \bar{\bar{x}}}{12} \quad (7)$$

is useful if x is asymmetric. Although it is apparent that these approximations of the true value are useful to facilitate computations, they contradict the purpose of taking into account the data uncertainty into fuzzy models. Additionally, in some situations, as that shown in Fig. 2, the midpoint (or center of area) approximation can give zero (or low) membership degree to significantly overlapped fuzzy objects. A measure of similarity between fuzzy granular data and models is needed to properly consider all relevant situations.

C. Similarity Between Fuzzy Objects

Similarity is a fundamental notion to construct rule-based systems from data. In this study, data and models are trapezoidal fuzzy objects. A possible similarity measure for trapezoids, say x and \mathcal{M}^i , is

$$S(x, \mathcal{M}^i) = 1 - D(x, \mathcal{M}^i) \quad (8)$$

where $D(x, \mathcal{M}^i)$ is a distance computed as follows:

$$D(x, \mathcal{M}^i) = \frac{|\underline{x} - l^i| + 2|\underline{x} - \lambda^i| + 2|\bar{x} - \Lambda^i| + |\bar{\bar{x}} - L^i|}{6}. \quad (9)$$

The value of S equals 1 for identical trapezoids (indicating the maximum degree of matching between them) and decreases linearly as x and \mathcal{M}^i withdraw from each other. In particular, (9) is a Hamming-like distance where the parameters of the trapezoids are directly compared. Core parameters have double weight in relation to support parameters. Although (8) and (9) are simple to compute, involving only basic arithmetic operations, there are no strong principled reasons to choose this measure. In fact, there is no generally accepted consensus on a best similarity measure [46].

Let the expansion region of a set \mathcal{M}^i be denoted by

$$E^i = [L^i - \rho, l^i + \rho] \quad (10)$$

where ρ is the maximum width that the set \mathcal{M}^i is allowed to expand to fit a datum x ; $L^i - l^i \leq \rho$ at any k . Define the membership degree of the datum x in the fuzzy set \mathcal{M}^i as $\mu^i = S(x, \mathcal{M}^i)$ if $x \in E^i$, and $\mu^i = 0$ otherwise.

A generalization of the similarity measure (8) for vectors of trapezoids, say $\mathbf{x} = [x_1 \dots x_\psi \dots x_\Psi]^T$ and $\mathbf{M}^i =$

$[\mathcal{M}_1^i \dots \mathcal{M}_\psi^i \dots \mathcal{M}_\Psi^i]^T$, is

$$S(\mathbf{x}, \mathcal{M}^i) = 1 - \frac{1}{6\Psi} \sum_{\psi=1}^{\Psi} (|\underline{x}_\psi - l_\psi^i| + 2|\underline{x}_\psi - \lambda_\psi^i| + 2|\bar{x}_\psi - \Lambda_\psi^i| + |\bar{x}_\psi - L_\psi^i|). \quad (11)$$

Then, $\mu^i = S(\mathbf{x}, \mathcal{M}^i)$ if $\mathbf{x} \in \mathbf{E}^i$. See [46] for a thorough discussion about similarity measures.

D. Incremental Adaptation

The purpose of adapting the parameters and structure of a functional fuzzy model is to assimilate new information about the process dynamics and keep an updated representation in response to unpredictable changes. This section addresses model structure identification and antecedent parameter estimation. An incremental learning method is suggested to avoid time-consuming training common to conventional learning methods based on multiple passes over the data.

Expansion regions \mathbf{E}^i [see (10)] help to decide if new input data belong to a granule in the input space. Different values of ρ produce different representations of the same dataset in different levels of granularities. For normalized data, ρ assumes values in $[0, 1]$. If ρ is equal to 0, then granules are not expanded. Learning creates a new rule for each sample, which causes overfitting and excessive complexity. If ρ is equal to 1, then a single granule covers the entire data domain. Evolvability is reached choosing intermediate values for ρ .

A rule is created whenever one or more entries of \mathbf{x} do not belong to the expansion regions \mathbf{E}^i of \mathcal{M}^i , $i = 1, \dots, c$. A new granule \mathcal{M}^{c+1} is constructed from fuzzy sets \mathcal{M}_ψ^{c+1} , $\psi = 1, \dots, \Psi$, whose parameters match \mathbf{x} , that is

$$\mathcal{M}_\psi^{c+1} = (l_\psi^{c+1}, \lambda_\psi^{c+1}, \Lambda_\psi^{c+1}, L_\psi^{c+1}) = (\underline{x}_\psi, \underline{x}_\psi, \bar{x}_\psi, \bar{x}_\psi). \quad (12)$$

Adaptation of an existing granule \mathcal{M}^i consists in expanding the support $[l_\psi^i, L_\psi^i]$ and updating the core $[\lambda_\psi^i, \Lambda_\psi^i]$ of its fuzzy sets. Among all granules, \mathcal{M}^i that can be expanded to include a sample \mathbf{x} , that with highest similarity according to (11) is chosen. Adaptation proceeds depending on where the datum x_ψ is placed. Conditions for support expansion are

$$\text{If } \underline{x}_\psi \in [L_\psi^i - \rho, l_\psi^i], \text{ then } l_\psi^i(\text{new}) = \underline{x}_\psi, \text{ and}$$

$$\text{If } \bar{x}_\psi \in [l_\psi^i, L_\psi^i + \rho], \text{ then } L_\psi^i(\text{new}) = \bar{x}_\psi.$$

Core parameters are updated recursively from

$$\lambda_\psi^i(\text{new}) = \frac{(w^i - 1)\lambda_\psi^i + \underline{x}_\psi}{w^i} \quad (13)$$

and

$$\Lambda_\psi^i(\text{new}) = \frac{(w^i - 1)\Lambda_\psi^i + \bar{x}_\psi}{w^i} \quad (14)$$

where w^i is the number of times that the granule \mathcal{M}^i was chosen to be adapted. Fig. 3 shows seven possible adaptation situations. In the figure, the datum $x = (\underline{x}, \underline{x}, \bar{x}, \bar{x})$ places either outside, partially inside, or inside the fuzzy set \mathcal{M}^i . The learning

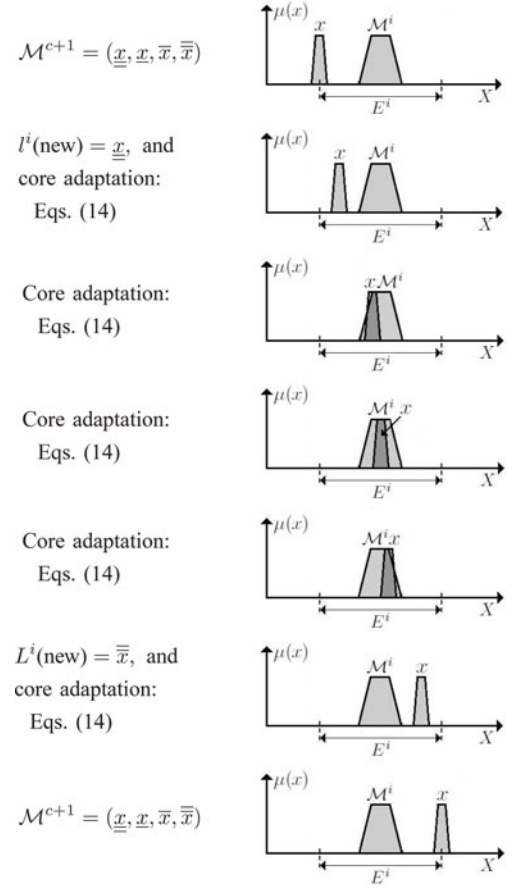


Fig. 3. Creation and recursive adaptation of fuzzy sets.

procedure creates a new granule \mathcal{M}^{c+1} or adapts the parameters of \mathcal{M}^i accordingly.

E. Specificity-Weighted Recursive Least-Squares Method

A recursive least squares-like algorithm is used to adapt the parameters of the rule consequents as follows. Consider the consequent part of rule R^i :

$$\mathbf{x}^i(k+1) = A^i \mathbf{x}(k) + B^i \mathbf{u}(k) \quad (15)$$

where $\mathbf{x} = [1 \ x_1 \dots x_\psi \dots x_\Psi]^T$ and $\mathbf{u} = [0 \ u_1 \dots u_\phi \dots u_\Phi]^T$. The elements of A^i are denoted $a_{\psi_1 \psi_2}^i$, $\psi_1, \psi_2 = 0, \dots, \Psi$. Matrix B^i is assumed to be known, time-invariant, and common to all rules. For an unknown B^i , the results follow straightforwardly. Rule R^i is chosen to be adapted whenever its antecedent part \mathcal{M}^i is more similar to $\mathbf{x}(k)$ than the antecedent part of the remaining rules.

When instance $\mathbf{x}(k+1)$ becomes known, (15) can be rearranged as

$$\hat{\mathbf{x}}(k+1) = A^i \mathbf{x}(k) \quad (16)$$

where $\hat{\mathbf{x}}(k+1) := \mathbf{x}^i(k+1) - B^i \mathbf{u}(k)$ and solved for A^i .

Expanding the ψ th row of (16), we get

$$\hat{x}_\psi(k+1) = a_{\psi 0}^i + a_{\psi 1}^i x_1(k) + \dots + a_{\psi \Psi}^i x_\Psi(k). \quad (17)$$

The standard RLS algorithm can be used for each row of (16) if we replace the trapezoids x_ψ by their midpoint (6) or center of gravity (7), depending on their symmetry. Data uncertainty is accounted for by weighing the adjustment of $a_{\psi_1 \psi_2}^i$ using specificity measures. Specificity measures refer to the amount of information conveyed by a fuzzy datum [47]. A fuzzy datum with greater uncertainty (lower specificity) may not be as important as one with smaller uncertainty (higher specificity).

Let $\mathbf{a}_\psi^i = [a_{\psi_0}^i \ a_{\psi_1}^i \ \dots \ a_{\psi_\Psi}^i]^T$ be the vector of unknown coefficients, $\mathfrak{X} = [1 \ \text{CoG}(x_1)(k) \ \dots \ \text{CoG}(x_\Psi)(k)]$ be the regression vector, and $\mathfrak{Y} = [\text{CoG}(x_\psi)(k+1)]$. Then, in matrix form, (17) is rewritten as

$$\mathfrak{Y} = \mathfrak{X} \mathbf{a}_\psi^i. \quad (18)$$

To estimate the coefficients \mathbf{a}_ψ^i , we let

$$\mathfrak{Y} = \mathfrak{X} \mathbf{a}_\psi^i + \Xi \quad (19)$$

where $\Xi := [\epsilon_\psi(k+1)]$, and

$$\epsilon_\psi(k+1) = \text{CoG}(x_\psi)(k+1) - \text{CoG}(\hat{x}_\psi)(k+1) \quad (20)$$

is the approximation error. While in batch estimation, the rows in \mathfrak{Y} , \mathfrak{X} , and Ξ increase with the number of available samples, in recursive mode, only two rows are kept, and we reformulate (18)–(20) as follows:

$$\mathfrak{Y} = \begin{bmatrix} \text{CoG}(x_\psi)(k) \\ \text{CoG}(x_\psi)(k+1) \end{bmatrix}, \Xi = \begin{bmatrix} \epsilon_\psi(k) \\ \epsilon_\psi(k+1) \end{bmatrix}, \text{ and}$$

$$\mathfrak{X} = \begin{bmatrix} 1 & \text{CoG}(x_1)(k-1) & \dots & \text{CoG}(x_\Psi)(k-1) \\ 1 & \text{CoG}(x_1)(k) & \dots & \text{CoG}(x_\Psi)(k) \end{bmatrix}. \quad (21)$$

The rows of the matrices in (21) refer to values before and just after adaptation. The RLS algorithm chooses \mathbf{a}_ψ^i to minimize the functional

$$J(\mathbf{a}_\psi^i) = \Xi^T \Xi. \quad (22)$$

\mathbf{a}_ψ^i is given by

$$\mathbf{a}_\psi^i = (\mathfrak{X}^T \mathfrak{X})^{-1} \mathfrak{X}^T \mathfrak{Y}. \quad (23)$$

Let $Q = (\mathfrak{X}^T \mathfrak{X})^{-1}$. From the matrix inversion lemma [48], we avoid inverting $\mathfrak{X}^T \mathfrak{X}$ using

$$Q(\text{new}) = Q(\text{old}) \left[I - \frac{\mathfrak{X}^T \mathfrak{X} Q(\text{old})}{1 + \mathfrak{X}_{(2)} Q(\text{old}) \mathfrak{X}_{(2)}^T} \right] \quad (24)$$

where I is the identity matrix, and $\mathfrak{X}_{(2)}$ is the second row of \mathfrak{X} . In practice, it is usual to choose large initial values for the entries of the main diagonal of Q . We use $Q(0) = 10^3 I$ as the default value.

After simple mathematical transformations, the vector of coefficients is rearranged recursively as

$$\mathbf{a}_\psi^i(\text{new}) = \mathbf{a}_\psi^i(\text{old}) + Q(\text{new}) \mathfrak{X}^T (\mathfrak{Y} - \mathfrak{X} \mathbf{a}_\psi^i(\text{old})) \quad (25)$$

or, similarly

$$\mathbf{a}_\psi^i(\text{new}) = \mathbf{a}_\psi^i(\text{old}) + Q(\text{new}) \mathfrak{X}^T \Xi. \quad (26)$$

Yager [45] defines the specificity of a trapezoid x_ψ as

$$\text{sp}(x_\psi) = 1 - \text{wdt}(x_{\psi(0.5)}). \quad (27)$$

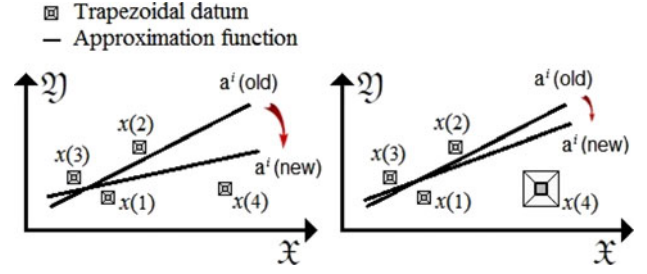


Fig. 4. Specificity-weighted RLS method.

This form of specificity measure means one minus the width of the 0.5 level set of x_ψ . In terms of the parameters of x_ψ , we get

$$\text{sp}(x_\psi) = 1 - \frac{(\bar{x}_\psi + \underline{\bar{x}}_\psi) - (\underline{x}_\psi + \underline{\underline{x}}_\psi)}{2}. \quad (28)$$

Let the specificity of \mathbf{x} be the diagonal matrix:

$$\text{sp}(\mathbf{x}) = \text{diag}([1 \ \text{sp}(x_1) \ \dots \ \text{sp}(x_\Psi)]). \quad (29)$$

Thus, we may add specificity into (26) to account for data uncertainty as follows:

$$\mathbf{a}_\psi^i(\text{new}) = \mathbf{a}_\psi^i(\text{old}) + \text{sp}(\mathbf{x}) Q(\text{new}) \mathfrak{X}^T \Xi. \quad (30)$$

Fig. 4 shows the idea of the specificity-weighted RLS algorithm. On the left side, the coefficients $\mathbf{a}^i(\text{old})$ of the approximation function result from recursive adaptation based on $x(1)$, $x(2)$, and $x(3)$. Note that the data granules $x(1)$, $x(2)$, and $x(3)$ are of the same size and, thus, have the same specificity. When the new datum $x(4)$ arrives (with the same specificity as that of previous data), the algorithm weights its contribution equivalently to the contribution of previous data to adapt $\mathbf{a}^i(\text{old})$ and yield $\mathbf{a}^i(\text{new})$. Conversely, on the right side, the specificity of the new datum $x(4)$ is lower than that of $x(1)$, $x(2)$, and $x(3)$. The higher uncertainty on the value of $x(4)$ causes a smaller adjustment of the function toward $x(4)$.

The specificity-weighted RLS algorithm described in this section is repeated for $\psi = 1, \dots, \Psi$ at each step. Detailed derivations of the RLS algorithm can be found in [2]. For a convergence analysis, see [49].

IV. EVOLVING CONTROL DESIGN

The PDC control design approach connects the antecedent part of controller rules directly with that of the model rules. Controller and model have the same membership functions, which makes controller design simpler and faster. The consequent part of the controller rules consists of linear state feedback laws. From a point of view, the PDC controller described in this section is an evolving, granular, and interpolative variation of the gain scheduling method [41].

Within the framework of functional fuzzy modeling and PDC control design, conditions for stability and performance can be formulated into an LMI problem. Finding a solution to the LMI is equivalent to finding a solution to the original problem [21], [23]. In the following, constraints such as Lyapunov stability conditions and bounded inputs are written as LMIs. The gains

of the evolving controller are determined automatically from the LMI formulation.

A. Closed-Loop Control System

To obtain feedback stabilizing conditions, the PDC approach is employed so that the controller rules have the same fuzzy sets as those of the fuzzy model. The stabilization rule is

R^i : IF $x_1(k)$ is \mathcal{M}_1^i AND ... AND $x_\Psi(k)$ is \mathcal{M}_Ψ^i
THEN $\mathbf{u}^i(k+1) = K^i \mathbf{x}(k)$

where, for simplicity of notation, $\mathbf{x}(k) = [1\text{CoG}(x_1)(k) \dots \text{CoG}(x_\Psi)(k)]^T$; and $\mathbf{u}(k) = [0 \ u_1(k) \dots u_\Phi(k)]^T$. K^i is a gain matrix of appropriate dimension to be determined so that the closed-loop system is asymptotically stable. Superscript i on the left-hand side of the consequent equation denotes a local control input. The overall control input is

$$\mathbf{u}(k+1) = \sum_{i=1}^c \mu^{ir} \mathbf{u}^i(k+1) \quad (31)$$

where μ^{ir} is the rescaled activation degree as in (2).

As the step k increases, (1) and (31) can be combined. The resulting closed-loop system is

$$\mathbf{x}(k+1) = \sum_{i=1}^c \sum_{j=1}^c \mu^{ir} \mu^{jr} G^{ij} \mathbf{x}(k) \quad (32)$$

where $G^{ij} := A^i + B^i K^j$, or, equivalently

$$\begin{aligned} \mathbf{x}(k+1) = & \sum_{i=1}^c (\mu^{ir})^2 G^{ii} \mathbf{x}(k) + \\ & + 2 \sum_{i < j} \mu^{ir} \mu^{jr} \left(\frac{G^{ij} + G^{ji}}{2} \right) \mathbf{x}(k). \end{aligned} \quad (33)$$

If the unforced system is stable, then K^j can be chosen to improve the transient response. Unstable systems require primarily the determination of an appropriate K^j to stabilize an equilibrium point. Trajectory following will be discussed elsewhere. A difficulty in evolving environment is that \mathcal{M}^i , A^i , B^i , and the number of rules c may be time varying.

B. Relaxed Stability Conditions

The stabilization problem based on the Lyapunov criterion can be translated in terms of finding a feasible solution to a set of LMIs. The following results have been reported in the literature [13], [50]. For completeness, in what follows, they are recalled. The stabilization theorem, which is object of this paper, extends these results.

Define a Lyapunov function

$$V(\mathbf{x}) = \mathbf{x}^T P \mathbf{x} \quad (34)$$

where P is a positive-definite matrix. Based on (34) and on the closed-loop system (32), the following stabilization result was derived and proved in [50] and [51].

Result 1: The closed-loop control system (32) is globally exponentially stable if there exists a positive-definite matrix $X = P^{-1}$ and a set of matrices Q^i , $i = 1, \dots, c$ such that the LMIs

$$\begin{bmatrix} -X & X(A^i)^T + (Q^j)^T (B^i)^T \\ A^i X + B^i Q^j & -X \end{bmatrix} < 0 \quad (35)$$

are satisfied for all combinations of $i, j = 1, \dots, c$.

This is a convex feasibility problem. If a feasible solution can be found, then the controller gains are given by

$$K^i = Q^i P, \quad i = 1, \dots, c. \quad (36)$$

The feasibility of (35) means that 1) a common P matrix for all rules c exists, 2) (34) is Lyapunov, and 3) the closed-loop system (32) is stable using the gains K^i as in (36).

Further results have been derived to reduce the number of LMIs and the conservatism of (35) [52], [53]. Conditions (35) are sufficient but not necessary for stability. A common P matrix may not exist for some stable systems. If the LMIs in (35) were less rigorous, that is, if the sufficiency necessity gap was smaller, then stabilizing gains could be obtained more easily and possibly faster. Processing time is a main concern in evolving environments. An approach to reduce conservatism is to consider fuzzy-combined positive-definite matrices P^i , $i = 1, \dots, c$.

A fuzzy Lyapunov function consists of a fuzzy combination of quadratic functions of the system states. Define a fuzzy combined Lyapunov function as

$$V(\mathbf{x}) = \sum_{i=1}^c \mu^{ir} \mathbf{x}^T P^i \mathbf{x} \quad (37)$$

where $P^i > 0 \ \forall i$. A stabilization result for the fuzzy system (32) based on (37) is as follows.

Result 2: The closed-loop control system (32) is globally exponentially stable if there exist positive-definite matrices $X^i = (P^i)^{-1}$ and matrices Q^i and Z^i , $i = 1, \dots, c$, such that

$$\begin{bmatrix} X^i - (Z^j)^T - Z^j & (Z^j)^T (A^i)^T + (Q^j)^T (B^i)^T \\ A^i Z^j + B^i Q^j & -X^k \end{bmatrix} < 0 \quad (38)$$

holds true for all combinations of $i, j, k = 1, \dots, c$.

For a proof, see [13]. The controller gains are

$$K^j = Q^j (Z^j)^{-1}, \quad j = 1, \dots, c. \quad (39)$$

Remark 1: If matrices X^i and $Z^i \ \forall i$ are equal, then Result 2 reduces to Result 1. The stabilization result based on (37) is less conservative than that based on (34) as a consequence of the fact that more degrees of freedom are added. The number of LMIs in (38) is larger than that in (35).

The conservatism of Result 2 can be reduced through a so-called right-hand-side relaxation. The result obtained and proved in [13] and [52] is as follows.

Result 3: The closed-loop control system (32) is globally exponentially stable if there exist positive-definite matrices $X^i = (P^i)^{-1}$, matrices Q^i and Z^i , $i = 1, \dots, c$, symmetric matrices \mathfrak{D}_s^i , $i, s = 1, \dots, c$, and matrices $\mathfrak{D}_{sj}^i = (\mathfrak{D}_{js}^i)^T$,

$i, j, s = 1, \dots, c; s < j$, such that

$$\begin{bmatrix} X^j - (Z^j)^T - Z^j + \mathfrak{D}_{jj}^i & (Z^j)^T (A^j)^T + (Q^j)^T (B^j)^T \\ A^j Z^j + B^j Q^j & -X^k \end{bmatrix} < 0 \quad (40)$$

$i, j, k = 1, \dots, c$

$$\begin{bmatrix} X^s - (Z^s)^T - Z^s + \mathfrak{D}_{ss}^i & (Z^s)^T (A^s)^T + (Q^s)^T (B^s)^T \\ A^s Z^s + B^s Q^s & -X^k \end{bmatrix} < 0$$

$$+ \begin{bmatrix} X^j - (Z^s)^T - Z^s + \mathfrak{D}_{js}^i & (Z^s)^T (A^j)^T + (Q^s)^T (B^j)^T \\ A^j Z^s + B^j Q^s & -X^k \end{bmatrix} < 0 \quad (41)$$

$i, j, k, s = 1, \dots, c; s < j$, and

$$\mathfrak{D}^i := \begin{bmatrix} 2\mathfrak{D}_{11}^i & \mathfrak{D}_{12}^i & \dots & \mathfrak{D}_{1c}^i \\ \mathfrak{D}_{21}^i & 2\mathfrak{D}_{22}^i & \dots & \mathfrak{D}_{2c}^i \\ \vdots & \vdots & \ddots & \vdots \\ \mathfrak{D}_{c1}^i & \mathfrak{D}_{c2}^i & \dots & 2\mathfrak{D}_{cc}^i \end{bmatrix} > 0 \quad (42)$$

$i = 1, \dots, c$, hold true.

If a feasible solution is found, the controller gains are

$$K^s = Q^s (Z^s)^{-1}, \quad s = 1, \dots, c. \quad (43)$$

Remark 2: If the entries of the matrices \mathfrak{D}^i are set to zero, Result 3 reduces to Result 2. Therefore, the stabilization result based on Result 3 is less conservative than that based on Result 2. However, the number of LMIs involved is higher.

The number of control rules is directly related to the complexity of analysis of LMI conditions. Finding a Lyapunov function for a large number of rules may be difficult [50]. Due to the characteristics of the learning algorithm described in Section III (which assumes that inactive rules do not change), recalculation of the gains (43) is needed only for the active rules at a step. Therefore, the number of LMIs in (40)–(42) can be greatly reduced by considering active rules. An important remark should be made at this point concerning global stability.

Remark 3: After the updating of the gains of the active rules at a step, if the granules assigned to these rules intersect with granules governed by rules that are not active, then such intersection regions are not guaranteed to be stable. However, if a new data sample falls within an unstable intersection region, then the corresponding gains are updated before a control input is given. Therefore, practically speaking, the control system is kept globally exponentially stable if the corresponding LMI problem is feasible. In the meantime, if a snapshot of the control rules is taken in a certain time step, then it may be the case that the closed-loop system is, theoretically, not stable everywhere.

Definition 1: The number of rules \mathfrak{r} activated by a given sample $\mathbf{x}(k)$ is equal to the number of terms $\mu^{ir}(\mathbf{x}(k)) \neq 0$, $i = 1, \dots, c$; μ^{ir} is the activation degree of rule R^i .

A more relaxed result can be derived from Result 3 by replacing the total number of rules c with the number of active rules \mathfrak{r} in a given time step. This is stated formally in the following.

Theorem: Given \mathfrak{r} , the number of active rules in a given time step, where $1 \leq \mathfrak{r} \leq c$. The closed-loop control system (32) is

globally exponentially stable if there exist positive-definite matrices $X^i = (P^i)^{-1}$, matrices Q^i and Z^i , $i = 1, \dots, \mathfrak{r}$, symmetric matrices \mathfrak{D}_s^i , $i, s = 1, \dots, \mathfrak{r}$, and matrices $\mathfrak{D}_{sj}^i = (\mathfrak{D}_{js}^i)^T$, $i, j, s = 1, \dots, \mathfrak{r}$; $s < j$, such that (40) holds true for $i, j = 1, \dots, \mathfrak{r}$. In addition, (41) must be true for $i, j, s = 1, \dots, \mathfrak{r}$; $s < j$, and

$$\mathfrak{D}^i := \begin{bmatrix} 2\mathfrak{D}_{11}^i & \mathfrak{D}_{12}^i & \dots & \mathfrak{D}_{1\mathfrak{r}}^i \\ \mathfrak{D}_{21}^i & 2\mathfrak{D}_{22}^i & \dots & \mathfrak{D}_{2\mathfrak{r}}^i \\ \vdots & \vdots & \ddots & \vdots \\ \mathfrak{D}_{\mathfrak{r}1}^i & \mathfrak{D}_{\mathfrak{r}2}^i & \dots & 2\mathfrak{D}_{\mathfrak{r}\mathfrak{r}}^i \end{bmatrix} > 0 \quad (44)$$

$i = 1, \dots, \mathfrak{r}$.

The controller gains assigned to active rules are

$$K^s = Q^s (Z^s)^{-1}, \quad s = 1, \dots, \mathfrak{r} \quad (45)$$

whereas the gains related to inactive rules are kept the same as computed in the previous step.

The proof of the theorem is in the Appendix. Efficient numerical algorithms for determining the feasibility of LMIs, such as (40), (41), and (44), are available. Interior point methods allow us to effectively decide about feasibility and to determine solutions in polynomial time.

C. Bounded Control Inputs

If the current state $\mathbf{x}(k)$ is known, then the following result is useful to handle bounded control input.

Result 4: Given positive-definite matrices $X^i = (P^i)^{-1}$ and matrices $Q^s = K^s Z^s$, $i, s = 1, \dots, \mathfrak{r}$, as in (45). The constraint $\|\mathbf{u}(k+1)\|_2 \leq \zeta$ is enforced if the LMIs

$$\begin{bmatrix} 1 & \mathbf{x}(k)^T \\ \mathbf{x}(k) & X^i \end{bmatrix} > 0 \quad (46)$$

and

$$\begin{bmatrix} X^i & (Q^s)^T \\ Q^s & \zeta^2 I \end{bmatrix} > 0 \quad (47)$$

$i, s = 1, \dots, \mathfrak{r}$ hold true.

The proof of Result 4 can be found in [21] with the current state $\mathbf{x}(k)$ replaced by the initial state $\mathbf{x}(0)$. Variable ζ bounds the maximum value allowed for control inputs, that is, it keeps the inputs within the limits of the actuators. LMIs (46) and (47) can be appended to LMIs (40), (41), and (44) in the design of stable fuzzy controllers satisfying input constraints.

V. ILLUSTRATIVE EXAMPLE

An illustrative example is given to show the potential of the evolving granular fuzzy control approach. Although the approach is oriented to control systems whose dynamics is complex and unknown, for expositional clarity, we elaborate on a benchmark problem of modern nonlinear dynamics, the Lorenz attractor. The Lorenz attractor is a nonlinear system that exhibits chaotic behavior for certain choices of parameters. We assume the Lorenz equations to be unknown; the equations are used only to generate a data stream. Therefore, the aim is to model and stabilize an unknown dynamical system using a stream of data. A granular fuzzy model and a fuzzy PDC controller are

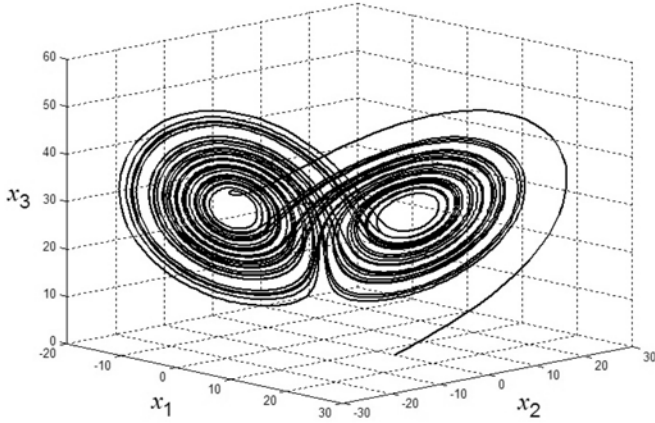


Fig. 5. Lorenz chaotic system: phase space trajectory.

evolved autonomously, with no human intervention, to stabilize the Lorenz system at the origin.

A. Lorenz Attractor

The Lorenz equations were derived as a simplified model of fluid convection, liquid or gases, induced by temperature change. The mathematical model of convection consists of three first-order differential equations:

$$\begin{aligned}\dot{x}_1 &= \sigma(x_2 - x_1) \\ \dot{x}_2 &= rx_1 - x_1x_3 - x_2 \\ \dot{x}_3 &= x_1x_2 - bx_3\end{aligned}\quad (48)$$

where x_1 is the speed of circulation of the fluid. Positive and negative values of x_1 represent clockwise and anticlockwise motion; x_2 is the difference in temperature between up and down fluids; and x_3 is the distortion from linearity of the vertical temperature profile. σ and r are the Prandtl and Rayleigh numbers, and b is a geometric factor [54]. The nonlinearities are x_1x_3 and x_1x_2 . If $\dot{x} = 0$, then we have three equilibrium points: $(0; 0; 0)$, $(\sqrt{b(r-1)}; \sqrt{b(r-1)}; r-1)$, and $(-\sqrt{b(r-1)}; -\sqrt{b(r-1)}; r-1)$. Similar to [54], we consider $\sigma = 10$, $r = 28$, and $b = 8/3$.

A modification of the continuous model (48) to an iterative recurrence model known as the Lorenz map is also widely used [55]. The iterative Lorenz equations with a control input term $u(k)$ added are

$$\begin{aligned}x_1(k+1) &= x_1(k) + \sigma(x_2(k) - x_1(k))dt + u(k) \\ x_2(k+1) &= x_2(k) + (rx_1(k) - x_1(k)x_3(k) - x_2(k))dt \\ x_3(k+1) &= x_3(k) + (x_1(k)x_2(k) - bx_3(k))dt\end{aligned}\quad (49)$$

where $dt = 0.005$ is the stepsize. The system is completely controllable and observable. Instead of solving (48) at each step using, e.g., the fourth-order Runge–Kutta method, simulations are carried out using (49), which is convenient for easier algorithm design. Unless otherwise stated, the initial state $\mathbf{x}(0)$ is $(13; 0; 0)$. As shown in Fig. 5, the trajectory of the unforced system in the phase space settles into an irregular aperiodic os-

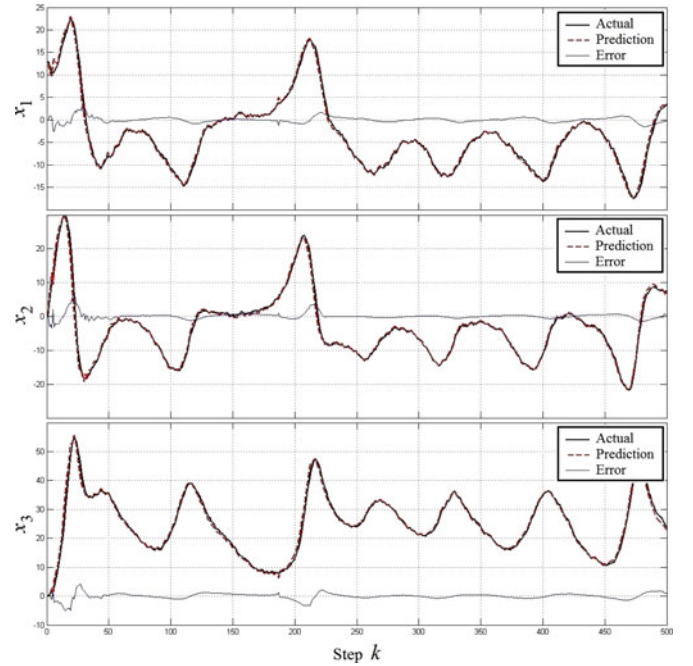


Fig. 6. One-step estimation of the Lorenz chaotic map.

cillation, which never repeats exactly. Although the trajectories are continually repelled from one unstable region to another, they are confined to a bounded set of zero volume (a fractal set) and manage to move in this set forever without intersecting.

B. Evolving Modeling of the Lorenz System

In the first experiment, a fuzzy model is evolved to approximate (49). The equations are perceived through pointwise input $([x_1(k) \ x_2(k) \ x_3(k)])$ and output $([x_1(k+1) \ x_2(k+1) \ x_3(k+1)])$ data. No data are available before learning starts; data are available stepwise to simulate a data stream. The one-step prediction results using the maximum width for granules $\rho = 40$ are shown in Fig. 6 for $k = 1, \dots, 500$. The root mean square error, calculated as

$$\text{RMSE} = \frac{1}{k_c} \sum_{k=1}^{k_c} \sqrt{\sum_{j=1}^3 (x_j(k+1) - \hat{x}_j(k+1))^2} \quad (50)$$

is $\text{RMSE} = 0.3855$ for $k_c = 500$. Three rules were developed during the simulation period. Their parameters are

Rule 1:

$$\mathcal{M}_1^1 = (2.6659, 11.6382, 11.6382, 20.6104)$$

$$\mathcal{M}_2^1 = (0, 14.9987, 14.9987, 29.9973)$$

$$\mathcal{M}_3^1 = (0, 19.4143, 19.4143, 38.8287)$$

$$A^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -4.5679 & -8.5766 & 9.4237 & -0.3404 \\ 209.8369 & 5.5858 & 1.0490 & -12.8230 \\ -143.5199 & 14.0343 & 10.3979 & -0.6229 \end{bmatrix}$$

Rule 2:

$$\mathcal{M}_1^2 = (6.6224, 14.5326, 14.5326, 22.4427)$$

$$\mathcal{M}_2^2 = (-16.3677, 4.1085, 4.1085, 24.5847)$$

$$\mathcal{M}_3^2 = (38.0141, 46.3665, 46.3665, 54.7189)$$

$$A^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 18.8130 & -5.8793 & 8.9295 & -1.7659 \\ 875.0623 & -8.7513 & -3.3158 & -22.1597 \\ 460.7916 & 21.4282 & 10.2167 & -18.3173 \end{bmatrix}$$

Rule 3:

$$\mathcal{M}_1^3 = (-17.5356, -6.1041, -6.1041, 5.3274)$$

$$\mathcal{M}_2^3 = (-21.7225, -6.4906, -6.4906, 8.7413)$$

$$\mathcal{M}_3^3 = (7.8274, 26.7340, 26.7340, 45.6405)$$

$$A^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -6.9656 & -9.4867 & 9.9042 & 0.3000 \\ -98.1609 & -4.4464 & 3.3558 & 4.0313 \\ 14.7558 & -9.6610 & -5.2738 & -4.5201 \end{bmatrix}$$

Vectors $B^i = [0 \ 1 \ 0 \ 0]^T$ are common to all rules.

From Fig. 6, the effectiveness of the evolving approach in predicting nonlinear systems without prior knowledge about the data and system equations can be verified. The error signals have relatively small amplitudes compared with the amplitudes of the system states. An important point to emphasize in this experiment is that due to exponential divergence of the trajectories for small differences in the measurements, parameters, or initial states, a nonevolving (offline-trained) modeling method is unlikely to track the trajectory of the states. Another point to be noted is that the higher the number of granules and rules, the more accurate the state estimation tends to be. However, the state estimation depends on the availability of sufficient data for choosing the local parameters.

C. Robustness to Uncertainty

An experiment was conducted to evaluate the ability of the evolving model in handling fuzzy data, and detecting and reacting to drifts and shifts in data streams. We consider the data as perceptions of the values of a variable. Imprecision of the values of x_j is represented by a fuzzy object of the form $(x_j - 0.5, x_j, x_j + 0.5)$. From $k = 1, \dots, 300$, the parameters of the Lorenz equations are set to $\sigma = 10$, $r = 28$, and $b = 8/3$. An abrupt change occurs at $k = 300$, where the parameters are shifted to $\sigma = r = b = 10$. At each step, from $k = 600$ to $k = 900$, an offset of 0.03 is added to σ , r , and b to produce gradual change of parameters. Fig. 7 shows the results for the state variable x_3 . The results for the remaining state variables are essentially the same.

Note from Fig. 7 that the error rate oscillates after the concept shift, but the quality of state estimates improves after few time steps. To maintain an acceptable level of prediction performance when the large and unknown change occurred, the learning algorithm created an additional fuzzy rule. Conversely, when gradual change of the values of the parameters occurred,

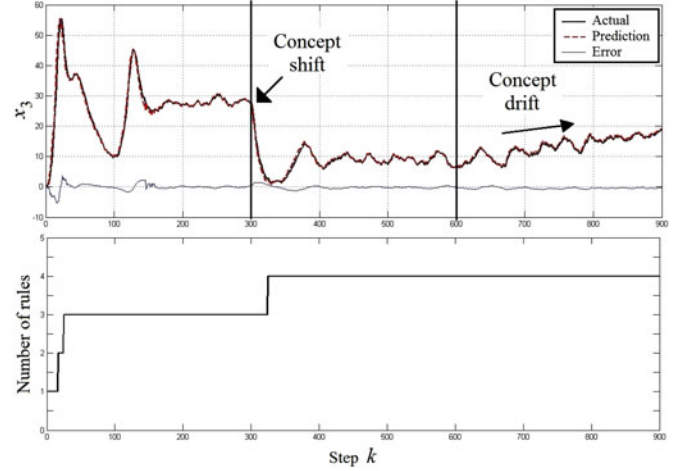


Fig. 7. One-step estimation of the variable x_3 of the Lorenz system subject to abrupt ($k = 300$) and gradual ($k = 600, \dots$) changes of parameters and evolution of the fuzzy rule base over the time.

TABLE I
LORENZ CHAOS—PREDICTION PERFORMANCE

Model	Avg. Rules	RMSE Best	RMSE Avg.
DENFIS	3.3 ± 0.5	0.3650	0.3864 ± 0.0128
eTS	3.1 ± 0.6	0.2585	0.2853 ± 0.0165
eGFM	3.4 ± 0.5	0.2080	0.2581 ± 0.0298
xTS	7.2 ± 1.2	0.2314	0.2526 ± 0.0095

the learning algorithm basically adapted the parameters of existing granules and rules to track the trajectory of the states. The evolving granular modeling method has shown to be robust to time-varying parameters and able to handle fuzzy data streams.

The evolving granular method was compared with alternative state-of-the-art evolving modeling methods. The following models were considered: eTS [7], dynamic evolving neurofuzzy inference system (DeNFIS) [56], extended Takagi–Sugeno (xTS) [7], and the evolving granular fuzzy model (abbreviated in Table I as eGFM) described in this paper. We prioritized model compactness and estimation performance. The models were developed from scratch, with no rules nor pretraining. Table I summarizes the results of one-step state estimation of the Lorenz chaos. The RMSE is calculated over nonnormalized data and averaged over ten runs. The number of samples, k_c [see (50)] is equal to 900 in each of the simulations.

The results of Table I show that, strictly speaking, xTS is the most accurate model from the average RMSE point of view. The eGFM produces an average of 3.4 ± 0.5 rules and achieves the best RMSE. eGFM provides more compact models, and its performance is comparable with xTS in terms of the average RMSE. The granular modeling approach does not take advantage from a large amount of local processing units (granules/rules) to achieve the average performance of 0.2581. eGFM benefits from a combination of ingredients concerning with structural assumptions, peculiarities of the learning algorithm, and fuzzy granular framework to attain its performance.

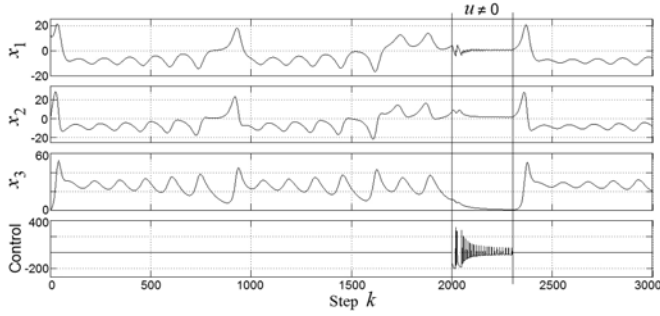


Fig. 8. Evolution of the nonforced states ($0 \leq k \leq 1999$; $k \geq 2301$) and stabilization of the Lorenz chaos ($2000 \leq k \leq 2301$).

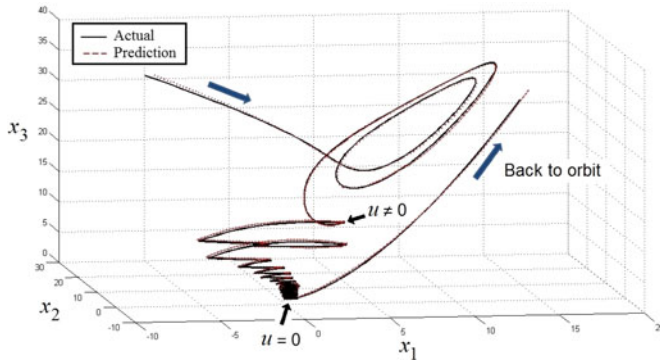


Fig. 9. Phase space trajectory of the forced three-state system ($u \neq 0$).

The effectiveness of the granular evolving approach in one-step estimation without prior knowledge about the data is verified in this experiment.

D. Stabilization of the Lorenz System

A fuzzy controller was evolved to stabilize the original Lorenz system based on information from the fuzzy model. The fuzzy model of the Lorenz system is only known approximately from a data stream, as shown in Section V-B. State trajectories are expected to settle down to an equilibrium when the controller is on and as time goes to infinity. Simulations concerning control design are performed using the Yalmip parser [57] and the SeDuMi solver [58] in MATLAB.

The evolving model starts learning from an empty rule base at $k = 1$. During the time interval from 2000 to 2300, the controller is ON, and a control action is performed. Otherwise, the control input is zero. Figs. 8 and 9 show the trajectories obtained. The designed controller was able to stabilize the origin, i.e., $x_1, x_2, x_3 \rightarrow 0$ as k increases. Stabilization is also possible if the control input is bounded. In this case, the transient response is fundamentally the same as that shown in Fig. 8. However, the state trajectories generally require a longer time interval to settle down to the equilibrium. If the bound on the control input is chosen to be a “very small” value, e.g., $\|u(k)\|_2 \leq 100 \forall k$, then the state trajectories can no longer escape the attractor.

Similarly as with the rules that assemble the model, the controller rules can be displayed at any step. For instance, a control rule at $k = 2300$ is

Rule 1:

$$\mathcal{M}_1^1 = (1.5757, 11.0515, 11.0515, 20.5273)$$

$$\mathcal{M}_2^1 = (0, 14.4673, 14.4673, 28.9347)$$

$$\mathcal{M}_3^1 = (0, 19.9041, 19.9041, 39.8083)$$

$$K^1 = [-8.6412 \ 14.3939 \ 5.7941]$$

The average time to process a data sample on a dual-core 2.5-GHz processor with 6-GB RAM was 13.7 ms. If the underlying process is known to have approximately constant parameters in a time horizon, adaptation of controller rules can be carried out after a number of time steps in order to handle high-frequency data streams. Stabilization by learning completely from scratch is not appropriate because a number of data samples are necessary for model development and least-squares estimation of parameters.

In the last set of experiments, the feasibility of stabilizing the Lorenz system without information about the system order was evaluated. First, we consider that only two state variables, namely x_1 and x_2 , are measured for model development. This means that a substantial part of the information is lost due to the lack of knowledge about variable x_3 . Instability of the closed-loop system was noticed in all simulations considering different initial states and instants of application of the control input. Evolving a state observer is absolutely necessary under such circumstances for estimating the unknown state. Evolving observers are out of the scope of this paper.

Second, we consider the existence of four-state variables. The fourth state variable x_4 was chosen to be a linear combination of x_1 and x_2 plus a noise term $\eta \in [-0.5, 0.5]$:

$$x_4(k+1) = x_4(k) + \left(\frac{1}{2}(x_1(k) + x_2(k)) \right) dt + \eta. \quad (51)$$

Addition of noise avoids numerical problems during the least-squares estimation of the A^i matrices. Uncertainty on the values of x_j was represented as a fuzzy object of the form $(x_j - 0.5, x_j, x_j + 0.5)$. We observed that although a feasible LMI solution is found in almost all time steps (meaning that the evolving fuzzy model is stabilizable), a variety of convergence situations arise regarding the original process. Fig. 10 shows two examples of stable trajectories for the hypothetical fourth-order Lorenz system. In Fig. 10(a), a smooth convergence with smaller oscillations than those of the third-order system shown in Fig. 9 is noticed. Conversely, in Fig. 10(b), some undesirable abrupt peaks occur after the application of the control input. Such peaks led the system to instability in some of the simulations; they are attributed to the simultaneous adaptation of model and controller as the state trajectories are guided toward a region of the phase space not previously covered by granules. Since no data were made available in the region around the origin (see Fig. 5), a very small modeling error or process variation can result in instability of the closed-loop control system.

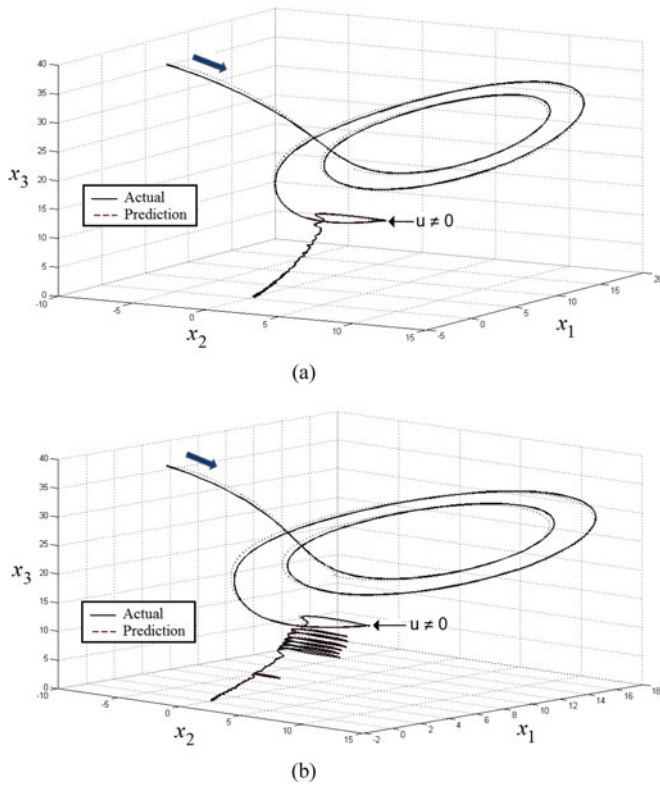


Fig. 10. Examples of state trajectories when the control action u is performed regarding the hypothetical fourth-order Lorenz system. (a) Smooth quasi-asymptotic convergence of the system states. (b) Irregular convergence of the system states.

We conclude from the last set of experiments that if the process is completely unknown (including unknown equations, parameters, system order, and historical data), stabilization based on online data streams is a complicated task in which analysis tools inherent to fuzzy model-based control methods, in principle, do not guarantee closed-loop stability. Nonstationary processes are sensitive to model and controller changes, especially if the control inputs drive the states through uncharted regions of the phase space. However, if *all state variables* can be measured, stability depends only on the model accuracy as well as on the existence of a Lyapunov function. The key problem is knowing how many state variables exist. A potential machine-learning solution approach to this problem is to monitor as many variables as possible and perform incremental selection of variables. It should be pointed out that adaptive and model-free evolving control methods, such as those discussed in [5], [12], [15], [38], and [40], which are oriented to “unknown” nonlinear systems, actually consider that the number of state variables is known.

The evolving granular model-based control method presented in this paper allows the structure and parameters of the fuzzy model and controller to be fully self-adjustable to concept changes. This is an eminent feature that distinguishes the proposed method from adaptive control methods. In addition, evolving granular model-based control allows dealing with measurement uncertainty and imposing bounds on the control in-

puts. These are key advantages of evolving granular control over model-free evolving control.

VI. CONCLUSION

This study has introduced a model-based evolving granular control approach for online modeling and stabilization of unknown time-varying processes. The approach is based on an incremental learning algorithm that simultaneously constructs and adapts functional fuzzy models, and the antecedent terms of fuzzy controllers using numeric and/or fuzzy data. A stable closed-loop system is obtained from a sequence of solutions concerning feasibility problems described in terms of relaxed LMIs. The inequalities were derived from a family of fuzzy Lyapunov functions, and from conditions that ensure bounded control inputs. Different from existing adaptive and evolving control methods, the evolving granular control method provides a way to handle data uncertainty and is able to develop the structure and parameters of fuzzy models and controllers with no prior information about the process. Experiments have considered the Lorenz attractor with changing parameters to illustrate the usefulness of the method developed.

Stabilization of nonlinear systems of unknown order has also been discussed. We assumed that only two state variables of the Lorenz system were measured for model development, as if the system was genuinely of lower order than the original Lorenz system. The evolving control system was not able to stabilize the hypothetical Lorenz system of “lower order.” The main reason for that is the lack of complete information about the system dynamics. Moreover, as state trajectories are forced into regions of the phase space not previously paved by granules, the system becomes quite sensitive to never-before-seen data. It was argued that the evolution of state observers is essential for estimating the unknown state. For the case of a Lorenz system of “higher order” than that of the original system, different convergence scenarios: smooth, oscillating, and irregular were observed. On top of these is the fact that although the fuzzy model is stabilizable, the original process may not be. This is due to the unpredictability inherent to nonstationary processes, which does not allow for the development of fully accurate fuzzy models such as those obtained using the well-known sector nonlinearity method. All the same, if all state variables can be measured, stability depends on the accuracy of the fuzzy model, as well as on the existence of a Lyapunov function.

In the future, we will continue to investigate issues related to evolving modeling and control of dynamic systems. For instance, as noted previously, further research is needed to account for unmeasurable state variables. A systematic design method for evolving fuzzy observers from input–output data shall be considered. In this paper, a direct incremental granulation algorithm was suggested to generate fuzzy membership functions. Although the method is fast, which is a major requirement in online environments, more elaborate methods can be devised to consider, for example, not only similarity among data within a granule but dissimilarity among data from different granules as well. We shall also look into issues concerning reliability

of evolving granular systems in dangerous and costly environments. Furthermore, control problems such as signal tracking and signal synchronization are also worth investigating.

APPENDIX PROOF OF THEOREM

The proof follows analogously as that of [13, Th. 4.4]. The system (32) is globally exponentially stable if

$$(G^{js})^T (X^k)^{-1} G^{js} - (X^i)^{-1} < 0 \quad (52)$$

with $i, j, k, s = 1, \dots, \mathfrak{r}$, and $G^{js} := A^j + B^j K^s$. Therefore, we will show that (40) and (41) imply (52) with gain matrices $K^s = Q^s (Z^s)^{-1}$, $s = 1, \dots, \mathfrak{r}$, as in (45).

Because $(X^j - Z^j)^T (X^j)^{-1} (X^j - Z^j) = X^j - Z^j - (Z^j)^T + (Z^j)^T (X^j)^{-1} Z^j \geq 0$, (40) can be rewritten as

$$\begin{bmatrix} -(Z^j)^T (X^j)^{-1} Z^j + \mathfrak{D}_{jj}^i & (Z^j)^T (G^j)^T \\ G^j Z^j & -X^k \end{bmatrix} < 0 \quad (53)$$

where $G^j := A^j + B^j K^j = A^j + B^j Q^j (Z^j)^{-1}$, and $i, j, k = 1, \dots, \mathfrak{r}$. From (53), by using Schur complement, we get

$$-(Z^j)^T (X^j)^{-1} Z^j + \mathfrak{D}_{jj}^i + (G^j Z^j)^T (X^k)^{-1} (G^j Z^j) < 0 \quad (54)$$

$i, j, k = 1, \dots, \mathfrak{r}$. Due to the positive definiteness of \mathfrak{D}_{jj}^i , and eliminating the quadratic terms Z^j , which are naturally positive definite, we obtain

$$-(X^j)^{-1} + (G^j)^T (X^k)^{-1} G^j < 0 \quad (55)$$

$i, j, k = 1, \dots, \mathfrak{r}$. Equation (55) is equivalent to (52); thus, we have shown that (40) implies (52).

Similarly, as $(X^s - Z^j)^T (X^s)^{-1} (X^s - Z^j) = X^s - Z^j - (Z^j)^T + (Z^j)^T (X^s)^{-1} Z^j \geq 0$, it follows from (41) that

$$\begin{bmatrix} -(Z^j)^T (X^s)^{-1} Z^j + \mathfrak{D}_{sj}^i & (Z^j)^T (G^{sj})^T \\ G^{sj} Z^j & -X^k \end{bmatrix} + \begin{bmatrix} -(Z^s)^T (X^j)^{-1} Z^s + \mathfrak{D}_{js}^i & (Z^s)^T (G^{js})^T \\ G^{js} Z^s & -X^k \end{bmatrix} < 0 \quad (56)$$

$i, j, k, s = 1, \dots, \mathfrak{r}$; $s < j$. Via the Schur complement, we obtain

$$\begin{aligned} & -(Z^j)^T (X^s)^{-1} Z^j + \mathfrak{D}_{sj}^i + (G^{sj} Z^j)^T (X^k)^{-1} (G^{sj} Z^j) \\ & -(Z^s)^T (X^j)^{-1} Z^s + \mathfrak{D}_{js}^i + (G^{js} Z^s)^T (X^k)^{-1} (G^{js} Z^s) \\ & < 0 \end{aligned} \quad (57)$$

$i, j, k, s = 1, \dots, \mathfrak{r}$; $s < j$. Since \mathfrak{D}_{sj}^i and \mathfrak{D}_{js}^i are positive values and by removing the quadratic terms related to Z^j and

Z^s , we obtain

$$\begin{aligned} & [(X^s)^{-1} + (G^{sj})^T (X^k)^{-1} G^{sj}] + \\ & + [-(X^j)^{-1} + (G^{js})^T (X^k)^{-1} G^{js}] < 0 \end{aligned} \quad (58)$$

$i, j, k, s = 1, \dots, \mathfrak{r}$; $s < j$. Both expressions in brackets are equivalent to (52). Therefore, like (40), (41) also implies (52) using (45), which completes the proof \square .

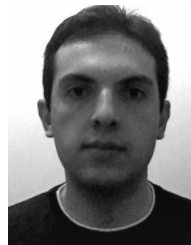
ACKNOWLEDGMENT

Comments and suggestions of the anonymous referees are gratefully acknowledged.

REFERENCES

- [1] P. J. Antsaklis, "Intelligent control," *Wiley Encycl. Electr. Electron. Eng.*, vol. 10, pp. 493–503, 1999.
- [2] K. J. Astrom and B. Wittenmark *Adaptive Control*, 2nd ed. Reading, MA, USA: Addison-Wesley, 1995.
- [3] A. Sala, T. M. Guerra, and R. Babuska, "Perspectives of fuzzy systems and control," *Fuzzy Sets Syst.*, vol. 156, no. 3, pp. 432–444, 2005.
- [4] K. Zhou, J. C. Doyle, and K. Glover *Robust and Optimal Control*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.
- [5] P. Angelov, "A fuzzy controller with evolving structure," *Inform. Sci.*, vol. 161, pp. 21–35, 2004.
- [6] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive Control: Algorithms, Analysis and Applications*, 2nd ed. London, U.K.: Springer-Verlag, 2011.
- [7] P. Angelov and R. Buswell, "Identification of evolving fuzzy rule-based models," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 5, pp. 667–677, Oct. 2002.
- [8] S. Labiod and T. M. Guerra, "Fuzzy adaptive control for a class of nonlinear systems with unknown control gain," *Evolving Syst.*, vol. 3, no. 1, pp. 57–64, 2012.
- [9] G. Tao, *Adaptive Control Design and Analysis*. Hoboken, NJ, USA: Wiley-IEEE Press, 2003.
- [10] R. F. Stengel, "Intelligent failure tolerant control," *IEEE Control Syst. Mag.*, vol. 11, no. 4, pp. 14–23, Jun. 1991.
- [11] R. Babuska and H. B. Verbruggen, "An overview of fuzzy modeling for control," *Control Eng. Pract.*, vol. 4, no. 11, pp. 1593–1606, 1996.
- [12] A. B. Cara, H. Pomares, I. Rojas, Z. Lendek, and R. Babuska, "Online self-evolving fuzzy controller with global learning capabilities," *Evolving Syst.*, vol. 1, no. 4, pp. 225–239, 2010.
- [13] G. Feng *Analysis and Synthesis of Fuzzy Control Systems: A Model-Based Approach*. Boca Raton, FL, USA: CRC, 2010.
- [14] C.-C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control," *IEEE Trans. Neural Netw.*, vol. 6, no. 1, pp. 144–156, Jan. 1995.
- [15] A. B. Cara, L. J. Herrera, H. Pomares, and I. Rojas, "New online self-evolving neuro fuzzy controller based on the TaSe-NF model," *Inform. Sci.*, vol. 220, pp. 226–243, 2013.
- [16] J.-S. R. Jang and C.-T. T. Sun, "Neuro-fuzzy modeling and control," *Proc. IEEE*, vol. 83, no. 3, pp. 378–406, Mar. 1995.
- [17] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc. IEEE*, vol. 121, no. 12, pp. 1585–1588, Dec. 1974.
- [18] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.
- [19] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, Oct. 2006.
- [20] J. Li, H. O. Wang, D. Niemann, and K. Tanaka, "Dynamic parallel distributed compensation for Takagi–Sugeno fuzzy systems: An LMI approach," *Inform. Sci.*, vol. 123, no. 3, pp. 201–221, 2000.
- [21] K. Tanaka and H. O. Wang, *Fuzzy Control Systems Design and Analysis: A LMI Approach*. New York, NY, USA: Wiley, 2001.
- [22] H. O. Wang and J. Li, "Parallel distributed compensation for Takagi–Sugeno fuzzy models: new stability conditions and dynamic feedback designs," *Acta Automatica Sinica*, vol. 27, no. 4, pp. 460–476, 2001.
- [23] S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan *Linear Matrix Inequalities in Systems and Control Theory*. Philadelphia, PA, USA: SIAM, 1994.

- [24] L. A. Mozelli, R. M. Palhares, and G. S. C. Avellar, "A systematic approach to improve multiple Lyapunov function stability and stabilization conditions for fuzzy systems," *Inform. Sci.*, vol. 179, no. 8, pp. 1149–1162, 2009.
- [25] J. G. VanAntwerp and R. D. Braatz, "A tutorial on linear and bilinear matrix inequalities," *J. Process Control*, vol. 10, pp. 363–385, 2000.
- [26] V. C. S. Campos, F. O. Souza, L. A. B. Torres, and R. M. Palhares, "New stability conditions based on piecewise fuzzy Lyapunov functions and tensor product transformations," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 4, pp. 748–760, Aug. 2013.
- [27] L. A. Mozelli, R. M. Palhares, and E. M. Mendes, "Equivalent techniques, extra comparisons, and less conservative control design for TS fuzzy systems," *IET Control Theory Appl.*, vol. 4, no. 12, pp. 2813–2822, 2010.
- [28] L. A. Zadeh, "Generalized theory of uncertainty (GTU)—Principal concepts and ideas," *Comput. Statist. Data Anal.*, vol. 51, pp. 15–46, 2006.
- [29] D. Dubois and H. Prade, Eds., *Fundamentals of Fuzzy Sets*. Norwell, MA, USA: Kluwer, 2000.
- [30] W. Lodwick, "An overview of flexibility and generalized uncertainty in optimization," *J. Comput. Appl. Math.*, vol. 31, no. 3, pp. 569–589, 2012.
- [31] D. Leite, P. Costa, and F. Gomide, "Evolving granular neural networks from fuzzy data streams," *Neural Netw.*, vol. 38, pp. 1–16, 2013.
- [32] D. Leite, R. Ballini, P. Costa, and F. Gomide, "Evolving fuzzy granular modeling from nonstationary fuzzy data streams," *Evolving Syst.*, vol. 3, no. 2, pp. 65–79, 2012.
- [33] D. Leite P. Costa and F. Gomide "Interval approach for evolving granular system modeling," *Learning in Non-stationary Environments: Methods and Applications*, M. Mouchaweh and E. Lughofer Eds. New York, NY, USA: Springer, 2012, pp. 271–300.
- [34] W. Pedrycz, J. C. Bezdek, and R. J. Hathaway, "Two nonparametric models for fusing heterogeneous fuzzy data," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 3, pp. 411–425, Aug. 1998.
- [35] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, no. 2, pp. 111–127, 1997.
- [36] P. Angelov, D. Filev and N. Kasabov Eds., *Evolving Intelligent Systems: Methodology and Applications (Series on Computational Intelligence)*. Hoboken, NJ, USA: Wiley-IEEE Press, 2010.
- [37] E. Lughofer, *Evolving Fuzzy Systems—Methodologies, Advanced Concepts and Applications*. Berlin, Germany: Springer-Verlag, 2011.
- [38] G. Feng, "An approach to adaptive control of fuzzy dynamic systems," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 268–275, Apr. 2002.
- [39] T. A. Johansen, "Fuzzy model based control: Stability robustness and performance issues," *IEEE Trans. Fuzzy Syst.*, vol. 2, no. 3, pp. 221–233, Aug. 1994.
- [40] C. W. Park and Y. W. Cho, "T-S model based indirect adaptive fuzzy control using online parameter estimation," *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 6, pp. 2293–2302, Dec. 2004.
- [41] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [42] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Berlin, Germany: Springer-Verlag, 2007.
- [43] W. Pedrycz and F. Gomide, *Fuzzy Systems Engineering: Toward Human-Centric Computing*. Hoboken, NJ, USA: Wiley, 2007.
- [44] W. Pedrycz, A. Skowron, and V. Kreinovich, Eds., *Handbook of Granular Computing*. Chichester, U.K.: Wiley, 2008.
- [45] R. R. Yager, *Learning from Imprecise Granular Data Using Trapezoidal Fuzzy Set Representations (Lecture Notes in Computer Science)*, vol. 4772, H. Prade and V. S. Subrahmanian, Eds. Berlin, Germany: Springer, 2007, pp. 244–254.
- [46] V. V. Cross and T. A. Sudkamp, *Similarity and Compatibility in Fuzzy Set Theory: Assessment and Applications (Studies in Fuzziness and Soft Computing)*. Heidelberg, Germany: Physica-Verlag, 2002.
- [47] R. R. Yager, "Measures of specificity over continuous spaces under similarity relations," *Fuzzy Sets Syst.*, vol. 159, pp. 2193–2210, 2008.
- [48] P. C. Young, *Recursive Estimation and Time-Series Analysis: An Introduction*. Berlin, Germany: Springer-Verlag, 1984.
- [49] C. R. Johnson, *Lectures on Adaptive Parameter Estimation*. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.
- [50] K. Tanaka, T. Ikeda, and H. O. Wang, "Fuzzy regulators and fuzzy observers: Relaxed stability conditions and LMI-based designs," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 2, pp. 250–265, May 1998.
- [51] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets Syst.*, vol. 12, pp. 135–156, 1992.
- [52] E. Kim and H. Lee, "New approaches to relaxed quadratic stability condition of fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 523–534, Oct. 2000.
- [53] W. J. Wang and C. H. Sun, "Relaxed stability and stabilization conditions for a T-S fuzzy discrete system," *Fuzzy Sets Syst.*, vol. 156, no. 2, pp. 208–225, 2005.
- [54] E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.*, vol. 20, no. 2, pp. 130–141, 1963.
- [55] W. Song and J. Liang, "Difference equation of the Lorenz system," *Int. J. Pure Appl. Math.*, vol. 83, no. 1, pp. 101–110, 2013.
- [56] N. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 144–154, Apr. 2002.
- [57] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Symp. Comput.-Aided Control Syst. Des.*, Taipei, Taiwan, 2004, pp. 284–289.
- [58] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Method Softw.*, vol. 11, no. 1–4, pp. 625–653, 1999.



Daniel Leite (S'07–M'13) received the Ph.D. degree in electrical engineering from the University of Campinas, Campinas, Brazil, in 2012 and the B.Sc. and M.Sc. degrees in electrical and control engineering from the Polytechnic Institute, Pontifical Catholic University of Minas Gerais, Belo Horizonte, Brazil, in 2005 and 2007, respectively.

From 2012 to 2013, he was a Lecturer with the Department of Electronics Engineering, Federal University of Minas Gerais, Belo Horizonte, where he is currently a Postdoctoral Researcher. His current research interests include adaptive and evolving systems from data streams, intelligent control, nonlinear system modeling, machine learning, and granular computing. He has contributed as a reviewer for several journals and conferences in his research fields.



Reinaldo M. Palhares received the B.Sc. degree in electrical engineering from the Federal University of Goiás, Goiânia, Brazil, in 1992 and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Campinas, Campinas, Brazil, in 1995 and 1998, respectively.

From 1998 to 2002, he was with the Polytechnic Institute, Pontifical Catholic University of Minas Gerais, Belo Horizonte, Brazil. Since 2002, he has been an Associate Professor with the Federal University of Minas Gerais, Belo Horizonte. His main research interests include robust linear/nonlinear control/filtering theory, time delays, fault detection and isolation, fuzzy theory, soft computing, and multi-criteria optimization theory and applications.

Dr. Palhares is a Member of the Editorial Board of the *Journal of Applied Mathematics*, as well as a Member of the Review Board of *Applied Intelligence*.



Victor C. S. Campos was born in Belo Horizonte, Brazil, in 1987. He received the B.Sc. degree in control and automation engineering in 2009 and the M.Sc. degree in electrical engineering in 2011, both from the Federal University of Minas Gerais (UFMG), Belo Horizonte, where he is currently working toward the Ph.D. degree in electrical engineering.

He is a Member of the Group for Research and Development of Autonomous Vehicles, UFMG. His current research interests include nonlinear control theory, Takagi–Sugeno fuzzy systems, and mobile robotics.



Fernando Gomide (M'83–SM'09) received the B.Sc. degree in electrical engineering from the Polytechnic Institute, Pontifical Catholic University of Minas Gerais, Belo Horizonte, Brazil, the M.Sc. degree in electrical engineering from the University of Campinas (Unicamp), Campinas, Brazil, and the Ph.D. degree in systems engineering from Case Western Reserve University, Cleveland, OH, USA.

He has been a Professor with the Department of Computer Engineering and Automation, School of Electrical and Computer Engineering, Unicamp, since 1983. His research interests include fuzzy systems; neural, granular, and evolutionary computation; intelligent data analysis; modeling, control and optimization; decision making; and applications.

Dr. Gomide was the past Vice President of the International Fuzzy Systems Association (IFSA), past IFSA Secretary, Member of the board of NAFIPS (North American Fuzzy Information Processing Society, and former Associate Editor of the *EUSFLAT Mathware and Soft Computing*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, FUZZY SETS AND SYSTEMS, and IEEE TRANSACTIONS ON FUZZY SYSTEMS, as past Member of the IEEE Emergent Technology Technical Committee. He currently serves on the editorial boards of *Evolving Systems*, *Fuzzy Optimization and Decision Making*, the *International Journal of Fuzzy Systems*, and *Soft Computing*. He was also a past Associate Editor of *Controle & Automacao*, the *Journal of the Brazilian Society of Automatics (SBA)*, and the Brazilian National Member Organization of IFAC and IFSA. He is on the Advisory Board of the *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, and the *Journal of Advanced Computational Intelligence*, and *Intelligent Automation and Soft Computing*. He is a Member of the NAFIPS, EUSFLAT, the IFSA Fellow Class of 2009. He received the NAFIPS K. S. Fu Award in 2011. He also serves on the IEEE Task Force on Adaptive Fuzzy Systems and the Fuzzy Technical Council of the IEEE Computational Intelligence Society.