Parameter Estimation of Dynamic Fuzzy Models from Uncertain Data Streams

Daniel Leite*, Walmir Caminhas*, Andre Lemos*, Reinaldo Palhares*, Fernando Gomide+
*Department of Electronics Engineering, Federal University of Minas Gerais, Brazil
+School of Electrical and Computer Engineering, University of Campinas, Brazil
{danfl7, gomide}@dca.fee.unicamp.br, {caminhas, andrepl, palhares}@cpdee.ufmg.br

Abstract—Modeling of time-varying dynamic systems in real time requires the use of incremental algorithms based on streams of sensor data. This paper introduces an incremental fuzzy modeling approach based on uncertain data streams. By uncertain data we mean data originated from unreliable sensors, imprecise perception, or description of the value of a variable represented as a fuzzy interval. An online incremental learning algorithm is used to develop the antecedent part of functional fuzzy rules and the rule base that assembles the model. A recursive least squares algorithm is used to develop the antecedent part of functional fuzzy rules as a fuzzy interval. An online incremental learning algorithm is proposed to take into account data uncertainty in the adjustment of consequent parameters. Section III shows an illustrative application on one-step estimation of the Lorenz system. Conclusions are given in Section IV.

I. INTRODUCTION

Most real-world processes are characterized by nonlinear behavior and uncertain, time-varying parameters that hampers the description of their dynamical behavior using differential equations derived from first principles. An evolving functional fuzzy method oriented to uncertain data streams is proposed in this paper for online modeling of time-varying nonlinear dynamical systems.

Data streams from dynamical systems are considered carrying uncertainty represented as fuzzy intervals. The idea is to guarantee the inclusion of the actual values of pointwise data into fuzzy objects independent of the different types and levels of uncertainties that may be involved in the processes of data generation, measurement and processing. Fuzzy data streams are potentially endless and may be subject to changes of various kinds. Direct application of system identification and computational intelligence methods to fuzzy data streams is very often infeasible because it is difficult to maintain all the data in memory. A challenge faced in stream modeling concerns how to handle data uncertainty [7] [10].

Uncertainty is an attribute of information since our ability to perceive reality is often limited [4] [14]. The more complex a system is, potentially, the more uncertain we are of the available information, and the more imprecise is our understanding of that system. Fuzzy granular computing [2] [9], hypothesizes that accepting some level of uncertainty may be beneficial and therefore suggests a balance between precision and uncertainty. Of concern to this paper are fuzzy interval data and discrete-time state-space fuzzy models.

The paper is structured as follows. Section II addresses an evolving modeling method capable of handling uncertain data streams. The evolving method: learns continuously from numerical or fuzzy data; does not store previous samples; does not depend upon prior structural knowledge; self-adapts its structure when needed; is independent of statistical properties of data; and does not use ‘prototype’ initialization. A specificity-weighted recursive least squares algorithm is proposed to take into account data uncertainty in the adjustment of consequent parameters. Section III shows an illustrative application on one-step estimation of the Lorenz system. Conclusions are given in Section IV.

II. EVOLVING GRANULAR SYSTEM MODELING

A. Evolving fuzzy modeling

A functional fuzzy model of an unknown time-varying dynamic system can only be obtained from data streams. In general, a finite number of past states $x(k), x(k-1), \ldots, x(k-m)$, outputs and other exogenous variables can be considered as elements of the antecedent part of fuzzy rules. Nevertheless, a common conjecture is that rules are of the form:

$$R_i: \text{IF } x_1(k) \text{ IS } M_{i1} \text{ AND } \cdots \text{ AND } x_P(k) \text{ IS } M_{iP} \text{ THEN } x'(k+1) = A_i x(k)$$

where $x(k) = [x_1(k) \ldots x_P(k)]^T$. In evolving modeling, $A_i$ is a matrix of appropriate dimension with variable coefficients; $M_{i\psi}, \psi = 1, \ldots, \Psi$, are membership functions built in light of the data being available. The total number of rules $R_i$, $i = 1, \ldots, c$, is also variable. Superscript $i$ on the left-hand side of the consequent equation means a local estimation. We shall make the assumption that all states are available. State observers are out of the scope of this paper.

Consequent matrices $A_i$ and the state vector $x(k)$ can be extended to include affine terms as follows:

$$\tilde{A}_i = \begin{bmatrix} 1 & 0 \\ a_{i1} & A_i \end{bmatrix}, \quad \tilde{x} = \begin{bmatrix} 1 \\ x \end{bmatrix},$$

(1)

where $a_{i0} = [a_{i10} \ldots a_{i0} \ldots a_{i10}]^T$. Rules $R_i$ are rewritten as:

$$R_i: \text{IF } x_1(k) \text{ IS } M_{i1} \text{ AND } \cdots \text{ AND } x_P(k) \text{ IS } M_{iP} \text{ THEN } \tilde{x}'(k+1) = \tilde{A}_i \tilde{x}(k)$$
For clearance, in the rest of the paper we omit the ‘tilde’ from the notation and consider affine matrices and vectors. For the same reason, the time index $k$ is omitted from the time-varying membership functions $A_t^i$, and from matrices $A^i$.

The overall state estimation of the functional fuzzy model is found as the weighted mean value:

$$x(k+1) = \sum_{i=1}^{c} \mu_i^r x^i(k+1),$$

where $\mu_i^r$ is the rescaled activation degree of the $i$-th rule,

$$\mu_i^r = \frac{\mu_i}{\sum_{j=1}^{c} \mu_j}, \text{ so that } \mu_i^r \geq 0 \text{ and } \sum_{i=1}^{c} \mu_i^r = 1. \quad (3)$$

Activation degrees $\mu_i^r$ are determined using any extended conjunctive aggregation operator, e.g., a triangular norm [3], [9]. T-norms ($T$) are commutative, associative and monotone operators on the unit hypercube $[0,1]^n$ whose boundary conditions are $T(\omega, \omega, \ldots, 0) = 0$ and $T(\omega, 1, \ldots, 1) = \omega, \omega \in [0,1]$. The neutral element of T-norms is $e = 1$. In this work we shall adopt the product operator, then

$$\mu_i^r = \prod_{\psi=1}^{\psi} \mu_i^\psi. \quad (4)$$

While many works assume that the activation degree $\mu_i^r$ 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 data domain. The number of rules $c$ increases by a unit whenever, e.g., $\mu_i^r = 0 \forall i$. In this case, $\mu_i^{c+1} = 1$, 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 [7], [10].

Trapezoidal fuzzy data and models are of concern to this study. A generic trapezoidal fuzzy set $N = (l, \lambda, \Lambda, L)$ allows the modeling of a wide class of granular objects [11]. A triangular fuzzy set is a trapezoid where $\lambda = \Lambda$; an interval is a trapezoid where $l = \lambda$ and $\Lambda = L$; a singleton is a trapezoid where $l = \lambda = \Lambda = L$. Additional features that make the trapezoidal representation attractive comprise: (i) ease of acquiring the necessary parameters: only four parameters need to be captured. A trapezoidal model is formed straightforwardly from a trapezoidal datum; and (ii) 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 $N : X \to [0,1]$ is upper semi-continuous if the set $\{x \in X | \mu(x) > \alpha\}$ is closed, that is, if the $\alpha$-cuts of $N$ are closed intervals. If the universe $X$ is the set of real numbers and $N$ is normal, $\mu(x) = 1 \forall x \in [\lambda, \Lambda]$, then $N$ is a model of a fuzzy interval, with monotone increasing function $\zeta_{N} : [l, \lambda] \to [0,1]$, monotone decreasing function $\iota_{N} : [\lambda, L] \to [0,1]$, and zero otherwise [9]. A fuzzy interval $N$ has the following canonical form:

$$N : x \to \mu(x) = \begin{cases} \zeta_{N}, & x \in [l, \lambda] \\ \iota_{N}, & x \in [\lambda, L] \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

where $x$ is a real number in $X$. The fuzzy interval $N$ 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

$$\zeta_{N} = \frac{x - l}{\lambda - l} \quad \text{and} \quad (6)$$

$$\iota_{N} = \frac{L - x}{L - \lambda} \quad (7)$$

then the fuzzy interval (5) reduces to the model of a trapezoidal membership function. Moreover, when $\lambda = \Lambda$, then $\mu(x) = 1$ for a single element $x$ in $X$. In that case, the corresponding fuzzy entity is a fuzzy number.

Denote $x = (x, \underline{x}, x, \overline{x})$ as a trapezoidal datum. The membership degree of $x$ in the fuzzy set $N$ can be obtained from (5) if $x$ is degenerated in a singleton. Otherwise, if $x$ is a symmetric object, i.e. if $x - \underline{x} = \overline{x} - x \neq 0$, its membership degree in $N$ can be calculated using the midpoint:

$$mp(x) = \frac{\overline{x} + x}{2}. \quad (8)$$

The center of gravity

$$CoG(x) = \frac{\underline{x} + 5(x + \overline{x}) + \overline{x}}{12} \quad (9)$$

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

Fig. 1. Case where the membership degree of the fuzzy datum $x$ in the fuzzy model $N$ obtained by (8) is zero despite their significant similarity.
the data uncertainty into fuzzy models. Additionally, in some situations, as that shown in Fig. 1, 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 consider properly all relevant situations.

C. Similarity between fuzzy objects

Similarity measures are fundamental in the construction of rule-based systems from data. In this work, data and models are trapezoidal fuzzy objects. A possible similarity measure for trapezoids, say \( x \) and \( M^t \) is:

\[
S(x, M^t) = 1 - D(x, M^t),
\]

where \( D(x, M^t) \) is a distance obtained as follows:

\[
D(x, M^t) = \frac{|x_l - l^t| + 2|\bar{x} - \lambda^t| + 2|\bar{\bar{x}} - \Lambda^t| + |\bar{\bar{x}} - L^t|}{6},
\]

The measure \( S \) equals 1 for identical trapezoids (indicating the maximum degree of matching between them) and decreases linearly as \( x \) and \( M^t \) withdraw from each other. In particular, (11) 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 (10) - (11) are simple to compute, involving only basic arithmetic operations, there are no strong principled reasons to impose this measure. In fact, there is no generally accepted consensus on a best similarity measure [5].

Let the expansion region of a set \( M^t \) be denoted by

\[
E^t = [L^t - \rho, \bar{L}^t + \bar{\rho}],
\]

where \( \rho \) is the maximum width \( M^t \) is allowed to expand to fit a datum \( x \); \( L^t - \rho \leq \rho \) at any time. Consider the membership degree of the datum \( x \) in the fuzzy set \( M^t \) as \( \mu^t = S(x, M^t) \) if \( x \in E^t \). Contrariwise, \( \mu^t = 0 \).

A generalization of the similarity measure (10) for vectors of trapezoids, say \( x = [x_1, x_2, ..., x_{\Psi}]^T \) and \( M^t = [M^t_1, M^t_2, ..., M^t_{\Psi}]^T \), is

\[
S(x, M^t) = 1 - \frac{1}{6\Psi} \sum_{\psi=1}^{\Psi} \left( |x_\Psi - l^t_\psi| + 2|\bar{x}_\Psi - \lambda^t_\psi| + 2|\bar{\bar{x}}_\Psi - \Lambda^t_\psi| + |\bar{\bar{x}}_\Psi - L^t_\psi| \right),
\]

then \( \mu^t = S(x, M^t) \) if \( x \in E^t \). Refer to [5] 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 bottom-up learning method is suggested to avoid time consuming training common to conventional learning methods based on multiple passes over the data.

Expansion regions \( E^t \), see (12), help to derive criteria for deciding if data samples \( x \) belong to a same granule. Different values of \( \rho \) produce different representations of the same data set in different granularities. For normalized data, \( \rho \) assumes values in \([0, 1] \)." If \( \rho \) is equal to 0, then granules are not expanded. Learning creates a new rule for each sample, which causes overfitting and leads to excessive complexity. If \( \rho \) is equal to 1, then a single granule covers the entire data domain.

Rule creation is needed whenever one or more entries of \( x \) do not belong to the expansion regions \( E^t \) of \( M^t \), \( i = 1, ..., c \). A new granule \( M^{t+1} \) is assembled from fuzzy sets \( M^{t+1} \), \( \psi = 1, ..., \Psi \), whose parameters match \( x \), that is,

\[
M^{t+1} = \left[ \lambda^{t+1}, \lambda^{t+1}, \Lambda^{t+1}, \Lambda^{t+1}, \bar{L}^{t+1}, \bar{L}^{t+1}, \bar{\bar{L}}^{t+1} \right] = \left[ \bar{x}_\psi, \bar{x}_\psi, \bar{x}_\psi, \bar{x}_\psi, \bar{x}_\psi, \bar{x}_\psi, \bar{x}_\psi \right].
\]

Adaptation of an existing granule \( M^t \) consists in expanding the support \([l^t_\psi, L^t_\psi]\) and updating the core \([\lambda^t_\psi, \Lambda^t_\psi]\) of its fuzzy sets. Among all granules \( M^t \) that can expand to accommodate a particular sample \( x \), that with highest similarity according to (13) should be chosen. Adaptation proceeds depending on where the datum \( x_\psi \) is placed. Conditions for support expansion are:

\[
\begin{align*}
&\text{If } x_\psi \in [L^t_\psi - \rho, l^t_\psi] \text{ then } l^t_\psi(\text{new}) = x_\psi \text{ and } \lambda^t_\psi(\text{new}) = \lambda^t_\psi, \\
&\text{If } \bar{x}_\psi \in [L^t_\psi, l^t_\psi + \rho] \text{ then } L^t_\psi(\text{new}) = \bar{x}_\psi.
\end{align*}
\]

Core parameters are updated recursively from:

\[
\begin{align*}
\lambda^t_\psi(\text{new}) &= \frac{(w^t - 1)\lambda^t_\psi + x_\psi}{w^t}, \\
\Lambda^t_\psi(\text{new}) &= \frac{(w^t - 1)\Lambda^t_\psi + \bar{x}_\psi}{w^t},
\end{align*}
\]

where \( w^t \) is the number of times that the granule \( M^t \) was chosen to be adapted. Figure 2 shows seven possible adaptation situations. In the figure, the datum \( x = [x_1, x_2, \bar{x}, \bar{\bar{x}}] \) places either outside, partially inside or inside the fuzzy set \( M^t \). The learning procedure creates a new granule \( M^{t+1} \) or adapts the parameters of \( M^t \) accordingly.

E. Specificity-weighted recursive least squares method

The recursive least squares (RLS) algorithm is used to adapt the consequent parameters of rules as follows. Consider the consequent part of rule \( R^t \):

\[
x(k + 1) = A^t x(k)
\]

where \( x = [x_1, x_2, \bar{x}_\psi, x_\psi]^T \). The elements of \( A^t \) are denoted \( a^t_\psi, \psi = 0, ..., \Psi \). Rule \( R^t \) is chosen to be adapted whenever the antecedent part \( M^t \) is more similar to \( x(k) \) than the antecedent part of the remaining rules. When instance \( x(k + 1) \) becomes known, equation (17) can be solved for \( A^t \).
Form, equation (18) rewrites \( Y \) as the regression vector; and coefficients. A fuzzy datum with greater uncertainty is conveyed by a datum [12]. Data uncertainty can be preserved by weighing the adjustment or center of gravity (9), which depends on their symmetricity.

Let \( \psi \) be the vector of unknown coefficients; \( x = [x_0, a_0^1, ... a_{\psi}^i, x_{\psi}]^T \) be the vector of unknown coefficients; \( \chi = [1 \ CoG(x_1) \ ... \ CoG(x_{\psi})]^T \) be the regression vector; and \( \Omega = [CoG(x_{\psi})(k+1)] \). Then, in matrix form, equation (18) rewrites

\[
\Omega = Xa_{\psi}.
\]  

Expanding the \( \psi \)-th row of (17) we get

\[
x_{\psi}(k+1) = a_{\psi}^i + a_{\psi}^{i+1}x_1(k) + ... + a_{\psi}^{i+\psi}x_{\psi}(k).
\]  

The standard RLS algorithm can be applied for each row of (17) if we replace the trapezoids \( x_{\psi} \) by their midpoint (8) or center of gravity (9), which depends on their symmetricity. Data uncertainty can be preserved by weighing the adjustment of \( a_{\psi}^i \). Specificity measures refer to the amount of information conveyed by a datum [12]. A fuzzy datum with greater uncertainty (lower specificity) may not be as important as one with smaller uncertainty.

Let \( a_{\psi} = [a_{\psi}^0, a_{\psi}^1, ... a_{\psi}^{i+\psi}]^T \) be the vector of unknown coefficients; \( \chi = [1 \ CoG(x_1) \ ... \ CoG(x_{\psi})]^T \) be the regression vector; and \( \Omega = [CoG(x_{\psi})(k+1)] \). Then, in matrix form, equation (18) rewrites

\[
\Omega = Xa_{\psi}.
\]

To estimate the coefficients \( a_{\psi} \) we let

\[
\Omega = Xa_{\psi} + \Xi,
\]

where \( \Xi := [\epsilon_{\psi}(k+1)] \) and

\[
\epsilon_{\psi}(k+1) = CoG(x_{\psi})(k+1) - CoG(\hat{x}_{\psi})(k+1)
\]

is the approximation error. While in batch estimation the rows in \( \Omega \), \( X \) and \( \Xi \) increase with the number of available samples, in recursive mode only two rows are kept and we reformulate equations (19)-(21) as follows:

\[
\Omega = \begin{bmatrix} CoG(x_{\psi})(k) \\ CoG(x_{\psi})(k+1) \end{bmatrix}, \quad \Xi = \begin{bmatrix} \epsilon_{\psi}(k) \\ \epsilon_{\psi}(k+1) \end{bmatrix}
\]

\[
X = \begin{bmatrix} 1 & CoG(x_1)(k-1) & \cdots & CoG(x_{\psi})(k-1) \\ 1 & CoG(x_1)(k) & \cdots & CoG(x_{\psi})(k) \end{bmatrix}.
\]

The rows of the matrices in (22) refer to values before and just after adaptation. The RLS algorithm chooses \( a_{\psi} \) to minimize the functional

\[
J(a_{\psi}) = \Xi^T \Xi.
\]

\( a_{\psi} \) is given by

\[
a_{\psi} = (X^T X)^{-1} X^T \Omega.
\]

Let \( Q = (X^T X)^{-1} \). From the matrix inversion lemma [13] we avoid inverting \( X^T X \) using:

\[
Q(\text{new}) = Q(\text{old}) \left[ I - \frac{X^T X Q(\text{old})}{1 + X^T X Q(\text{old})} \right],
\]

where \( I \) is the identity matrix. 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

\[
a_{\psi}(\text{new}) = a_{\psi}(\text{old}) + Q(\text{new})X^T(\Omega - Xa_{\psi}(\text{old}))
\]

or, similarly,

\[
a_{\psi}(\text{new}) = a_{\psi}(\text{old}) + Q(\text{new})X^T \Xi.
\]

Yager [11] defines the specificity of a trapezoid \( x_{\psi} \) as

\[
sp(x_{\psi}) = 1 - \text{wdt}(x_{\psi}(0.5)).
\]

This simply means one minus the width of the 0.5 level set of \( x_{\psi} \). In terms of the parameters of \( x_{\psi} \) we get

\[
sp(x_{\psi}) = 1 - \frac{(\overline{x}_{\psi} + \underline{x}_{\psi}) - (\overline{x}_{\psi} + \underline{x}_{\psi})}{2}.
\]
Let the specificity of \( x \) be a diagonal matrix:

\[
sp(x) = diag([sp(x_1) \ldots sp(x_\Psi)]).
\]  

(30)

Then, equation (27) can be rewritten as

\[
a_i^\Psi_{(new)} = a_i^\Psi_{(old)} + sp(x)Q(new)x^T\Xi
\]  

(31)

to consider data uncertainty.

Figure 3 shows an intuitive idea of how the specificity-weighted RLS method works. In the left graph, the coefficients \( a_i'(old) \) of the approximation function resulted 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 \( x(4) \) arose (with the same specificity as that of previous data), the algorithm pondered its contribution equivalently to the contribution of previous data to adapt \( a_i'(old) \) and yield \( a_i'(new) \). Conversely, in the right graph, the specificity of the new datum \( x(4) \) was lower than that of \( x(1), x(2) \) and \( x(3) \). The higher uncertainty on the value of \( x(4) \) caused a smaller adjustment of the approximation function toward \( x(4) \).

The specificity-weighted RLS algorithm described in this section is repeated for \( \psi = 1, \ldots, \Psi \) at each time step. Detailed derivations of the RLS algorithm can be found in [1]. For a convergence proof, see [6].

III. THE LORENZ ATTRACTOR

An application example is given to illustrate the usefulness of the evolving modeling approach. The Lorenz attractor is a nonlinear system derived as a simplified model of fluid convection induced by temperature change. The system 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 an unknown dynamical system from data.

The discrete-time Lorenz equations are:

\[
\begin{align*}
x_1(k+1) &= x_1(k) + \sigma(x_3(k) - x_1(k)) + \eta_1 \\
x_2(k+1) &= x_2(k) + (rx_1(k) - x_1(k)x_3(k) - x_2(k)) + \eta_2 \\
x_3(k+1) &= x_3(k) + (x_1(k)x_2(k) - bx_3(k)) + \eta_3
\end{align*}
\]  

(32)

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. \( \sigma \) and \( r \) are the Prandtl and Rayleigh numbers, \( b \) is the geometric factor [8]. The nonlinearities are \( x_1x_3 \) and \( x_1x_2 \). We consider \( \sigma = 10, r = 28, b = 8/3; dt = 0.005 \) is the time step. \( \eta_i \) is a random value in \([-0.5, 0.5]\). The initial state \( x(0) \) is \((13; 0; 0)\). As shown in Fig. 4, the trajectory of the system states in the phase space settles into an irregular, aperiodic oscillation, 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 manages to move in this set forever without intersecting.

![Fig. 3. Intuitive idea of the specificity-weighted RLS method](image)

![Fig. 4. Lorenz chaotic system: phase space trajectory](image)

The one-step prediction results using the maximum width for granules \( \rho = 40 \) and pointwise data is shown in Fig. 5. The root mean square error, calculated as

\[
RMSE = \sqrt{\frac{1}{K} \sum_{k=1}^{K} (x(k+1) - \hat{x}(k+1))^2},
\]

(33)

is 0.3855. Three rules were estimated. Their parameters are:

Rule 1:

\[
\begin{align*}
M_1^1 &= (2.6659, 11.6382, 11.6382, 20.6104) \\
M_2^1 &= (0, 14.9987, 14.9987, 29.9973) \\
M_3^1 &= (0, 19.4143, 19.4143, 38.8287) \\
A_1 &= \begin{bmatrix}
1 & 0 & 0 & 0 \\
-4.5679 & -8.5766 & 9.4237 & -0.3404 \\
209.8369 & 5.5858 & 1.0490 & -12.8230 \\
-143.5199 & 14.0343 & 10.3979 & -0.6229
\end{bmatrix}
\end{align*}
\]

Rule 2:

\[
\begin{align*}
M_1^2 &= (6.6224, 14.5326, 14.5326, 22.4427) \\
M_2^2 &= (-16.3677, 4.1085, 4.1085, 24.5847) \\
M_3^2 &= (38.0141, 46.3665, 46.3665, 54.7189) \\
A_2 &= \begin{bmatrix}
1 & 0 & 0 & 0 \\
18.8130 & -5.8793 & 8.9295 & -1.7659 \\
875.0623 & -8.7513 & -3.3158 & -22.1597 \\
\end{bmatrix}
\end{align*}
\]
of the Lorenz equations are

\[ \sigma_k(x_j) \]

perceptions of the values of a variable. Imprecision of the shifts and interval fuzzy data. We consider the data as is unlike to track the trajectory of the states. Initial states, a non-evolving (offline-trained) modeling method theories for small differences in the measurements, parameters or experiment is that due to exponential divergence of the trajectory of the system states. An important point to emphasize in this have relatively small amplitudes compared to the amplitudes the data and system equations can be verified. The error signals predicting nonlinear systems without prior knowledge about

\( \sigma_k(x_j) \) shifted to \( k \) to \( 900 \) \( 0 \) \( 0 \) \( 0 \) \( 0 \) \( 0 \) \( 0 \)

Fig. 5. One-step estimation of the Lorenz chaotic map

From Fig. 5, 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 to 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 non-evolving (offline-trained) modeling method is unlike to track the trajectory of the states.

A second experiment was conducted to evaluate drifts, shifts and interval fuzzy data. We consider the data as perceptions of the values of a variable. Imprecision of the values of \( x_j \) was represented as fuzzy objects of the form \( (x_j - 0.5, x_j, x_j + 0.5) \). From \( k = 1, \ldots, 300 \), the parameters of the Lorenz equations are \( \sigma = 10, r = 28, b = 8/3 \). An abrupt change happens at \( k = 300 \) where the parameters are shifted to \( \sigma = r = b = 10 \). Then, at each step from \( k = 600 \) to \( k = 900 \), an offset of 0.03 is added to \( \sigma, r \) and \( b \) to simulate gradual change of parameters. Figure 6 shows the results for variable \( x_3 \). The results for other variables are similar.

Note from Fig. 6 that, after the concept shift, the error rate oscillates. Nevertheless, after some time steps, the estimation performance is recovered. 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, 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.

IV. CONCLUSION

This paper has introduced an evolving fuzzy granular method for online modeling of nonlinear time-varying systems. The method is able to process and learn incrementally from numeric and fuzzy interval data. Computational experiments have considered the Lorenz system with changing parameters to show the usefulness of the evolving granular method. Further studies are needed to account for different kinds of nonstationarities and uncertainties in data streams. Stabilization of time-varying nonlinear systems based on linear matrix inequalities shall also be investigated.

ACKNOWLEDGMENT

The first author acknowledges CNPq, the Brazilian National Research Council, for grant 500616/2013-2.

REFERENCES