State-Space Evolving Granular Control of Unknown Dynamic Systems

Daniel Leite[1]

[1]Adolfo Ibanez University (UAI), 2640 Diag. Las Torres, Santiago, Chile

Abstract

We present an approach for data-driven modeling and evolving control of unknown dynamic systems called State-Space Evolving Granular Control. The approach is based on elements of granular computing, discrete state-space systems, and online learning. First, the structure and parameters of a granular model is developed from a stream of state data. The model is formed by information granules comprising first-order difference equations. Partial activation of granules gives global nonlinear approximation capability. The model is supplied with an algorithm that constantly updates the granules toward covering new data; however, keeping memory of previous patterns. A granular controller is derived from the granular model for parallel distributed compensation. Instead of difference equations, the content of a control granule is a gain matrix, which can be redesigned in real-time from the solution of a relaxed locally-valid linear matrix inequality derived from a Lyapunov function and bounded control-input conditions. We have shown asymptotic stabilization of a chaotic map assuming no previous knowledge about the source that produces the stream of data.

Keywords

Evolving Control, Granular Computing, Dynamic Systems, Chaos

1. Introduction

1.1. Evolving Granular Systems

Evolving granular-computing models are structures with online learning, and summarizing and representational capabilities [1, 2, 3, 4]. An evolving granular model is equipped with an incremental algorithm that gradually builds its structure of interconnected elements. The elements are called information granules. Granules features, their location, size and geometry on search spaces, and the content they carry, may change at any time according: (i) to the spatio-temporal patterns noticed on a numerical, word, image, or relatively-finer granular-data stream; and (ii) to inter-granular relationships. From the dynamic aggregation of the local estimates – offered by the content of the current set of active granules – emerging nonlinear and time-varying behaviors and outputs can be produced by the model, including those often needed to track concept drifts and shifts.

Evolving granular systems [1] is a general-purpose online-learning framework to construct classifiers, regressors, predictors, and controllers in which at least one of several possible

OLUD 2022: First Workshop on Online Learning from Uncertain Data Streams, July 18, 2022, Padua, Italy.

daniel.furtado@uai.cl (D. Leite)
https://sites.google.com/site/danfl7 (D. Leite)
0000-0003-3135-2933 (D. Leite)

aspects of a problem may assume a granular value. In other words, non-pointwise uncertain characterization (e.g., interval, fuzzy, rough, statistical, and mixtures of uncertain objects) can be admitted to: original data instances, pre-processed or space-transformed instances; to model parameters, learning equations, covering regions, and so on [1, 3, 5]. Generalized constraints, in the sense of Zadeh's general theory of uncertainty [6], are used to delimit granules. Data granulation, granular computation, and mapping granules of different domains are attractive features to manage data uncertainty, to reduce complexity, and to reason in a structured and more meaningful way from the point of view of human-centered systems [7, 8, 9].

Evolving granular systems are aligned with issues and needs of Big data processing, fully autonomous systems, and eXplainable Artificial Intelligence. The incremental evolution of the structure of a granular model can be performed by supervised [10, 11], unsupervised [12, 13], semi-supervised [14, 15], and weakly (inaccurately) supervised [16] algorithms. As the algorithms operate on an instance-per-instance basis, they are generally very fast compared to machine learning algorithms in general. Nonetheless, time granulation – aiming at reducing the sampling rate of fast data streams and/or synchronizing concurrent data streams that are provided at random time intervals – has also been discussed [1]. A time granule describes the data stream for a certain time period. The result of dynamic time granulation is a unique granule per segment, which can further be used as an input for spatial computation. Of concern to the rest of this paper is state-space evolving fuzzy granular control only.

Fuzzy control, as an instance of granular control, is a well-founded framework to deal with complex nonlinear systems [17, 18, 19]. If the equations that describe a dynamical system are known, an exact fuzzy representation can be obtained, e.g., by the Sector Nonlinearity method [19, 20]. Thus, the accurateness on the assumptions toward obtaining nonlinear equations is the sole aspect that restricts the performance of a fuzzy model-based controller applied to a physical system to be similar to the performance required from the controller in computer simulations. Data-stream-driven fuzzy control comes into play when the system equations are time-varying, uncertain, and/or partially known or unknown [21, 22, 23]. In particular, time varying systems imply a clear drawback on the use of offline-designed model-based and model-free control methods in general, since some initial assumptions and some parameters inherent to the system change over time [21].

Adaptive and evolving control, as well as model-free and model-based control, should be differentiated. Evolving control is a step toward a higher level of autonomy compared to adaptive control and classical parameter identification methods [24, 25]. The evolving aspect accounts for unbounded amounts of data, changing concepts, and structural adaptation. Evolving control does not depend upon prior structural knowledge. In model-free control, the controller generally receives error-related variables (after comparing sensor data and reference values) and outputs control signals. In model-based control, besides accurate control signals, the accompanying model may carry meaningful information about the underlying process and, therefore, allow analysis of closed-loop system properties and the design of a controller oriented to attain certain performance requirements. The possibility of analyzing stability, performance, and robustness has given rise to a keen interest in model-based control [21, 26, 27].

This paper presents a state-space variety of evolving granular model, which we call SS-EGM, and an evolving granular control approach that we call SS-EGC. SS-EGC is designed based on the SS-EGM aiming at parallel distributed compensation in a sense that closed-loop Lyapunov stability and bounded control inputs are guaranteed [19, 21]. While stability is a fundamental concern in control design, bounded inputs may avoid actuator saturation. SS-EGM observes the dynamical system from a state data stream. The model provides one-step state estimates. Its structure gives support to control design. Model granules and their contents may change at any time step. In this case, the corresponding granules of the controller have their gains redesigned from the solution of a relaxed linear matrix inequality (LMI). Efficient LMI parser and solver, Yalmip'18 [28], and Mosek'20 [29], are utilized so that new gains for the changed control granules are secured quickly in online environment.

The effectiveness of the SS-EGM as a one-step predictor, and that of SS-EGC on regularizing a chaotic system, known as the Henon map [30], is demonstrated. Chaotic systems are nonlinear systems whose trajectories are sensitive to the initial conditions, parameter variations, and perturbation. The stabilization of unknown chaotic systems is challenging since slightly inaccurate parameter estimates may trigger undesired closed-loop behavior [31]. Although trajectories of chaotic systems are bounded, this is not the case when the system is influenced by control, that is, the solution may escape in finite time [32]. Broadly speaking, we have achieved stabilization of unknown, nonlinear and nonstationary systems using no *a priori* information and fully-autonomous online granular learning. We visualize applications on secure communication, such as in smart IoT and synchronization of cyber-physical systems; suppression of interference and artifacts in cardiac, electroencephalogram, and speech signals; and control of switched and time-varying systems in general, such as mobile robots and unmanned aerial vehicles, to mention some.

1.2. Problem definition

Consider a discrete, nonlinear and time-varying system,

$$\begin{aligned} \mathbf{x}(k+1) &= f(\mathbf{x}(k), \mathbf{u}(k), k) \\ \mathbf{y}(k+1) &= h\left(\mathbf{x}(k), k\right), \end{aligned} \tag{1}$$

in which $\mathbf{x}(k) \in R^n$, $\mathbf{u}(k) \in R^m$, and $\mathbf{y}(k) \in R^p$ are the state, input and output vectors at time step k. The maps f and h describe the relations (1). Consider all states accessible. Then, $\mathbf{y}(k) = \mathbf{x}(k)$, and the output equation is omitted. The goal is to design a state feedback control law

$$\mathbf{u}(k) = g\left(\mathbf{x}(k), k\right), \tag{2}$$

such that a fixed point of the unknown map f, in particular the origin, is stabilized.

Since f is unknown, the control law, g, can only be designed based on f' – a model of f. An evolving granular model f' constructed from a stream of state data is considered. As f is time varying in general, the model f' must be provided with mechanisms for learning from stream data $\mathbf{x}(k)$, $k = 1, ...$ In other words, f' must be evolved over time to track the behavior of f. It is expected that f is stabilized by the control $\mathbf{u}(k)$ if f' is a relatively accurate approximation of f, and g is an appropriately designed control law.

A solution $\mathbf{x}^*(k)$ of (1), with initial condition $\mathbf{x}^*(0)$, is called chaotic if it is Lyapunov unstable and all the solutions starting from some neighborhood of $\mathbf{x}^*(0)$ are bounded on $(-\infty, \infty)$. Stabilizing $\mathbf{x}^*(k)$ means to drive the states $\mathbf{x}(k)$ to $\mathbf{x}^*(k)$, i.e.,

$$\lim_{k\to\infty} (\mathbf{x}(k) - \mathbf{x}^*(k)) = 0. \tag{3}$$

Figure 1 shows the closed-loop system with the state-space evolving granular model, SS-EGM, playing the role of f', and its associated online learning algorithm; and the evolving granular controller, SS-EGC, playing the role of g, and its associated online LMI-based design method. In the figure, $\tilde{\mathbf{x}}(k+1)$ is a one-step estimate of the states; and z^{-1} is a unit delay. The *unknown nonlinear time-varying system*, f, is a general term that extends to phenomena, processes, machines, robots, and a variety of virtual and real-world systems that can be controlled. We access, learn, and change the dynamics of f from a stream of state data.

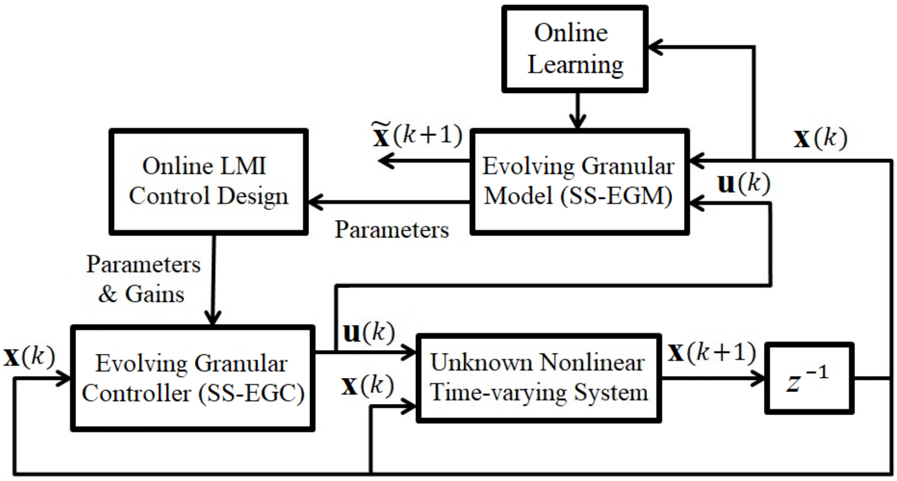
Online Learning
$\tilde{\mathbf{x}}(k+1)$
$\mathbf{x}(k)$
Online LMI Control Design
Evolving Granular Model (SS-EGM)
$\mathbf{u}(k)$
Parameters
Parameters & Gains
$\mathbf{x}(k)$
Evolving Granular Controller (SS-EGC)
$\mathbf{u}(k)$
$\mathbf{x}(k)$
Unknown Nonlinear Time-varying System
$\mathbf{x}(k+1)$
z^{-1}

Figure 1: State-space evolving granular control system

2. State-Space Evolving Granular Model

A discrete state-space fuzzy variety of evolving granular model is described. SS-EGM learning emphasizes data coverage and memory of the past. The model aims to assist online LMI-based control design for parallel distributed compensation.

2.1. Basic framework

We give an online learning algorithm to build a granular model with state-space consequent from a stream of data generated by an unknown dynamical system. In general, the original system equations are unknown, nonlinear and time-varying. A finite number of past states

$\mathbf{x}(k), \mathbf{x}(k-1), ..., \mathbf{x}(k-d)$; control inputs $\mathbf{u}(k), \mathbf{u}(k-1), ..., \mathbf{u}(k-v)$; and other external variables can be considered attributes. Nonetheless, we use descriptors of granules of the form

R^i: IF $x_1(k)$ is $\mathscr{M}_1^i$ AND ... AND $x_\Psi(k)$ is $\mathscr{M}_\Psi^i$
THEN $\mathbf{x}^i(k+1) = A^i\mathbf{x}(k) + B^i\mathbf{u}(k)$

in which $\mathbf{x}(k) = [x_1(k) \; ... \; x_\psi(k) \; ... \; x_\Psi(k)]^T$, and $\mathbf{u}(k) = [u_1(k) \; ... \; u_\phi(k) \; ... \; u_\Phi(k)]^T$. A variable amount of rules R^i, $i = 1, ..., c$, describes smooth granules boundaries and the granules content. In evolving modeling, A^i is a matrix of appropriate dimension with variable coefficients; and $\mathscr{M}_\psi^i$, $\psi = 1, ..., \Psi$, are normal (height 1) trapezoidal membership functions defined by strictly increasing parameters, $\mathscr{M}_\psi^i = (l_\psi^i, \lambda_\psi^i, \Lambda_\psi^i, L_\psi^i)$, which are updated in light of changes of the system states reflected on the data stream. We assume $B^i = [1 \; 0 \; ... \; 0]^T$ common to all rules, i.e., the control input is applied to x_1, without loss of generality. Superscript i on the left-hand side of the consequent equation refers to a local one-step estimate.

Affine terms are added to consequent matrices and state and control vectors as follows,

$$\widetilde{A}^i = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{a}_0^i & A^i \end{bmatrix}, \; \tilde{B}^i = \begin{bmatrix} 0 \\ B^i \end{bmatrix}, \; \tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}, \; \widetilde{\mathbf{u}} = \begin{bmatrix} u_0 \\ \mathbf{u} \end{bmatrix}, \tag{4}$$

in which $\mathbf{a}_0^i = [a_{10}^i \; ... \; a_{\psi 0}^i \; ... \; a_{\Psi 0}^i]^T$; and u_0 is the controller offset term, which is obtained straightforwardly from the design method (Section 3). Granule descriptors are rewritten as

R^i: IF $x_1(k)$ is $\mathscr{M}_1^i$ AND ... AND $x_\Psi(k)$ is $\mathscr{M}_\Psi^i$
THEN $\tilde{\mathbf{x}}^i(k+1) = \widetilde{A}^i\tilde{\mathbf{x}}(k) + \tilde{B}^i\widetilde{\mathbf{u}}(k)$

We omit '(k)' from the time-varying membership functions $\mathscr{M}_\psi^i$ and system matrices A^i.

The overall state estimate of the granular model is given by

$$\tilde{\mathbf{x}}(k+1) = \sum_{i=1}^{c} \mu^{ir}\tilde{\mathbf{x}}^i(k+1), \tag{5}$$

in which μ^{ir} is the rescaled activation degree of the i-th rule,

$$\mu^{ir} = \frac{\mu^i}{\sum\limits_{i=1}^{c} \mu^i}, \text{ so that } \mu^{ir} \geq 0 \text{ and } \sum_{i=1}^{c} \mu^{ir} = 1. \tag{6}$$

Rules activation degrees μ^i, $i = 1, ..., c$, can be determined by any T-norm aggregation [33], i.e., $\mu^i = T(\mu_1^i, ..., \mu_\Psi^i)$, in which μ_ψ^i is the membership degree of $x_\psi(k)$ in $\mathscr{M}_\psi^i$. Nonetheless, we obtain μ^i from a similarity measure as follows.

Let the expansion region of $\mathscr{M}_\psi^i$ be

$$E_\psi^i = [L_\psi^i - \rho, \; l_\psi^i + \rho], \tag{7}$$

in which ρ is the maximum width that $\mathscr{M}^i_\psi$, $\forall\psi, i$, is allowed to expand to enclose a datum x_ψ. In other words, $L^i_\psi - l^i_\psi \leq \rho$, $\forall\psi, i$, at any time step k.

Define $\mathscr{M}^i = [\mathscr{M}^i_1 \; ... \; \mathscr{M}^i_\Psi]^T$ as a Ψ-dimensional information granule within the state space of a dynamical system. $\mathscr{M}^i$ is obtained from the cylindrical extension of its components across the state space. $\mathbf{E}^i = [E^i_1 \; ... \; E^i_\Psi]^T$ is the expansion region of the granule $\mathscr{M}^i$. Geometrically, $\mathbf{E}^i$ is a hyperbox enclosing the support of $\mathscr{M}^i$, that is, $\text{supp}(\mathscr{M}^i) \subseteq \mathbf{E}^i$.

Given a numerical instance $\mathbf{x}$ at instant k, the activation degree of the i-th granule is $\mu^i = S(\mathbf{x}, \mathscr{M}^i)$ if $\mathbf{x} \in \mathbf{E}^i$ – being $S(.)$ a similarity measure; otherwise $\mu^i = 0$. We use

$$S(\mathbf{x}, \mathscr{M}^i) = 1 - \frac{1}{6\Psi} \sum_{\psi=1}^{\Psi} (|x_\psi - l^i_\psi| + 2|x_\psi - \lambda^i_\psi| + 2|x_\psi - \Lambda^i_\psi| + |x_\psi - L^i_\psi|). \tag{8}$$

The value of S is equal to 1 – indicating maximum activation – if the trapezoids $(l^i_\psi, \lambda^i_\psi, \Lambda^i_\psi, L^i_\psi)$, $\forall\psi$, are degenerated in singletons, and match $\mathbf{x}$. The similarity reduces linearly as $\mathbf{x}$ withdraws from $\mathscr{M}^i$ in any dimension. In particular, the summatory function in Eq. (8) determines a Hamming-like distance [11, 34].

In online environment no granule exists *a priori.* Model granules are created and updated to cover the state space. The amount of granules, c, increases by a unit if $\mu^i = 0$, $\forall i$. In this case, $\mu^{c+1} = 1$, i.e., the new granule matches the instance responsible for its own creation. A granular structure and a granular input-output map evolve without human in the loop.

2.2. Online learning

An SS-EGM is generally built from scratch. The model retains new patterns if changes or new behaviors arise in the flowing data. We describe a learning method that deals with time-varying nonlinear dynamical systems and avoids time-consuming batch training – common to conventional offline and some online window-based learning methods.

Expansion regions, $\mathbf{E}^i$, (7), are essential to the decision on whether or not a new instance belongs to a granule $\mathscr{M}^i$. Different values of the hyperparameter $\rho \in (0, 1)$ produce different granular representations of the same dynamical system. The lower the value of ρ, the larger the amount of descriptors, and the greater the details the model can seize. However, if ρ approaches 0, a granule is created for each non-coincident instance, and the model becomes excessively complex. In contrast, if ρ is equal to 1, a single granule covers all instances in the unit hypercube (rescaled data), which is insufficient to nonlinear modeling and control.

A granule is created if one or more entries of $\mathbf{x}(k)$ do not belong to the expansion regions $\mathbf{E}^i$ of $\mathscr{M}^i$, $\forall i$, $i = 1, ..., c$. The new granule $\mathscr{M}^{c+1}$ is built from trapezoidal fuzzy sets $\mathscr{M}^{c+1}_\psi$, $\psi = 1, ..., \Psi$, whose parameters match $\mathbf{x}(k)$,

$$\mathscr{M}^{c+1}_\psi = (l_\psi, \lambda_\psi, \Lambda_\psi, L_\psi)^{c+1} = (x_\psi, x_\psi, x_\psi, x_\psi). \tag{9}$$

This is a bottom-up procedure since a granule starts as a point, and tends to enlarge shortly afterwards. The opposite – top-down procedure – is also possible, that is, a granule can be initialized as large as possible based on ρ, and shrink gradually.

Subsequently, when $\mathbf{x}(k+1)$ is available, a supervised learning step is given by considering the input-output pair, $(\mathbf{x}(k), \mathbf{x}(k+1))$, and the Recursive Least Squares method [21, 24]. Thus,

$$\widetilde{A}^{c+1} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{a}_0^{c+1} & A^{c+1} \end{bmatrix} \text{ and } \tilde{B}^{c+1} = \begin{bmatrix} 0 \\ B^{c+1} \end{bmatrix}, \tag{10}$$

with offset coefficients initialized as

$$a_{\psi 0}^{c+1} = \frac{x_\psi(k+1)}{x_\psi(k)}, \;\; \psi = 1, ..., \Psi. \tag{11}$$

$B^{c+1} = [1\ 0\ ...\ 0]^T$ is constant.

Updating a granule $\mathscr{M}^{i^*}$ consists in enlarging the support $[l_\psi^{i^*}, L_\psi^{i^*}]$ and updating the core $[\lambda_\psi^{i^*}, \Lambda_\psi^{i^*}]$ of its components. Among all granules $\mathscr{M}^i$ that can be expanded to include an $\mathbf{x}(k)$, that with the highest similarity, $\mathscr{M}^{i^*}$, in which

$$i^* = arg \max_{i=1,...,c} (S(\mathbf{x}, \mathscr{M}^i)) \tag{12}$$

is chosen. $S(.)$ is obtained from (8). Granular coverage of the space of states and model memory to support control design are emphasized by: (i) avoiding typical granule deleting and merging procedures [35, 36]; (ii) keeping the model granularity ρ constant; and (iii) considering the following updating relations, which keep granules centers static:

$$\text{If} \quad x_\psi(k) \in [L_\psi^{i^*} - \rho, l_\psi^{i^*}] \quad \text{then} \quad l_\psi^{i^*}(k+1) = x_\psi(k) \tag{13}$$

$$\text{If} \quad x_\psi(k) \in [L_\psi^{i^*}, l_\psi^{i^*} + \rho] \quad \text{then} \quad L_\psi^{i^*}(k+1) = x_\psi(k) \tag{14}$$

$$\text{Otherwise,} \quad l_\psi^{i^*}(k+1) = l_\psi^{i^*}(k) \text{ and } L_\psi^{i^*}(k+1) = L_\psi^{i^*}(k) \tag{15}$$

for $\psi = 1, ..., \Psi$. Core parameters are updated from

$$\lambda_\psi^{i^*}(k+1) = \Lambda_\psi^{i^*}(k+1) = \frac{l_\psi^{i^*}(k+1) + L_\psi^{i^*}(k+1)}{2}. \tag{16}$$

The learning method creates a new granule $\mathscr{M}^{c+1}$ or adapts the parameters of $\mathscr{M}^{i^*}$, accordingly. The Recursive Least Squares method updates $\widetilde{A}^{i^*}$ [21, 24].

3. Control Design

From the granular state-space model, SS-EGM, we formulate conditions for closed-loop Lyapunov stability and bounded control inputs as an LMI feasibility problem. The gain matrices of the granular controller, SS-EGC, are derived automatically from the solution of a relaxed LMI solved only for active granules in a time step. Some SS-EGC local gains are quickly redesigned when their corresponding SS-EGM granules change.

3.1. Closed-loop system

We consider parallel distributed compensation, i.e., evolving granular model and controller share the same attributes. A control rule is given by

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

in which $\tilde{\mathbf{x}}(k) = [1\ x_1(k)\ ...\ x_\Psi(k)]^T$ and $\widetilde{\mathbf{u}}(k) = [u_0(k)\ u_1(k) ...\ u_\Phi(k)]^T$. $K^i \in \mathfrak{R}^{(\Phi+1\times\Psi)}$ is a gain matrix – with offset terms in the first column – to be determined in order to make the closed loop system asymptotically stable and/or to drive the states faster and smoother to a reference. Superscript i on the left-hand side of the consequent state feedback law means a local control input. The overall control signal, effectively sent to actuators, is

$$\widetilde{\mathbf{u}}(k+1) = \sum_{i=1}^{c} \mu^{ir}\widetilde{\mathbf{u}}^i(k+1), \tag{17}$$

in which μ^{ir} is the rescaled activation degree (6).

Having model and controller c rules, (5) and (17) combined yield the closed-loop system

$$\tilde{\mathbf{x}}(k+1) = \sum_{i=1}^{c}\sum_{j=1}^{c} \mu^{ir}\mu^{jr}G^{ij}\tilde{\mathbf{x}}(k), \tag{18}$$

in which $G^{ij} := \widetilde{A}^i + \tilde{B}^i K^j$, or, equivalently,

$$\tilde{\mathbf{x}}(k+1) = \sum_{i=1}^{c}(\mu^{ir})^2 G^{ii}\tilde{\mathbf{x}}(k) + 2\sum_{i<j}^{c} \mu^{ir}\mu^{jr}\left(\frac{G^{ij}+G^{ji}}{2}\right)\tilde{\mathbf{x}}(k). \tag{19}$$

If the unforced system is stable, K^i $\forall i$ may be used to improve the transient response. Unstable systems require suitable K^i's for stabilization primarily. An issue in online environment is that granules $\mathscr{M}^i$, system matrices A^i, and the number of rules c, are time-varying. Local K^i's should be reviewed after changes of the corresponding model granules.

3.2. Lyapunov stability

A fuzzy Lyapunov function is a fuzzy combination of quadratic functions of the system states,

$$V(\mathbf{x}) = \sum_{i=1}^{c} \mu^{ir}\mathbf{x}^T P^i \mathbf{x}, \tag{20}$$

in which $P^i > 0$, $\forall i$ (positive definiteness). A stabilization result for the closed-loop system (18) based on (20) is as follows [37].

Result: The system (18) is asymptotically stable if there exist positive definite matrices $X^i = (P^i)^{-1}$ and matrices Q^i and Z^i, $i = 1, ..., c$, such that

$$\begin{bmatrix} X^i - (Z^j)^T - Z^j & (Z^j)^T (A^i)^T + (Q^j)^T (B^i)^T \\ A^i Z^j + B^i Q^j & -X^k \end{bmatrix} < 0 \tag{21}$$

holds true for all combinations of $i, j, k = 1, ..., c$. See [37] for a proof. If a feasible solution for (21) is found, the controller gains are

$$K^j = Q^j (Z^j)^{-1}, \; j = 1, ..., c. \tag{22}$$

The number of granules affects the complexity of the LMI analysis. Finding a Lyapunov function for a large 'c' may be difficult [19, 21, 26]. In view of the features of the SS-EGM learning method, namely, inactive granules do not change, then, recalculation of gains (22) is needed only for the active granules at a time step. Therefore, the number of rows and columns of (21) can be greatly reduced by considering active granules only.

Definition: The number of active granules, $\mathfrak{x}$, for an instance $\mathbf{x}(k)$ is equal to the number of terms that make the rescaled activation degree $\mu^{ir}(\mathbf{x}(k)) > 0, i = 1, ..., c$.

Additionally, we reduce the number of concatenated rows in (21) by making matrices Z^j $\forall j$ and X^k $\forall k$ equal to X^i $\forall i$, with X^i symmetric, to obtain a theorem.

Theorem: The closed-loop system (18) is asymptotically stable if there are positive definite matrices $X^i = (P^i)^{-1}$, and matrices Q^j; $i, j = 1, ..., \mathfrak{x}$, such that

$$\begin{bmatrix} -X^i & X^i (A^i)^T + (Q^j)^T (B^i)^T \\ A^i X^i + B^i Q^j & -X^i \end{bmatrix} < 0 \tag{23}$$

are satisfied for all combinations of $i, j = 1, ..., \mathfrak{x}$ – being $\mathfrak{x}$ the number of active granules for an instance $\mathbf{x}(k)$.

If a feasible solution to (23) is found, then the control gains assigned to the active control granules are redefined as

$$K^i = Q^j P^i, \quad i, j = 1, ..., \mathfrak{x}. \tag{24}$$

The gains related to inactive granules are kept as computed in previous time steps.

The feasibility problem (23) is dynamic and convex. Finding a solution means that (20) is Lyapunov for $i = 1, ..., \mathfrak{x}$; and (18) is stable using K^i (24). The proof of the Theorem follows analogously to that in the appendix of [21]. Efficient LMI parser and solver, Yalmip'18 [28], and Mosek'20 [29], are available.

3.3. Bounded input

Being the state $\mathbf{x}(k)$ known, the following result applies for a bounded control input [19, 21].

Result: Given positive definite matrices X^i, and matrices $Q^j = K^i X^i$, $i = 1, ..., \mathfrak{c}$, as in (24). The constraint $||\mathbf{u}(k+1)||_2 \leq \zeta$ is enforced if

$$\begin{bmatrix} 1 & \mathbf{x}(k)^T \\ \mathbf{x}(k) & X^i \end{bmatrix} > 0 \text{ and } \begin{bmatrix} X^i & (Q^j)^T \\ Q^j & \zeta^2 I \end{bmatrix} > 0 \qquad (25)$$

hold true for $i, j = 1, ..., \mathfrak{c}$. See [19] for a proof. We replaced the initial state $\mathbf{x}(0)$ in (25) by the current state $\mathbf{x}(k)$ for online design. Parameter ζ bounds the maximum input, thus keeping it within the operation range of actuators. The equations (25) are appended to (23) for a stable granular controller satisfying input constraints.

4. Control of Chaotic Map: Result and Discussion

The effectiveness of the SS-EGM and associated SS-EGC is assessed through a deterministic chaotic system, the Henon map [30]. We use the Henon equations to generate a data stream. State-space granular modeling and control are performed on the fly.

4.1. Henon map

The nonlinear equations of the Henon map are:

$$\begin{aligned} x_1(k+1) &= x_2(k) + 1 - \alpha x_1(k)x_1(k) \\ x_2(k+1) &= \beta x_1(k). \end{aligned} \qquad (26)$$

The phase portrait for $\alpha = 1.4$, $\beta = 0.3$, and initial state $\mathbf{x}(0) = [1 \;\; 0]^T$ is shown in Figure 2. In this case, $[0.6314 \;\; 0.1894]$ and $[-1.1314 \;\; -0.3394]$ are the fixed points of the map. In general, if the orbit or trajectory spreads over the phase plane, we have a stochastic process. As we see a deterministic orbit, then, we have chaos. In fact, the Henon map is a model of the Poincaré section of the Lorenz system. Notice that the orbit of the unforced system settles into an irregular oscillation – confined in a fractal set – which never repeats exactly.

Aiming at stabilizing (26), i.e., driving the chaotic orbit $\mathbf{x}(k)$ to the origin $\mathbf{x}^* = [0 \;\; 0]$, $\lim_{k \to \infty}(\mathbf{x}(k) - \mathbf{x}^*) = 0$, we add a control input, $u(k)$, to the first relation. Thus,

$$\begin{aligned} x_1(k+1) &= x_2(k) + 1 - 1.4x_1(k)x_1(k) + u(k) \\ x_2(k+1) &= 0.3x_1(k). \end{aligned} \qquad (27)$$

The online model-based granular controller is designed to opportunely inject values $u(k)$ to lead the states $\mathbf{x}(k)$ to the origin $\mathbf{x}^*$. The Henon system as in (27) is completely controllable and observable. The states are considered measurable.

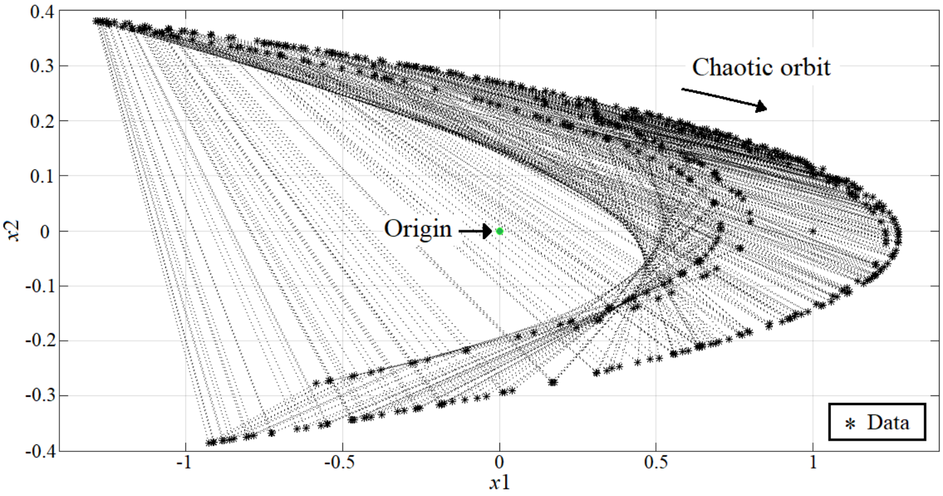

Figure 2: Phase plane of the Henon chaotic map (26)

4.2. Online modeling and one-step estimation

Consider the controller, SS-EGC, off for k_f time steps. Then, $u(k) = 0,\ k = 1, ..., k_f$. We want to assess the accuracy of the model, SS-EGM, in one-step prediction. The model is built from scratch, with no knowledge about the dynamic system that produces the data stream.

Let the root mean square error be

$$\text{RMSE} = \frac{1}{k_f}\sum_{k=1}^{k_f}\sqrt{\sum_{j=1}^{n}\left(x_j(k+1) - \tilde{x}_j(k+1)\right)^2}, \tag{28}$$

in which $\mathbf{x}(k+1)$ is the actual value given by (27); and $\tilde{\mathbf{x}}(k+1)$, given by (5), is the SS-EGM estimate. Table 1 shows the one-step prediction results for $k_f = 500$, and different maximum widths, ρ, allowed for the model granules.

Table 1
One-step SS-EGM prediction based on different model granularities

Granularity (ρ)	Final Structure (# granules)	RMSE
0.1	60	0.0305
0.2	32	0.0317
0.3	20	0.0359
0.4	16	0.0442
0.5	10	0.0445
0.6	9	0.0478
0.7	7	0.0487
0.8	5	0.0573

We notice from Table 1 that the number of resulting granules and, therefore, the number of SS-EGM parameters, reduces with the increasing of ρ, which facilitates further control design. However, the SS-EGM accuracy reduces, and, in this case, the controller is designed for a worse picture of the original system, which may lead worse-than-expected transient responses and instability. A trade-off between accuracy and model compactness is observed.

Figure 3 shows how the state space is granulated for $\rho = \{0.1, 0.2, 0.3, 0.5\}$, yielding more detailed or coarser fuzzy partitions. Notice in the last of the graphs how wider supports of fuzzy granules (larger boxes) start to embrace data on both upper and lower parts of the Henon moon so that locally-valid estimates given by linear difference equations start to get confused with respect to a positive or negative slope. As a consequence, some red dots (one-step prediction) are noticed a bit farther from the black dots (actual values), specially in the middle, more curvy, part of the moon. Figure 4 shows the SS-EGM structural evolution. The smaller the support of the fuzzy granules, the more accurate the local model, but the more complex the LMI to be solved for further controller adaptation.

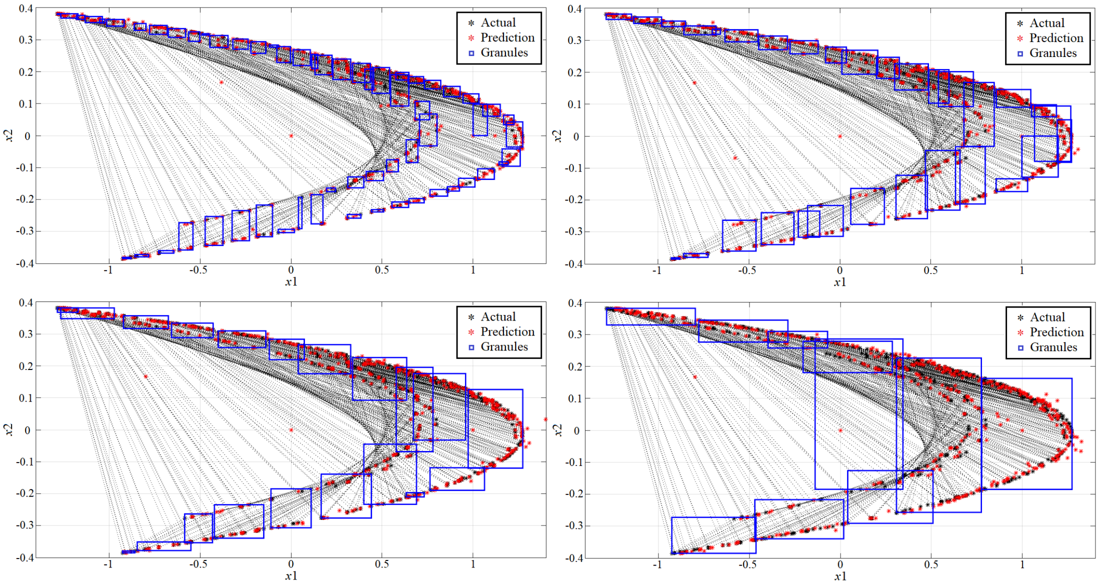

Figure 3: SS-EGM partitions of the Henon map using 60, 32, 20, and 10 granules

As an example, the parameters of an SS-EGM descriptor, using $\rho = 0.3$ at $k = 500$, are:

Granule 1:

$\mathscr{M}_1^1 = (0.9382, 1.1437, 1.1437, 1.3491)$
$\mathscr{M}_2^1 = (-0.1194, 0.0046, 0.0046, 0.1285)$

$$A^1 = \begin{bmatrix} 1 & 0 & 0 \\ -0.3484 & -0.6542 & 0.7188 \\ 16.0058 & 0.4261 & 0.0800 \end{bmatrix} \tag{29}$$

Vector $B^1 = [0\ 1\ 0]^T$, which is common to all descriptors.

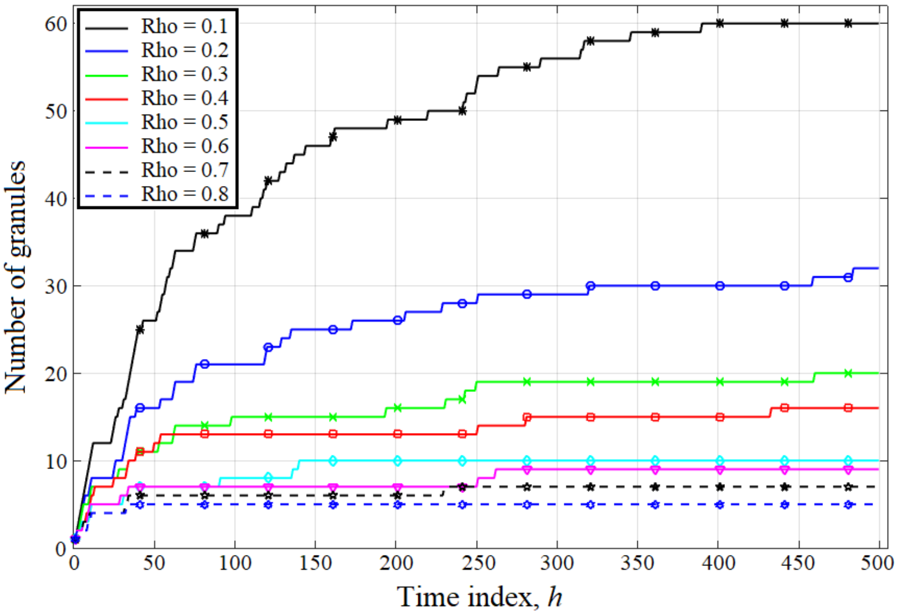

Figure 4: SS-EGM structural evolution for different model granularities ρ

4.3. Closed-loop stabilization

The granular controller, SS-EGC, is designed based on the model SS-EGM. Nonetheless, the output of the SS-EGC is applied to the original Henon system (27). Thus, $\mathbf{x}$ is the actual system state, i.e., we want to control a completely *unknown* chaotic system on the fly.

Define the settling time, t_s, as the number of time steps from the application of the control input, u, until all states, $\mathbf{x}$, enter and remain within a 2% band around the origin, $\mathbf{x}^*$. The range of values of the system (26) are $x_1 \in [-1.2836,\ 1.2727]$ and $x_2 \in [-0.3851,\ 0.3818]$. Thus, $x_1(k) \leq x_{1(2\%)} = 0.0256$, and $x_2(k) \leq x_{2(2\%)} = 0.0077$, $\forall k$, after $u \neq 0$, determines t_s.

Let the energy $E_\mathbf{x}$ of the states $\mathbf{x}$, for $u \neq 0$, be

$$E_\mathbf{x} = \langle \mathbf{x}(k), \mathbf{x}(k) \rangle = \sum_{j=1}^{n} \sum_{k_{u\neq 0}} |x_j(k)|^2. \tag{30}$$

Particularly, $n = 2$ is the number of states; and $E_\mathbf{x}$ is a measure of the energy of the closed-loop system during the stabilization transient. We turned the controller on at $k = 500$, and off at $k = 600$. The lower the values of t_s and $E_\mathbf{x}$, the better – as this implies that the chaotic behavior was dissipated faster. Table 2 summarizes the SS-EGC closed-loop results for different granularities, ρ, and upper bounds on the control input, ζ.

We notice from Table 2 that an intermediate granularity, $\rho = 0.3$, and a wider operation range for inputs, $\zeta = 1.5$, provide the best closed-loop performance. As expected, larger values of ρ cause higher modeling error, such that the controller is designed based on a less accurate model. Therefore, in cases such as for the granularities 0.5 and 0.6, despite converging evidence, the

Table 2
SS-EGC results on the stabilization of the Henon chaos

Granularity ρ	Upper bound ζ	Steps for settling $[k]$	Energy $E_{\mathbf{x}}$
0.1	1.5	21	1.9574
0.2	1.5	23	1.9629
0.3	1.5	**17**	**1.5953**
0.4	1.5	85	1.9152
0.5	1.5	39	1.9131
0.6	1.5	61	2.7607
0.1	1.1	53	2.1995
0.2	1.1	53	2.1264
0.3	1.1	77	2.0304
0.4	1.1	46	2.4073
0.5	1.1	–	2.5735
0.6	1.1	–	3.8334

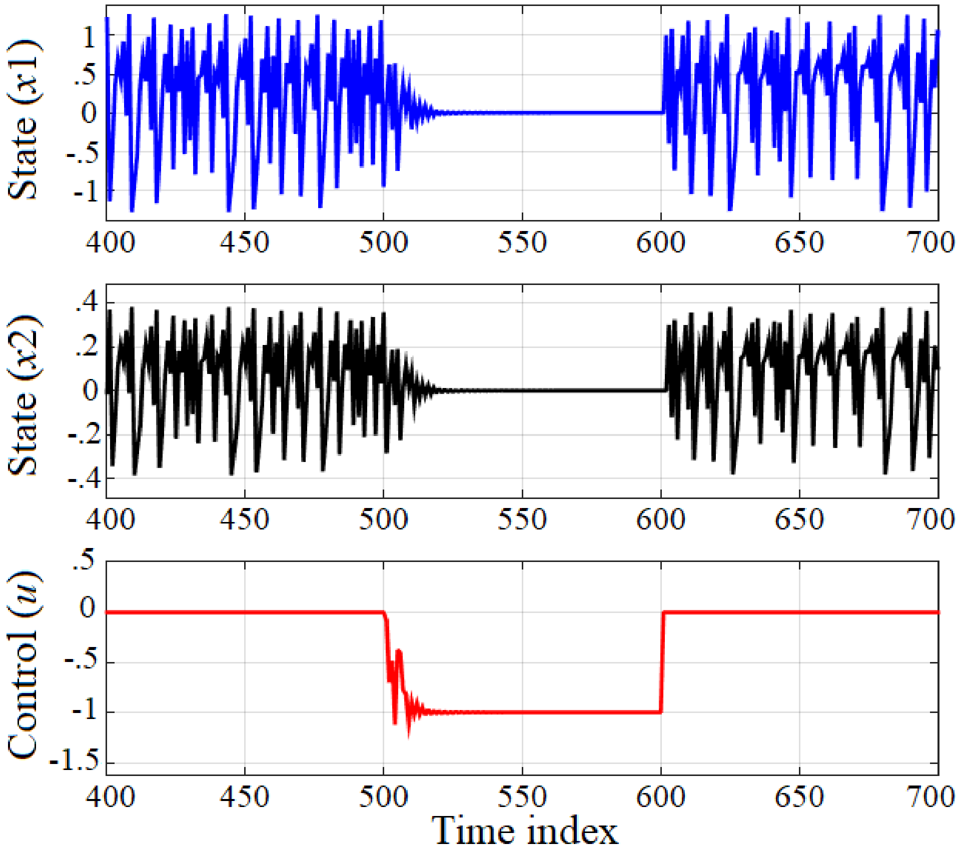

Figure 5: SS-EGC regularization of the Henon chaos. Example using $\rho = 0.3$ and $\zeta = 1.5$. The controller is designed and updated from $k = 500$ to $k = 600$ based on the SS-EGM

states still spiral on a band larger than 2% around $\mathbf{x}^*$ after the 100 time steps with $u \neq 0$. To illustrate the regularization of the chaotic behavior, Figure 5 shows the states convergence when u is enabled at $k = 500$. Figure 6 highlights the asymptotic convergence to the origin $(0, 0)$ from the phase plane perspective. Notice that the Henon states backtrack to their chaotic orbit when the control is turned off at $k = 600$.

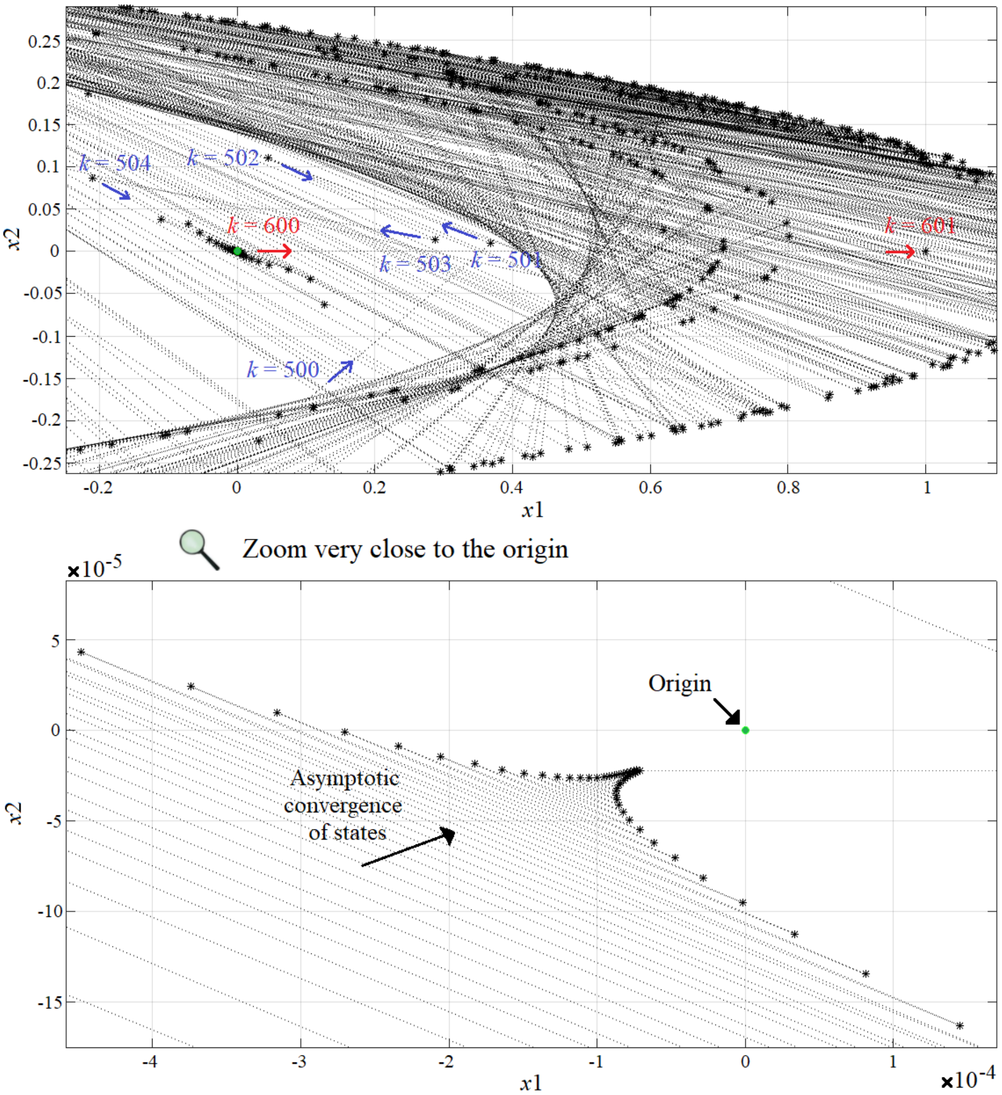

Figure 6: Phane plane perspective of the result in Figure 5 emphasizing the convergence of the Henon states toward the origin $(0, 0)$ in a spiral-shaped trajectory. Control is applied at $k = 500$. The blue arrows show the deviation of the states from the usual unforced trajectory. The system backs to the chaotic orbit at $k = 600$ when the evolving controller is turned off (red arrows)

5. Conclusion

We described an evolving state-space variation of granular method to model and control unknown nonlinear dynamical systems. The model SS-EGM and the controller SS-EGC have their structures developed from scratch and their parameters updated over time. The online learning method is particularly devoted to cover the data space and keep memory of past patterns. Control gains are derived from a relaxed LMI feasibility problem and guarantee closed-loop Lyapunov stability and bounded control inputs from parallel distributed compensation. Bounded inputs avoid actuator saturation.

We have shown asymptotic stabilization of the Henon chaos as an example. The results remark a relatively low RMSE in one-step SS-EGM prediction, 0.0359; and a settling time of 17 steps using the original Henon equations in the loop and an intermediate granularity, 0.3. The Henon equations are assumed unknown for any modeling and control design purpose. SS-EGM notices the dynamic system by means of the data stream only. In the future we envision a diversity of applications, such as on secure communication, artifact suppression in physiological signals, and control of nonlinear time-varying systems in general, such as mobile robots and unmanned aerial vehicles, to mention some. We should also address trajectory following control and LMI-optimal design issues. A higher level of autonomy, structural adaptability, human-level interpretability, and the possibility to deal with heterogeneous data streams have been achieved by means of evolving granular computing.

References

[1] D. Leite, Evolving Granular Systems, Ph.D. thesis, School of Electrical and Computer Engineering, University of Campinas, 2012.

[2] D. Leite, G. Andonovski, I. Skrjanc, F. Gomide, Optimal rule-based granular systems from data streams, IEEE Transactions on Fuzzy Systems 28 (2020) 583–596.

[3] C. Garcia, D. Leite, I. Škrjanc, Incremental missing-data imputation for evolving fuzzy granular prediction, IEEE Transactions on Fuzzy Systems 28 (2020) 2348–2362.

[4] L. Decker, D. Leite, F. Viola, D. Bonacorsi, Comparison of evolving granular classifiers applied to anomaly detection for predictive maintenance in computing centers, in: IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Bari, 2020, p. 8p.

[5] D. Wang, W. Pedrycz, Z. Li, Granular data aggregation: An adaptive principle of the justifiable granularity approach, IEEE Transactions on Cybernetics 49 (2019) 417–426.

[6] L. Zadeh, Generalized theory of uncertainty (gtu) - principal concepts and ideas, Computational Statistics and Data Analysis 51 (2006) 15–46.

[7] W. Pedrycz, A. Skowron, V. K. (Eds.), Handbook of Granular Computing, Wiley, 2008.

[8] W. Pedrycz, F. Gomide, Fuzzy systems engineering: toward human-centric computing, John Wiley & Sons, 2007.

[9] Y. Yao, Perspectives of granular computing, in: IEEE International Conference on Granular Computing, 2005, pp. 85–90.

[10] L. A. Q. Cordovil, P. H. S. Coutinho, I. V. de Bessa, M. F. S. V. D'Angelo, R. M. Palhares,

Uncertain data modeling based on evolving ellipsoidal fuzzy information granules, IEEE Transactions on Fuzzy Systems 28 (2020) 2427–2436.

[11] D. Leite, R. Ballini, P. Costa, F. Gomide, Evolving fuzzy granular modeling from nonstationary fuzzy data streams, Evolving Systems 3 (2012) 65–79.

[12] C. Aguiar, D. Leite, Unsupervised fuzzy eix: Evolving internal-external fuzzy clustering, in: IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Bari, 2020, p. 8p.

[13] L. Decker, D. Leite, D. Bonacorsi, Explainable log parsing and online interval granular classification from streams of words, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Padua, 2022, p. 8p.

[14] D. Leite, L. Decker, M. Santana, P. Souza, Egfc: Evolving gaussian fuzzy classifier from never-ending semi-supervised data streams – with application to power quality disturbance detection and classification, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Glasgow, 2020, p. 9p.

[15] D. Leite, P. Costa, F. Gomide, Evolving granular neural network for semi-supervised data stream classification, in: International Joint Conference on Neural Networks (IJCNN), Barcelona, 2010, p. 8p.

[16] L. Decker, D. Leite, L. Giommi, D. Bonacorsi, Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Glasgow, 2020, p. 8p.

[17] R.-E. Precup, T.-A. Teban, A. Albu, A.-B. Borlea, I. A. Zamfirache, E. M. Petriu, Evolving fuzzy models for prosthetic hand myoelectric-based control, IEEE Transactions on Instrumentation and Measurement 69 (2020) 4625–4636.

[18] A. Nguyen, T. Taniguchi, L. Eciolaza, V. Campos, R. Palhares, M. Sugeno, Fuzzy control systems: Past, present and future, IEEE Computational Intelligence Magazine 14 (2019) 56–68.

[19] K. Tanaka, H. O. Wang, Fuzzy control systems design and analysis: a linear matrix inequality approach, John Wiley & Sons, 2004.

[20] C. Aguiar, D. Leite, D. Pereira, G. Andonovski, I. Skrjanc, Nonlinear modeling and robust lmi fuzzy control of overhead crane systems, Journal of the Franklin Institute 358 (2021) 1376–1042.

[21] D. Leite, R. M. Palhares, V. C. Campos, F. Gomide, Evolving granular fuzzy model-based control of nonlinear dynamic systems, IEEE Transactions on Fuzzy Systems 23 (2015) 923–938.

[22] A. Bento, L. Oliveira, I. Scola, V. Leite, F. Gomide, Evolving granular control with high-gain observers for feedback linearizable nonlinear systems, Evolving Systems 12 (2021) 935–948.

[23] L. Oliveira, A. Bento, V. Leite, F. Gomide, Evolving granular feedback linearization: design, analysis, and applications, Applied Soft Computing 86 (2020) 105927.

[24] K. J. Astrom, B. Wittenmark, Adaptive Control, Prentice-Hall, Addison-Wesley, Boston, 2nd edition, 1994.

[25] L. Ljung, System Identification - Theory for the User, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[26] G. Feng, Analysis and Synthesis of Fuzzy Control Systems: A Model-Based Approach, Boca Raton, FL, US: CRC, 2010.

[27] A. Sala, T. M. Guerra, R. Babuska, Perspectives of fuzzy systems and control, Fuzzy Sets

and Systems 156 (2005) 432–444.

[28] J. Lofberg, Yalmip: A toolbox for modeling and optimization in MATLAB, in: IEEE International Conference on Robotics and Automation, 2004, pp. 284–289.

[29] E. Andersen, K. Andersen, The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm, in: High Performance Optimization, Springer, 2000, pp. 197–232.

[30] M. Hénon, A two-dimensional mapping with a strange attractor, Communications in Mathematical Physics 50 (1976) 69–77.

[31] D. Leite, M. Santana, A. Borges, F. Gomide, Fuzzy granular neural network for incremental modeling of nonlinear chaotic systems, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, 2016, pp. 64–71.

[32] A. L. Fradkov, R. J. Evans, B. R. Andrievsky, Control of chaos: methods and applications in mechanics, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 364 (2006) 2279–2307.

[33] G. Beliakov, A. Pradera, T. Calvo, et al., Aggregation functions: A guide for practitioners, volume 221, Springer, 2007.

[34] V. Cross, T. Sudkamp, Similarity and compatibility in fuzzy set theory: Assessment and Applications, volume 93, Springer, 2002.

[35] I. Skrjanc, J. A. Iglesias, A. Sanchis, D. Leite, E. Lughofer, F. Gomide, Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey, Information Sciences 490 (2019) 344–368.

[36] D. Leite, I. Škrjanc, F. Gomide, An overview on evolving systems and learning from stream data, Evolving Systems 11 (2020) 181–198.

[37] G. Feng, Analysis and synthesis of fuzzy control systems: A Model-Based Approach, volume 37, CRC press, 2018.