Adaptive Inferential Sensors Based on Evolving Fuzzy Models

Plamen Angelov, *Senior Member, IEEE*, and Arthur Kordon, *Member, IEEE*

***Abstract*—A new technique to the design and use of inferential sensors in the process industry is proposed in this paper, which is based on the recently introduced concept of evolving fuzzy models (EFMs). They address the challenge that the modern process industry faces today, namely, to develop such adaptive and self-calibrating online inferential sensors that reduce the maintenance costs while keeping the high precision and interpretability/transparency. The proposed new methodology makes possible inferential sensors to recalibrate automatically, which reduces significantly the life-cycle efforts for their maintenance. This is achieved by the adaptive and flexible open-structure EFM used. The novelty of this paper lies in the following: 1) the overall concept of inferential sensors with evolving and self-developing structure from the data streams); 2) the new methodology for online automatic selection of input variables that are most relevant for the prediction; 3) the technique to detect automatically a *shift* in the data pattern using the *age* of the clusters (and fuzzy rules); 4) the online standardization technique used by the learning procedure of the evolving model; and 5) the application of this innovative approach to several real-life industrial processes from the chemical industry (evolving inferential sensors, namely, *eSensors*, were used for predicting the chemical properties of different products in The Dow Chemical Company, Freeport, TX). It should be noted, however, that the methodology and conclusions of this paper are valid for the broader area of chemical and process industries in general. The results demonstrate that well-interpretable and with-simple-structure inferential sensors can automatically be designed from the data stream in real time, which predict various process variables of interest. The proposed approach can be used as a basis for the development of a new generation of adaptive and evolving inferential sensors that can address the challenges of the modern advanced process industry.**

***Index Terms*—Concept shift in data streams, evolving fuzzy systems, fuzzy-rule aging, inferential sensors, learning and adaptation, Takagi–Sugeno (TS) fuzzy models.**

I. Introduction

INFERENTIAL sensors [1], [21], [23], [27], [28] are able to provide accurate real-time estimates of difficult-to-measure parameters or expensive measurements (like emissions, biomass, melt index, etc.) from the available cheap sensors (like temperatures, pressures, and flows). Different empirical methods have been used to develop inferential sensors, such as statistical models [2], neural networks (NNs) [3], support-vector machines [4], [22], and genetic programming [5], [13]. Model-based techniques for process-quality monitoring [1] often provide a valuable advantage over conventional approaches that rely on manual intervention and laboratory tests. Such models, however, are costly to build and maintain since the environment in which an industrial process takes place is dynamically changing, the equipment is getting older and contaminated or being replaced, raw materials usually alter in quality, and the complexity of processes leads to a number of aspects of the process being ignored by the models. A crucial weakness of model-based approaches is that they do not take into account the *shift* and *drift* in the data pattern that is related to the fact that these models are developed offline under certain conditions. Even minor process changes outside these conditions may lead to unacceptable performance deterioration that requires manual maintenance and recalibration.

The challenge is to develop inferential sensors with flexible yet interpretable structure [6] and adaptive parameters. The *gradual* evolution of the model structure (fuzzy rules) will mean that a retraining of the sensor when required will only modify (add, remove, or replace) one or few fuzzy rules [7]. Contrast this to a possible option of iteratively retraining an NN, which, in effect, will lead to a completely new NN and a loss of previous information [29]. Ideally, we would require inferential sensors that can automatically recalibrate and detect *shifts* and *drifts* in the data stream [4], [8]. One such methodological framework is presented by the evolving Takagi–Sugeno (ETS) fuzzy models [9], [10]. In this paper, we use this framework and build upon it a methodological concept for evolving inferential sensors, namely, *eSensors*, which is new and original. The main contributions of this paper include the following: 1) the overall concept of *eSensors*; 2) the new methodology for online automatic selection of input variables that are most relevant for the prediction; 3) the technique to detect automatically a *shift* in the data pattern using the *age* of the clusters (and fuzzy rules); 4) the online standardization technique used by the learning procedure of the evolving model; and 5) the application of this innovative approach to four real-life industrial processes from the chemical industries.

II. Adaptive Inferential Sensors Based on EFM

A. Principles of EFM

Evolving fuzzy models (EFMs) were first introduced as a technique for online adaptation of fuzzy-rule-based systems'

Manuscript received October 17, 2008; revised March 27, 2009 and June 26, 2009. This paper was recommended by Associate Editor T. H. Lee.

P. Angelov is with the Intelligent Systems Research Laboratory, Infolab21, Lancaster University, LA1 4WA Lancaster, U.K. (e-mail: p.angelov@lancaster.ac.uk).

A. Kordon is with The Dow Chemical Company, Freeport, TX 77541 USA (e-mail: AKKordon@Dow.com).

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/TSMCB.2009.2028315

structure (rule-based fuzzy sets), as well as their parameters [7], [14]. In that respect, they make a step further by comparing the aforementioned technique to the well-established adaptive-system theory [15], which is applicable to linear systems only and to a small circle of nonlinear systems. EFM systems are nonlinear, linguistically interpretable, yet adaptable online in a (local) least squares (LS) sense. The approach was further refined for the specific case of the so-called TS fuzzy models [16] by introducing a fully recursive algorithm called ETS [9], [10]. ETS fuzzy models are particularly suited as a framework for addressing the challenges that the process industry faces nowadays. They can provide the algorithmic backbone of systems that can be implemented as embedded autonomous intelligent sensors with self-calibration and self-maintenance capabilities.

The basic idea of ETS is to allow the TS fuzzy system structure to grow, shrink, adapt, and self-develop in an automatic fashion learned online from the data streams in a locally optimal way. TS fuzzy systems [16] are very attractive due to their dual nature—they combine the fuzzy linguistic antecedent part with a linear functional consequent part, thus being locally linear but nonlinear overall and being proven universal approximators [17]. The antecedent part is a linguistic representation of a partition of the measurable-variable space into fuzzily overlapping regions (see Fig. 14). The linguistic antecedent parts of TS fuzzy systems make them attractive for human operators (compared to NN, SVM, or polynomial models, for example). The architecture of an ETS fuzzy system is based on fuzzily weighted local linear models of the following form [9], [10]:

$$LM^i : y^i = \overline{x}^T \Theta \tag{1}$$

where LM^i denotes the ith local model, $i = 1, 2, \ldots, N$; $\overline{x} = [1, x_1, x_2, \ldots, x_n]^{\mathrm{T}}$ represents the $(n+1) \times 1$ extended vector of measurable variables; $y^i = [y_1^i, y_2^i, \ldots, y_m^i]^{\mathrm{T}}$ is the $m \times 1$ vector of estimated variables; and $\Theta^i = [\theta_0^i \quad \theta_1^i \quad \cdots \quad \theta_n^i]^{\mathrm{T}}$ denotes the matrix of consequent parameters.

All of the N local linear models describe the process in a local area defined by fuzzy rules and are blended in a fuzzy way to produce the overall output that is nonlinear in terms of measurable variables x's but is linear in terms of parameters Θ's

$$y = \psi^T \Theta \tag{2}$$

where $\psi = [\lambda^1 \overline{x}^T, \lambda^2 \overline{x}^T, \ldots, \lambda^N \overline{x}^T]^{\mathrm{T}}$ is a vector of measurable variables that are weighted by the normalized activation levels of the rules, λ^i, $i = 1, 2, \ldots, N$, with λ^i being the normalized firing level of the ith fuzzy rule that is a function of x, i.e., $\lambda^i(x)$.

The overall TS fuzzy model can then be described by a set of fuzzy rules of the following form:

$$\begin{aligned} R^i : \text{IF } (x_1 \text{ is around } x_1^{i*}) \text{ AND}, \ldots \\ \text{AND } \left(x_n \text{ is around } x_n^{i*}\right), \text{ THEN } (y^i = LM^i) \end{aligned} \tag{3}$$

where R^i denotes the ith fuzzy rule, with $i = [1, N]$; N is the number of fuzzy rules; (x_j is around x_j^{i*}) denotes the jth fuzzy set of the ith fuzzy rule, with $j = 1, 2, \ldots, n$; and x^{i*} is the focal point of the ith-rule antecedent part.

The degree of membership of a certain data point (x) to any of the fuzzy rules can be described by a Gaussian centered at its focal point

$$\mu^i = e^{-\frac{\sum_{j=1}^{n}\left(x_j - x_j^{i*}\right)^2}{2\left(\sigma_j^i\right)^2}} \tag{4}$$

having a spread that is learned based on the data variance [10]

$$\begin{aligned} \left(v_{jk}^i\right)^2 &= \rho\left(v_{j(k-1)}^i\right)^2 + (1-\rho)\frac{1}{n_k^i}\sum_{l=1}^{n_k^i} \|z^{i*} - z_l\|_j^2, \\ v_{j1}^i &= 1, \quad \sigma_{jk}^i \leftarrow v_{jk}^i \end{aligned} \tag{5}$$

where v_{jk}^i denotes the variance of the data in the ith cluster in the jth dimension (jth variable) calculated at the kth time instant, σ_{jk}^i represents the spread of the Gaussian of the jth fuzzy set of the ith fuzzy rule calculated at the kth time instant, $z = [x, y]^{\mathrm{T}}$ depicts the overall data vector, and n_k^i denotes the *support* of the ith cluster/rule—the number of samples that are associated with it based on the distance to the focal point.

The firing strength of a fuzzy rule is determined by a *t-norm*, which can be represented as inner product [18]

$$\tau^i = \prod_{j=1}^{n} \mu_j^i(x_j) \tag{6}$$

and is normalized so that it sums to one

$$\lambda^i = \frac{\tau^i}{\sum_{j=1}^{N} \tau_j}. \tag{7}$$

B. Monitoring the Quality of the Rule Base

One can monitor and analyze online the quality of the clusters that are formed and the fuzzy rules, respectively—for example, the number of points that support them or their *age* [19]. The support of the rules is determined by a simple counting of the samples that are associated with the *nearest* focal point

$$n_{k+1}^i = n_k^i + 1, \quad i = \arg\min_{i=1}^{N} \|x_k - x^{i*}\|, \quad k = 2, 3, \ldots. \tag{8}$$

The support is initiated by one at the moment a rule is created

$$n_k^{N+1} \leftarrow 1, \quad k = 2, 3, \ldots. \tag{9}$$

In this paper, we introduce a recursive formula to calculate the *age* of the ith cluster/rule calculated at the kth moment in time (data sample)

$$A_k^i = k - \frac{1}{n_k^i}\left(k - A_{k-1}^i + k_{n_k^i}\right) \tag{10}$$

where k_l is the time index when the data sample was read.

This follows from

$$A_k^i = k - \frac{1}{n_k^i}\sum_{l=1}^{n_k^i} k_l \quad A_{k-1}^i = k - \frac{1}{n_{k-1}^i}\sum_{l=1}^{n_{k-1}^i} k_l.$$

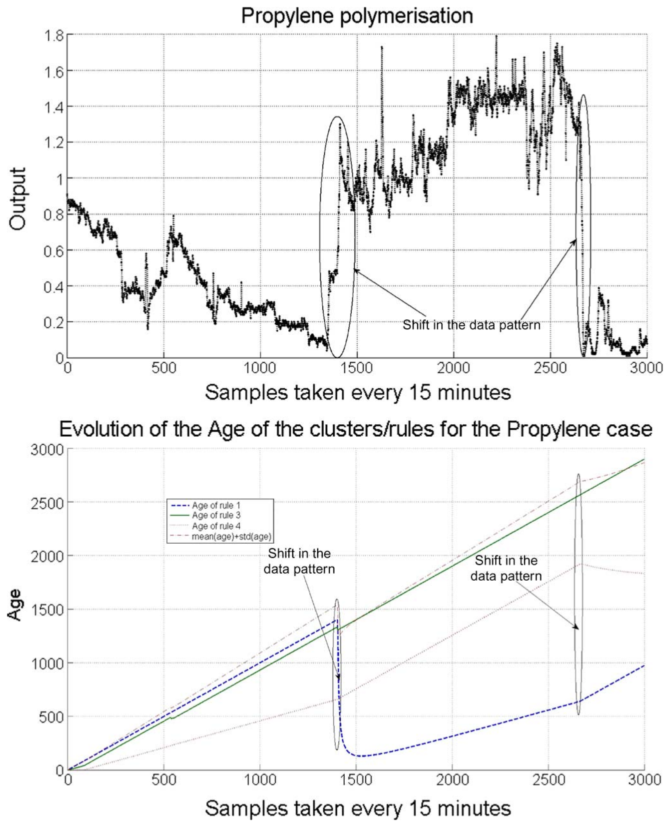

Fig. 1. (a) Top plot—output variable in case study 4—polymerization; (b) Bottom plot—*age* of the fuzzy rules describing the propylene-polymerization process. The two instants when a *shift* in the data pattern occurs are marked. This corresponds to a change in the *aging* rate seen from the bottom plot.

From there, we get

$$\sum_{l=1}^{n_{k-1}^i} k_l = \left(k - A_{k-1}^i\right) \quad A_k^i = k - \frac{1}{n_k^i}\left(\sum_{l=1}^{n_{k-1}^i} k_l + k_{n_k^i}\right).$$

Combining these two expressions, we arrive at (10).

Each time a new rule is created, its *age* is initiated by the index of the data sample that is used as a focal point of that rule. Each time a new data sample is associated to an existing rule (the distance from a sample to that focal point is smaller than that to any other focal points), the *age* of that rule gets smaller. If no sample is assigned to a rule, it gets older by one. Note that the *age* of a fuzzy rule can take values from the $[0; k]$ range. This is shown in Fig. 1 in the case of propylene estimation. From the top plot, one can see that there are three different stages of that process. The *aging* of three of the six fuzzy rules (rules ## 1, 3, and 4) are depicted in the bottom plot. One can see that precisely at the moment of a *shift* in the data pattern (a new phase), the *aging* of the rules is affected. By monitoring the derivative of A (i.e., *aging rate*), one can automatically detect such changes and respond by adapting the learning mechanism or rate.

Note that the *age rate* of rule #1 becomes ***negative*** before it increases again. This illustrates the so-called concept *shift* and is an indication of a transition from one operating state (which affects the data density in one local region, i.e., around the focal point of this rule) to another one (which affects the data density in another local region).

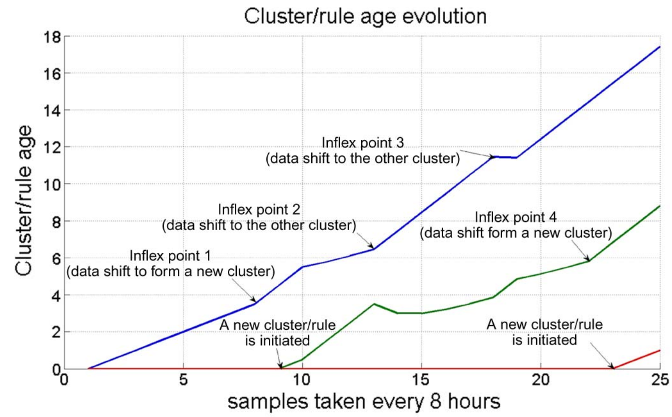

Fig. 2. Evolution of the *age* and *shift* in the data pattern, resulting in forming new clusters/rules for case study 2. The inflex points correspond to a shift of the data from one cluster to another existing cluster or to a newly formed cluster (as marked in the figure for each inflex point).

The *age* of the fuzzy rules (and the derivative of their *age* in terms of the sampling period (k), which represents the *aging rate*) can be very useful for online analysis of the concept *shift* in the data stream [12]. An *eSensor* can detect a concept *shift* [20] online by the rate of *aging* and the instances when it changes [the inflex points on the *age* evolution diagram that corresponds to the change of the sign of the *aging* rate indicate a *shift* (see Fig. 2)]. The aging corresponds to the first derivative of the *age* and is graphically represented by the slope of the age evolution lines in terms of the horizontal axis [see Fig. 1(b)].

In this paper, we use the following principle for the update of the rule base by removing the *older* rules (rules whose *age* exceeds the mean *age* for that rule by more than the standard-deviation [2] value calculated recursively up to that moment/sample):

$$\text{IF}\Big(A^i > \overline{A^i} + \text{std}(A^i)\Big), \text{THEN}\left(\text{remove } R^i; N \leftarrow N-1\right) \quad (11)$$

where $\overline{A^i}$ denotes the mean *age* (it is also denoted in Fig. 1(b) by a dash-dotted line) and $\text{std}(A^i)$ represents the standard deviation of the *age* of the ith rule.

C. Evolving the Structure of the Sensor From the Data Stream

The online design and learning of the *eSensor* are outlined here. Learning is based on decomposition of the identification problem into the following [7], [9], [10]: 1) fuzzy-rule-based structure design and 2) parameter identification. Both of these subproblems can be performed in online mode during one time step (per sample). The first subproblem, i.e., structure identification, can be approached using evolving clustering in the data space [9], [10], [12]. This partitioning leads to forming information granules, described linguistically by fuzzy sets. Thus, it serves the transformation of the data into primitive forms of knowledge. The basic notion of the partitioning algorithm is that of the data *density* [26], which is defined as a Cauchy function over the sum of distances d's between a certain data sample z_i and ***all*** other data samples in the feature space [10]

$$D_k(z_k) = \frac{1}{1 + \overline{v}_k^2} \quad (12)$$

where $\overline{v}_k^2 = (1/k-1)\sum_{i=1}^{k-1} d^2(z_k, z_i)$ is the variance of the data [2].

Data-space partitioning is based on the following principle: *The point with the highest density in the data space is chosen to be the focal point, and the antecedent of the first fuzzy rule is formed around it*. In this way, fuzzy rules with high descriptive power and generalization capabilities are generated. The density can be ***recursively*** calculated using the current data point (z_k^j) and $(n+1)$ memorized quantities only (β_k and χ_k^j, $j=[1,n]$) [10]

$$D_k(z_k) = (k-1)\left(\alpha_k(k-1) + \beta_k - 2\gamma_k + (k-1)\right)^{-1}, \qquad k = 2,3,\ldots \tag{13}$$

where $\alpha_k = \sum_{j=1}^{n+1}(z_k^j)^2$; $\beta_k = \beta_{k-1} + \alpha_{k-1}$, with $\beta_1 = 0$; and $\gamma_k = \sum_{j=1}^{n+1} z_k^j \chi_k^j$, with $\chi_k^j = \chi_{k-1}^j + z_{k-1}^j$ and $\chi_1^j = 0$.

Each time a new data sample is read, it affects the data density of the existing focal points and can be updated by [10]

$$D_k(z^{i*}) = \frac{(k-1)D_{k-1}(z^{i*})}{k-2+D_{k-1}(z^{i*}) + D_{k-1}(z^{i*})d(z^{i*}, z_k)}, \qquad k = 2,3,\ldots \tag{14}$$

where $d(z^{i*}, z_k)$ denotes the distance between the ith focal point and the current point.

Once the densities of the new coming data sample and of each of the previously existing focal points are recursively updated, they are compared. If the new coming data sample has a higher density than ***any*** of the previously existing focal points, then this means that it is a good candidate to become a focal point of a new rule (a new local linear model) because it has high descriptive power and generalization potential

$$D_k(z_k) > D_k(z^{i*}) \quad \forall i^* \in N. \tag{15a}$$

If the new coming data sample has a lower density than ***any*** of the previously existing focal points, then this means that it is also a good candidate to become a focal point of a new rule (a new local linear model) because it improves the coverage of the whole data space [12]

$$D_k(z_k) < D_k(z^{i*}) \quad \forall i^* \in N. \tag{15b}$$

Forming a new fuzzy rule around a newly added prototype leads to a *gradual* increase of the size of the rule base, which is why this approach is called "evolving"

$$z^{(N+1)*} \leftarrow z_k. \tag{16}$$

The density of the newly generated rule is set to one [10] temporarily (it will be updated to take into account later the influence of each new coming data sample on the generalization potential of this particular focal point)

$$D_k\left(z^{(N+1)*}\right) \leftarrow 1. \tag{17}$$

To increase the interpretability and update of the rule base, one needs also to remove the previously existing rules that become ambiguous after insertion of the new rule. Therefore, each time a new fuzzy rule is added, it is also checked whether any of the already existing prototypes in the rule base are described by this rule to a degree that is higher than 50%

$$\exists i, \quad i=[1,N]; \quad \mu_i^j(z^{N+1}) > 0.5 \quad \forall j, \quad j=[1,n]. \tag{18}$$

If any of the previously existing focal points satisfy this condition, the rules that correspond to them are being removed (replaced by the newly formed rule) [9], [19]. The spreads of the membership functions are also recursively updated by (5).

D. Self-Learning the eSensor

Once the antecedent part of the TS fuzzy model is formed, the consequent-parameter estimation (the second subproblem of the learning) is addressed as a fuzzily weighted recursive LS (RLS) estimation problem per rule [15]

$$\Theta_k^i = \Theta_{k-1}^i + C_k^i \lambda^i \overline{x}_k \left(y_k - \overline{x}_k^T \Theta_{k-1}^i\right), \quad \Theta_1^i = 0 \tag{19}$$

$$C_k^i = C_{k-1}^i - \frac{\lambda^i C_{k-1}^i \overline{x}_k \overline{x}_k^T C_{k-1}^i}{1 + \lambda^i \overline{x}_k^T C_{k-1}^i \overline{x}_k}, \quad C_1^i = \Omega I, \quad k = 2,3,\ldots \tag{20}$$

where $C \in R^{N(n+L)xN(n+L)}$ denotes the covariance matrix, Ω is a large positive number, and I is the identity matrix.

As a result, the *eSensor* blends in a fuzzy way local linear predictors. Moreover, it is optimally (in an LS sense) [15] tuned in terms of consequent parameters Θ's. In terms of its antecedents and rule-based structure, it is based on the robust online partitioning approach. The procedure of the *eSensor* self-development and self-calibration is represented as a pseudocode in the Appendix.

E. Online Normalization and Standardization of the Data in the eSensor

One specific issue related to this online algorithm is the normalization or standardization of the data. Both normalization and standardization are well-established techniques for the offline case when all the data are available [2]. An approach to update the normalization ranges of the data in a recursive manner is presented in [25], but in this paper, we use the recursive version of the standardization technique that can easily be inferred from the offline version [2] because it depends on the mean and variance of the data only. Let us remember that (offline) standardization is given by [2]

$$Z_{jk} = \frac{z_{jk} - \overline{z}_{jk}}{\zeta_{jk}}, \quad j = [1,n], \quad k = 2,3,\ldots \tag{21}$$

where Z_{jk} denotes the standardized value of z_{jk}; $\overline{z}_{jk} = (1/k)\sum_{l=1}^{k} z_{jl}$, $j=[1,n]$, $k=2,3,\ldots$, represents the mean value of z_{jk}; and v_{jk} is the standard deviation of the jth input calculated based on k data samples.

Both the mean and standard deviation can be updated recursively

$$\overline{z}_{jk} = \frac{k-1}{k}\overline{z}_{j(k-1)} + \frac{1}{k}z_{j(k-1)},$$
$$\overline{z}_{j1} = 0, \quad j = [1, n+m], \quad k = 2, 3, \ldots \tag{22a}$$
$$v_{jk}^2 = \frac{k-1}{k}v_{j(k-1)}^2 + \frac{1}{k-1}\left(z_{jk} - \overline{z}_{j(k-1)}\right),$$
$$v_{j1} = 0, \quad j = [1, n+m], \quad k = 2, 3, \ldots. \tag{22b}$$

In order to return to the original scale, one should apply destandardization by

$$z_{jk} = Z_{jk}\nu_{jk} + \overline{z}_{jk}, \quad j = [1, n+m], \quad k = 2, 3, \ldots. \tag{23}$$

III. Online Input-Variable Selection in the eSensor

Inferential sensors, as well as other online models, traditionally assume the number of input variables to be known beforehand or to be preselected. In what follows, we propose an original[1] method to online "on-fly" ranking and selection of input variables, which was successfully approbated on the industrial case studies reported in this paper, as well as on other real applications [30]. The importance of this technique should not be underestimated because, very often in practice, there are large sets of candidate variables that may influence the monitored or measured output, but often, it is not clear how much. The idea is based on online ranking of the accumulated values formed by the consequent parameters Θ_{jk}^i, $j=[1,N]$,$i=[1,R]$.

The accumulated values π's indicate that the weight of a particular consequent parameter is determined by simply adding the absolute values (because the consequent parameters are unrestricted in sign and value, and their contribution is judged by the modulus)

$$\pi_{jk}^i = \sum_{l=1}^{k}\left|\Theta_{jl}^i\right|, \quad j = [1, n], \quad i = [1, R]. \tag{24}$$

One can also form a weight of a particular feature by the ratio of π values

$$\omega_{jk}^i = \frac{\pi_{jk}^i}{\sum_{r=1}^{n}\pi_{jk}^i}, \quad i = [1, R], \quad j = [1, n]. \tag{25}$$

It is important to note that (24) and (25) represent sums only and are thus easily performed online. The values of the weights ω's indicate the contribution of a particular input to the overall output and are thus a measure of the sensitivity of the outputs. Therefore, an intuitive technique to simplify the inferential sensor structure in terms of inputs can be proposed, which gradually removes the input variables for which the weight ω is negligibly small across the rules (i.e., the inputs that contribute little to the overall output)

$$\text{IF}\left(\exists j^*\left|\omega_{j^*k}^i < \varepsilon\max_{j=1}^{n}\pi_{jk}^i\right.\right), \text{ THEN (remove } j^*) \tag{26}$$

[1]This technique is part of a pending patent: P. Angelov, Machine Learning (Collaborative Systems), WO2008053161, priority date: November 1, 2006; intern. filing date: October 23, 2007; http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=WO2008053161&F=0&QPN=WO2008053161

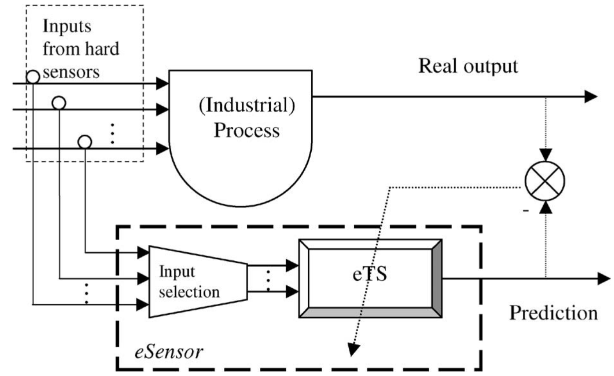

Fig. 3. Overall schematic representation of the *eSensor*.

where ε denotes a coefficient (the suggested values are [0.03; 0.1], which means that this input variable contributes 3%–10% to the overall output on average.

The rationale for the simplicity of this technique stems from the fact that the consequents represent locally linear combinations and can thus be analyzed. It should be noted that, when an input is removed (which does not usually occur very often), however, the dimension is reduced by one, which is reflected in the covariance matrices (a line and a column are removed), and the dimensions of the focal points are also updated, as well as the recursive variables in (13), i.e., α, β, γ, and χ.

The main advantages of the proposed *eSensor* approach that makes it suitable for implementation in the process industry are as follows.

1) It self-develops, *evolves*, and thus reduces the development and maintenance costs significantly.
2) It can provide high prediction rates.
3) It is one-pass and recursive and has low computational requirements; thus, it is suitable for hardware "on-chip" implementations [24].
4) It is useful for online analysis and monitoring of the concept *shift* using fuzzy-rule *aging* [see Figs. 1(b) and 2] and thus makes useful conclusions for possible faults and the quality of the process.
5) It can automatically select online a small subset of relevant inputs, thus fully automating the development process.
6) It can have a *multiple-input–multiple-output* structure and thus build a separate regression model for each output variable.

The procedure for adaptive and *evolving* inferential self-calibrating sensors, which we call *eSensor*, is presented by the pseudocode provided in the Appendix (see also Fig. 3).

IV. Case Study: Inferential Sensors for Chemical-Property Estimation

The capabilities of the proposed evolving inferential sensor are explored on four different industrial data sets for chemical-property estimation. All four cases include operating-regime changes with different impacts on specific chemical properties due to different levels of process change, various measurement methods with different accuracies, and a different number of potential process variables, related to the inferred chemical properties. However, all the changes create a challenge to

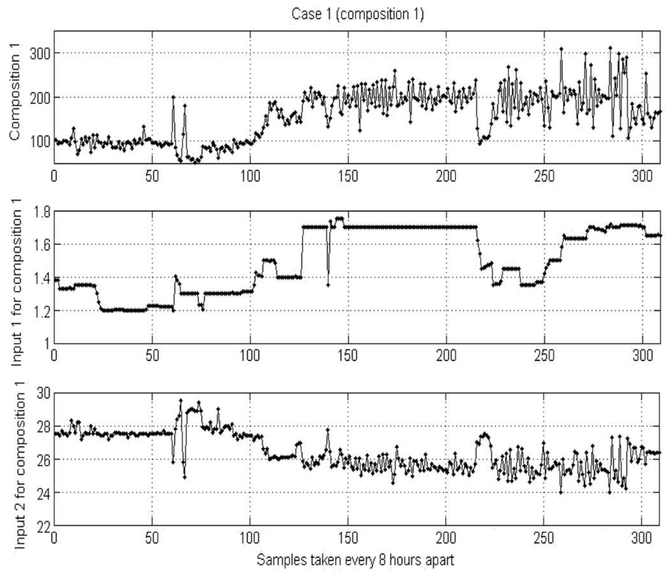

Fig. 4. Case study 1: Composition 1. Top plot—output variable (composition 1). Middle plot—input variable (x_1). Bottom plot—input variable (x_2).

existing inferential sensors with a fixed structure. As a basis for comparison, inferential sensors based on the most widely used methods in commercial soft-sensor products, such as the feedforward NN of multilayer perceptron (MLP) type [3] and PLS [1], were used, as well as a recently introduced algorithm for adaptive online NN, namely, DENFIS [31].

In the chemical industry, inferential sensors are mostly used to estimate chemical properties, measured by two techniques: 1) offline laboratory analysis of grab samples of the properties and 2) pseudo real-time analysis with low frequencies by gas chromatographs. The sampling period for the properties, measured by laboratory analysis, is several hours, and accuracy depends on different measurement methods and varies substantially. The sampling period of gas-chromatograph-based properties is much shorter (usually 15–30 min), and accuracy is, on average, an order of magnitude higher than that from offline laboratory measurements. Three of the selected cases are based on offline laboratory measurements, and one is based on gas chromatographs. In the cases with laboratory measurements, two different levels of accuracies have been selected. The level of operating-condition change (which could be quantified by the percentage increase from the average level for 50 samples before the process change to the average level for 50 samples after the change), as well as the number of process inputs, is also different.

The first case, called Composition 1, is based on product-composition estimation in a distillation tower. The measurements are based on laboratory analysis, taken every 8 h, and the method accuracy is low (2.2% measurement error), which, by itself, introduced a measurement noise. Process data are the hourly averaged values around the time when the sample for the laboratory measurement has been taken. The output composition and the two-input data (Fig. 4) include 309 records (samples). As it is seen in the middle plot in Fig. 4, a significant change in operating conditions has been introduced after sample 127 by input 1. It is interesting to note that the two input variables that were selected online using the *eSensor* are

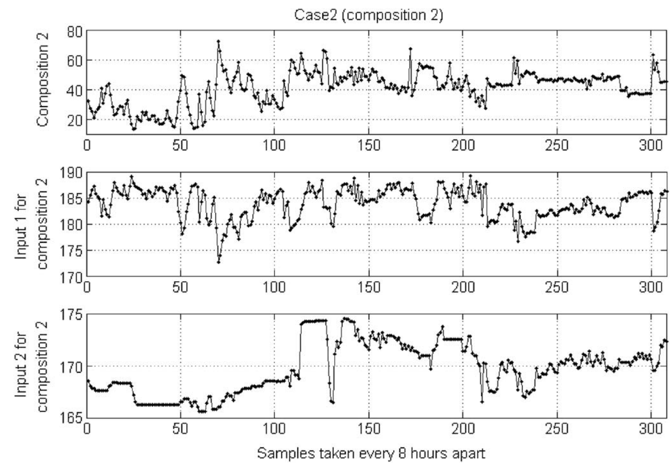

Fig. 5. Input and output variables for case study 2. Top plot—output variable (composition 2). Middle plot—input variable (x_1). Bottom plot—input variable (x_2).

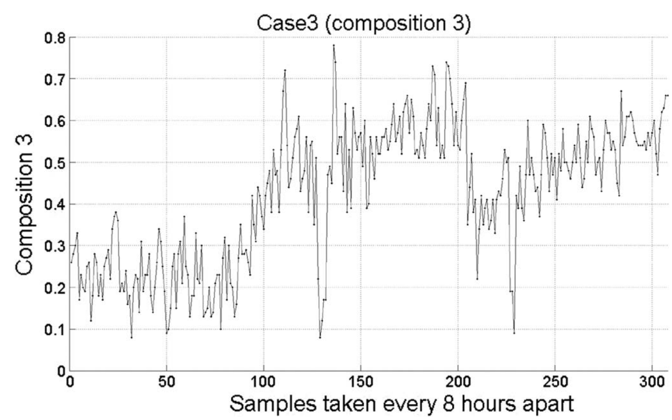

Fig. 6. Output variable for case study 3 (composition 3). There are seven selected inputs, and they are not shown for clarity purposes.

the most statistically significant process variables related to this composition.

The second case, called Composition 2, is based on product-composition estimation *in the bottom* of a distillation tower, which is different from the tower in Composition 1. The composition measurements are based on laboratory analysis, taken every 8 h with a more accurate method of 1.3% measurement error, and are less noisy. Process data are the hourly averaged values for the time when the sample for the laboratory measurement has been taken. The output composition and the two-input data (Fig. 5) include 308 records (samples), where a significant change in operating conditions has been introduced after sample 113 by input 2. Forty-seven different input variables were measured using "*hard*" (conventional) sensors.

The third case, called Composition 3, is based on product-composition estimation *in the top* of the same distillation tower as that in Composition 2. The output composition is shown in Fig. 6, and it also includes 308 data samples with a significant change in operating conditions (catalyzing agent replacement) introduced after sample 113. The key differences of Case 3 relative to the other laboratory-measurement-based cases are as follows: 1) higher level of operating-condition changes (275% increase versus 220% increase for Case 1 and 232% increase

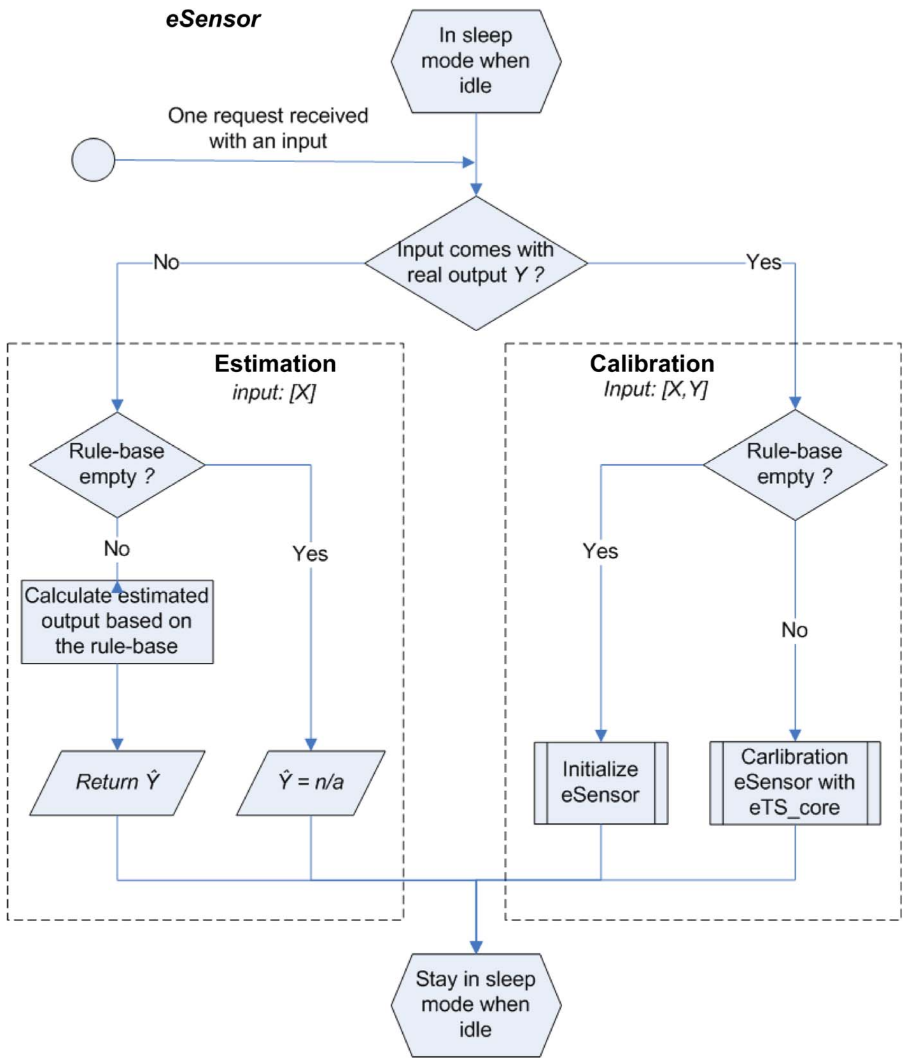

Fig. 7. Flowchart of the eSensor from the real-time software-realization point of view. Sleep mode means a default state expecting an external request. Note that all the stages of eSensor self-calibration are combined in one block on the right bottom part of the flowchart. This includes learning, the online input selection, as well as cluster/rule removal based on their *age*. The details of this procedure are provided in the Appendix.

for Case 2) and 2) larger number of process inputs (seven inputs versus two inputs for both Cases 1 and 2).

The fourth case is based on *propylene* estimation in the *top of a distillation tower*, which is different from the distillation towers in the previous cases. In this case 2, process variables that are related to propylene are used as inputs in the model development. The propylene measurements are based on gas-chromatograph analysis, taken every 15 min. Process data are the snapshot minute values for the time when the measurement has been taken. The data [Fig. 1(a)] include 3000 records (samples) with very broad range of operating conditions.

These four test cases (provided and used by The Dow Chemical Company, Freeport, TX) cover most of the real issues in applying inferential sensors in the advanced process industry, such as noisy data, changing operating conditions, a large number of correlated inputs, etc.

V. Experimental Results and Analysis

The main aim of the experimental study was to generate interpretable simple-to-understand models that are flexible and adaptive (evolving with time and following the dynamics of the data pattern) and are robust to noise and imprecise measurement data using the proposed technique *eSensor* and to compare these results with the available alternatives based on MLP-type NN, PLS, and a recently introduced evolving NN, i.e., DENFIS [31]. Precision was measured using root mean square errors (RMSE), as well as correlation [2]. The data in all experiments were standardized. The *eSensor* starts with an empty fuzzy-rule base (*no iniSensor*) and generates its rule-base "on fly" based on the data that are provided sample by sample and disregarded from the memory once processed. It also optimizes the parameters during retraining periods (it self-calibrates). The output prediction is calculated for every data sample and can be used at any time instant. Samples for recalibration are provided when they are available (see Fig. 7). DENFIS was also applied in an online mode.

The conventional inferential sensors (PLS and NN) that are not adaptive were trained initially using the first quarter of the available data samples, and afterward, they were retrained using samples from the third quarter of the available data stream. The error was only calculated on the second and fourth quarters of

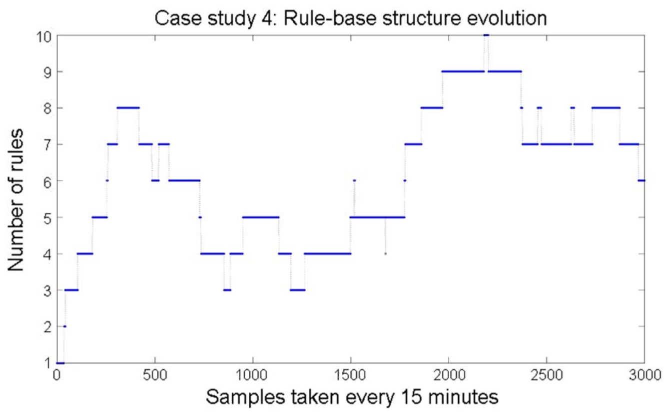

Fig. 8. Evolution of the rule base of the *eSensor* (number of rules vary starting from one—the first sample—finishing with six, and reaching at some stage 10).

TABLE I
VALIDATION RESULT USING ESENSOR AND REFERENCE APPROACHES

Case study		**1**	**2**	**3**	**4**
Total # of input variables		6	47	47	22
Total data samples		309	308	308	3000
Sampling period, *h*		1	8	8	1/4
eSensor* *(this paper)	# inputs	**2**	**2**	**7**	**2**
	RMSE	**18.533**	**2.3658**	**0.0716**	**0.0698**
	Correlation	**0.9477**	**0.9788**	**0.9026**	**0.9901**
	# rules	**4**	**3**	**4**	**6**
eTS [9,10] using all inputs	RMSE	18.918	3.786	0.075	0.116
	Correlation	0.884	0.834	0.782	0.976
	# rules	4	4	5	6
Feed-forward MLP	RMSE	23.12	2.87	0.098	0.628
	Correlation	0.890	0.91	0.802	0.85
PLS	RMSE	24.39	2.79	0.093	0.194
	Correlation	0.881	0.90	0.817	0.901
DENFIS [31] all inputs	RMSE	19.106	33.52	0.399	0.110
	# rules	19	32	32	235

the data stream in all cases (PLS, NN, and eSensor) to allow compatibility of the results. Note that the *eSensor* can also be retrained anytime when a training sample is available, and moreover, its structure (rule based) will be preserved and only gradually adapted/evolved.

The evolution of the fuzzy rule base is shown in Fig. 8, where the number of fuzzy rules generated is shown for the fourth case study (propylene). In retraining the NN and PLS, the parameters (weights) change completely and are not interpretable. Note that both PLS and NN require a separate training phase to build the model and, during this phase, use all training data, while the *eSensor* starts "from scratch" and uses each time the current data sample only plus the accumulated parameters β and χ^j [see (13)]. DENFIS also needs initialization and cannot start "from scratch" [31]. In addition, it is also noniterative. The fuzzy models that have automatically been extracted by the *eSensor* from the data streams are transparent and understandable by the operator of the process, yet they are robust and flexible. That means that the fuzzy-rule base that is extracted can be stored or directly presented to the operators without post-processing.

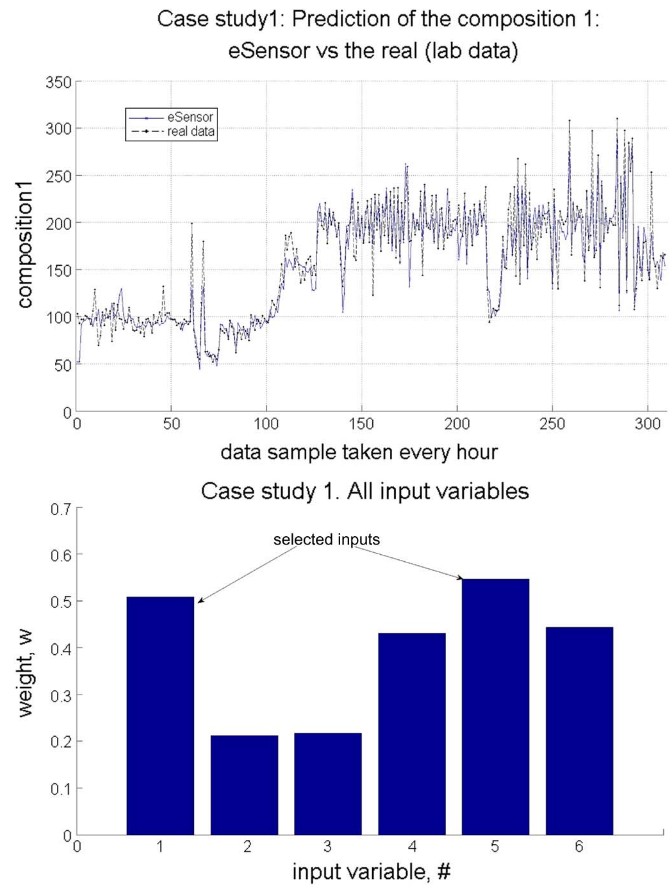

Fig. 9. Case study 1. (a) Top plot—prediction of composition 1 by the *eSensor* compared to the real data taken by laboratory samples every hour. (b) Bottom plot—selected input variables by the *eSensor*.

As seen from Table I, the *eSensor* significantly outperforms conventional inferential sensors, such as feedforward MLP and PLS-based approaches, as well as the adaptive DENFIS ones, in terms of precision. It also has significantly smaller number of rules as compared to DENFIS. The predicted versus the real (laboratory or chromatography) data are shown for all case studies in Figs. 9–12 in the top plots, together with input-variable selection in the bottom plots in Figs. 9–12.

One can see in Fig. 14 the local regions generated in another experiment (Composition 1), which are represented by dashed lines.

Additionally, the *eSensor* builds its entire structure, including input-variable online ranking and selection, fuzzy-rule generation, and self-recalibration, and is easily interpretable (linguistic). One example of the fuzzy-rule base generated automatically at the end of the training phase is given in the following for Case 2:

Final Rule Base for Composition 2:

R_1: *IF* (x_1 *is around 183.85*) *AND* (x_2 *is around 170.31*), *THEN* ($\overline{y} = 0.84 - 0.96\overline{x}_1 + 0.61\overline{x}_2$).

R_2: *IF* (x_1 *is around 178.09*) *AND* (x_2 *is around 166.84*), *THEN* ($\overline{y} = 0.87 - 0.98\overline{x}_1 + 0.54\overline{x}_2$).

R_3: *IF* (x_1 *is around 172.70*) *AND* (x_2 *is around 166.01*), *THEN* ($\overline{y} = 0.87 - 1.02\overline{x}_1 + 0.64\overline{x}_2$).

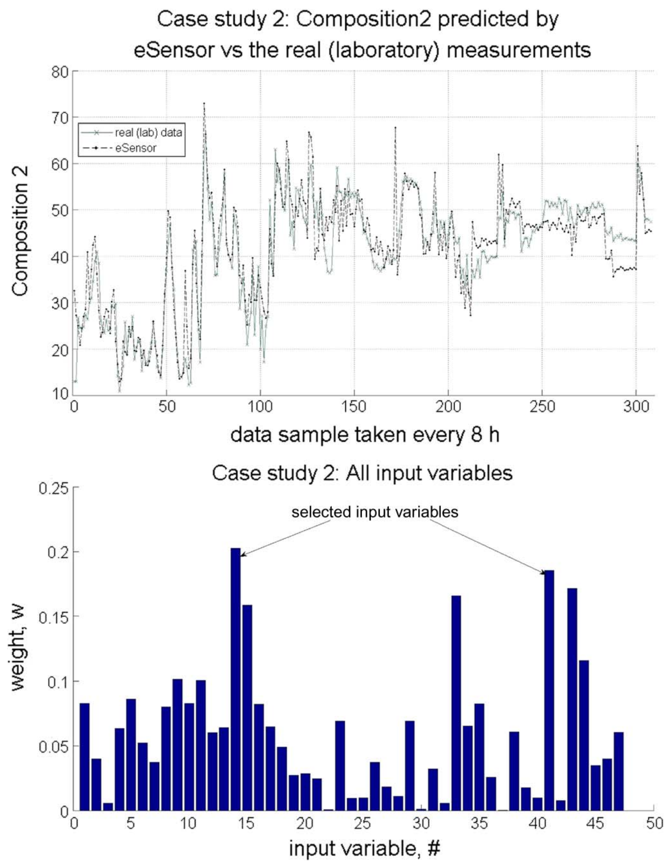

Fig. 10. Case study 2. (a) Top plot—prediction of composition 2 by the *eSensor* compared to the real data taken by laboratory samples every 8 h. (b) Bottom plot—selected input variables by the *eSensor*.

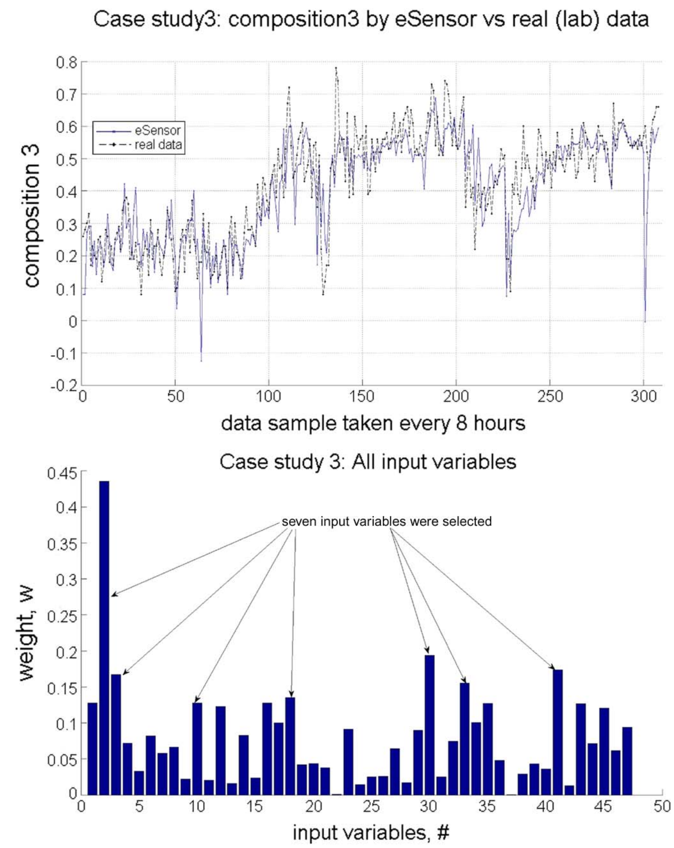

Fig. 11. Case study 3. (a) Top plot—prediction of composition 3 by the *eSensor* compared to the real data taken by laboratory samples every 8 h. (b) Bottom plot—selected input variables by the *eSensor*.

The interpretability of the fuzzy rules can be seen in Fig. 13, where the membership functions of the fuzzy sets that describe propylene polymerization are depicted. This illustrates for the input variable x_1 for the constant second input x_2 the rate with which the particular input (feature) affects the output in each of the local regions. Linear dependences are understandable for the human operators, and it is obvious from Fig. 13 that there are several linear dependences that are active for the values of x_1 (for example) around 25 and 40.

During the evolution of the rule base, the *age* of the clusters/rules is being monitored. Fig. 1(b) shows the *age* evolution of three rules from the rule base for propylene. Rule 1 is used extensively around sample 1400, and its *age* drops significantly around the same sample. At the same time, the *age rate* (first derivative of the *age*) for rule 4 is positive and increasing, which means that this particular fuzzy rule is getting older (*aging*). Such changes indicate that there is a *drift* in the data pattern, and *age rate* provides a mathematical tool to detect this automatically. A similar case occurs at around sample 2650, when a second significant *drift* is observed. Rule 3 is rarely used after its generation since its *age rate* is close to one during the whole process. This rule has been later removed automatically from the rule base.

VI. Conclusion

A new type of adaptive, self-calibrating, and self-developing inferential sensor that is based on the EFM of Takagi–Sugeno type (ETS) has been introduced in this paper and investigated on a range of case studies from the chemical and process industries. The proposed *eSensors* can be trained "on fly" starting either "from scratch" or being primed with an initial rule base. The results with data from real chemical processes demonstrate that the proposed adaptive and evolving inferential sensor is very flexible (it develops its model structure and adapts to sudden changes automatically, such as the introduced change of operating condition after sample 127 for *Composition 1* and after sample 113 for *Composition 2*). It does not need any pretraining and specific maintenance and thus reduces the life-cycle costs significantly. The structure of the proposed *eSensor* is transparent because it is composed of linguistic fuzzy rules that can be understood by an operator. The proposed evolving inferential sensor is also very robust. An illustration of this for the example of *Composition 3* was provided. Finally, due to the recursive calculations, the proposed technique is computationally very light (the computational complexity is on the order of $O(n \times R)$, where n is the number of inputs (in studied cases 2 or 7) and R is the number of fuzzy rules generated (usually a small number due to the very conservative requirement for generating new rules based on the data density (15); in the studied cases, the number of fuzzy rules generated was between two and six). It is important to note that the proposed *eSensor* is suitable for a range of process industries, including, but not limited to, chemical, biotechnology, oil refining, etc.

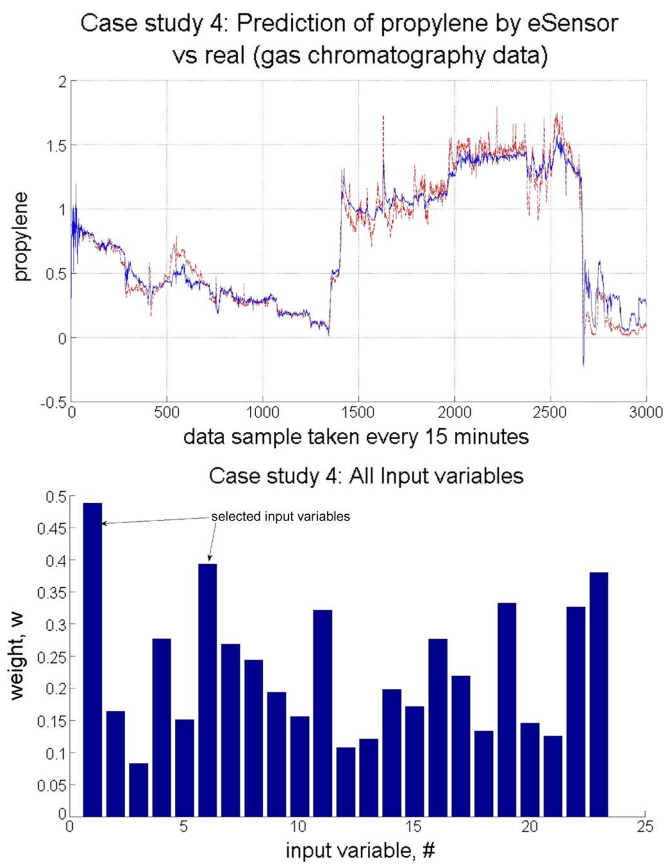

Fig. 12. Case study 4. (a) Top plot—prediction of propylene by the *eSensor* compared to the real data taken by the gas-chromatography test every 15 min. (b) Bottom plot—selected input variables by the *eSensor*.

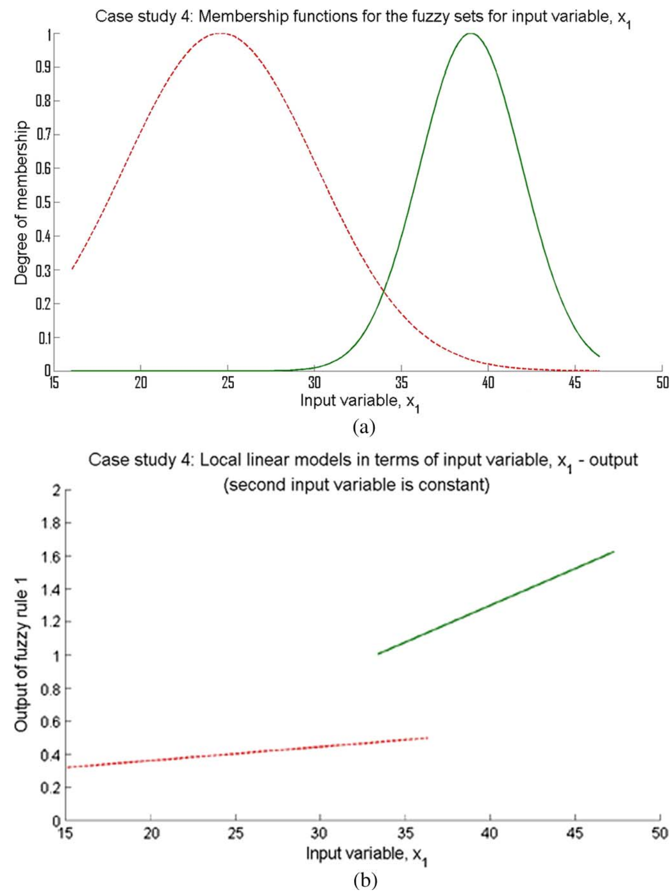

Fig. 13. (a) Membership functions of two of the fuzzy sets that form the antecedent part of the fuzzy rules of the *eSensor* at the end of the training for case study 4 (propylene). (b) Local linear models that form the consequent part of the fuzzy rules of the *eSensor* at the end of the training.

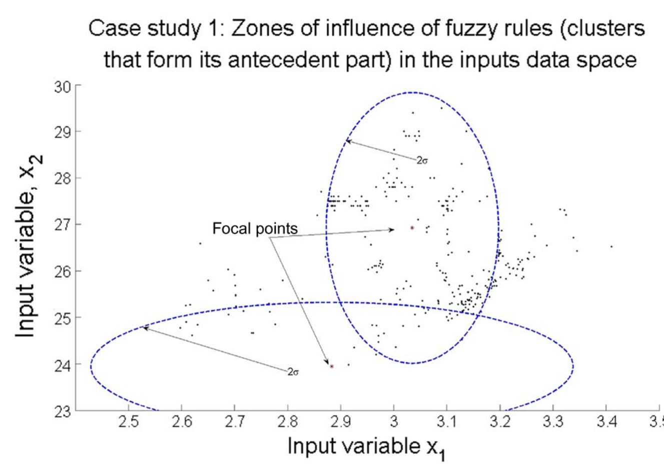

Fig. 14. Clusters that form the antecedent part of the fuzzy rules and illustrate the local areas of validity of the rules.

APPENDIX

Algorithm: *eSensor*
Begin eSensor
Initialize *eSensor* by the first data sample, $z_1 = [x_1, y_1]$; $(D_1)_1 \leftarrow 1$
(or by *iniSensor* if it exists)
DO for each data sample **WHILE** data are acquired
Read the measurable (by hard sensors) variables, x_k;
Calculate the membership to each of the fuzzy sets by (4);
Calculate the rule firing strength by (6) and (7);
Estimate the outputs, $\hat{y}_k$ by (1);
At the **next time step** $(k \leftarrow k + 1)$
IF (mode = 'self-calibration')
Get the **real** value of the estimated variables, y_k;
Calculate the density of the data sample, $D_k(z_k)$ by (13);
Update the density of the existing focal points, $D_k(z^{i*})$, by (14);
IF (15) holds **THEN**
Add a new focal point based on the new data point, (16);
Initiate its density to one, (17);
Update spreads of membership functions by (5);
IF (18) holds **THEN** Remove the rules for which it holds;
ELSE (IF (15) holds)
Ignore (do not change the cluster structure);
Update spreads of membership functions by (5);
Update the *age* of the clusters by (10);
Update the input weights by (25)
Remove the old rules (rules for which (11) holds);
Remove the inputs with low weight (26).
END (IF THEN ELSE)
Update the consequent parameters by (19) and (20).
END (self-calibration)
END (DO . . . WHILE)
END *(eSensor)*

Acknowledgment

The authors would like to thank Dr. X. Zhou for performing the numerical experiments and producing some of the figures.

References

[1] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*. London, U.K.: Springer-Verlag, 2007.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Heidelberg, Germany: Springer-Verlag, 2001.

[3] R. Rallo, J. Ferre-Gine, A. Arenas, and F. Giralt, "Neural virtual sensor for the inferential prediction of product quality from process variables," *Comput. Chem. Eng.*, vol. 26, no. 12, pp. 1735–1754, Dec. 2002.

[4] R. Klinkenberg and T. Joachims, "Detection concept drift with support vector machines," in *Proc. 7th ICML*, 2000, pp. 487–494.

[5] A. Kordon and G. Smits, "Soft sensor development using genetic programming," in *Proc. GECCO*, San Francisco, CA, 2001, pp. 1346–1351.

[6] F. Hopner and F. Klawonn, "Obtaining interpretable fuzzy models from fuzzy clustering and fuzzy regression," in *Proc. 4th Int. Conf. Knowl.-Based Intell. Eng. Syst.*, Brighton, U.K., 2000, pp. 162–165.

[7] P. Angelov, *Evolving Rule-Based Models: A Tool for Design of Flexible Adaptive Systems*. Berlin, Germany: Springer-Verlag, 2002.

[8] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, Apr. 1996.

[9] P. Angelov and D. Filev, "An approach to on-line identification of evolving Takagi–Sugeno models," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 484–498, Feb. 2004.

[10] P. Angelov and X. Zhou, "Evolving fuzzy systems from data streams in real-time," in *Proc. Int. Symp. Evolving Fuzzy Syst.*, Ambleside, U.K., Sep. 7–9, 2006, pp. 29–35.

[11] J. Macias, P. Angelov, and X. Zhou, "Predicting quality of the crude oil distillation using evolving Takagi–Sugeno fuzzy models," in *Proc. Int. Symp. Evolving Fuzzy Syst.*, Ambleside, U.K., Sep. 7–9, 2006, pp. 201–207.

[12] P. Angelov and X. Zhou, "On line learning fuzzy rule-based system structure from data streams," in *Proc. IEEE Int. Conf. Fuzzy Syst. Within IEEE World Congr. Comput. Intell.*, Hong Kong, Jun. 1–6, 2008, pp. 915–922.

[13] A. Kordon, G. Smits, A. Kalos, and E. Jordaan, "Robust soft sensor development using GP," in *Nature-Inspired Methods in Chemometrics*, R. Leardi, Ed. Amsterdam, The Netherlands: Elsevier, 2003, pp. 69–108.

[14] P. Angelov and R. Buswell, "Identification of evolving rule-based models," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 5, pp. 667–677, Oct. 2002.

[15] K. J. Astroem and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1989.

[16] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.

[17] L.-X. Wang, "Fuzzy systems are universal approximators," in *Proc. FUZZ-IEEE*, San Diego, CA, 1992, pp. 1163–1170.

[18] R. Yager and D. Filev, *Essentials of Fuzzy Modelling and Control*. New York: Wiley, 1994.

[19] P. Angelov and D. Filev, "Simpl_eTS: A simplified method for learning evolving Takagi–Sugeno fuzzy models," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Reno, NV, May 22–25, 2005, pp. 1068–1073.

[20] P. Domingos and G. Hulten, "Catching up with the data: Research issues in mining data streams," in *Proc. Workshop Res. Issues Data Mining Knowl. Discov.*, Santa Barbara, CA, 2001.

[21] E. Y. Nagai and L. V. R. Arruda, "Soft sensor based on fuzzy model identification," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, 2005.

[22] W. Yan, H. Shao, and X. Wang, "Soft sensing modeling based on support vector machine and Bayesian model selection," *Comput. Chem. Eng.*, vol. 28, no. 8, pp. 1489–1498, Jul. 2004.

[23] J. Liu, "On-line soft sensor for polyethylene process with multiple production grades," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, 2005.

[24] P. Angelov and M. Everett, "EvoMap: Evolving mapping—Implementation of evolving clustering on FPGA," Lancaster Univ., Lancaster, U.K., 2005.

[25] X. Zhou and P. Angelov, "An approach to autonomous self-localization of a mobile robot in completely unknown environment using evolving fuzzy rule-based classifier," in *Proc. 1st IEEE Int. Conf. Comput. Intell. Appl. Defense Security*, Honolulu, HI, Apr. 1–5, 2007, pp. 131–138.

[26] A. Ferreyra and J. J. Rubio, "A new on-line self-constructing neural fuzzy network," in *Proc. 45th IEEE Conf. Decision Control*, Dec. 13–15, 2006, pp. 3003–3009.

[27] P. S. Lee and A. L. Dexter, "A fuzzy sensor for measuring the mixed air temperature in air handling units," *Measurement*, vol. 37, no. 1, pp. 83–93, Jan. 2005.

[28] M. J. Arauzo-Bravo, J. M. Cano-Izquierdo, E. Gomez-Sanchez, M. J. Lopez-Nieto, Y. A. Dimitriadis, and J. Lopez-Coronado, "Automatization of a penicillin production process with soft sensors and an adaptive controller based on neuro-fuzzy systems," *Control Eng. Pract.*, vol. 12, no. 9, pp. 1073–1090, Sep. 2004.

[29] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends Cogn. Sci.*, vol. 3, no. 4, pp. 128–135, Apr. 1999.

[30] J. Kelly, P. Angelov, M. J. Walsh, H. M. Pollock, M. A. Pitt, P. L. Martin-Hirsch, and F. Martin, "A self-learning fuzzy classifier with feature selection for intelligent interrogation of mid-IR spectroscopy data derived from different categories of exfoliative cervical cytology," *Int. J. Comput. Intell. Res.—Special Issue on the Future of Fuzzy Systems Research*, vol. 4, no. 4, pp. 392–401, Dec. 2008, invited paper.

[31] N. K. 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.

Plamen Angelov (M'99–SM'04) received the M.Eng. degree in electronics and automation from Sofia Technical University, Sofia, Bulgaria, in 1989 and the Ph.D. degree in optimal control from the Bulgarian Academy of Sciences, Sofia, in 1993.

He spent over ten years as a Research Fellow working on computational intelligence and control. During 1995–1996, he was with Hans-Knoell Institute, Jena, Germany. In 1997, he was a Visiting Researcher with the Catholic University of Leuven, Leuven, Belgium. From 1998–2003, he was with Loughborough University, U.K. In 2003, he was appointed as a Lecturer with Lancaster University, U.K. In 2007, he was a Visiting Professor with University of Applied Sciences Braunschweig/Wolfenbüttel, Germany. He is currently a Senior Lecturer (Associate Professor) with Lancaster University, Lancaster, U.K., where he is with the Intelligent Systems Research Laboratory. He has authored or coauthored over 100 peer-reviewed publications, including the book entitled *Evolving Rule Based Models: A Tool for Design of Flexible Adaptive Systems* (Springer-Verlag, 2002), and over 30 journal papers, and he is a holder of a patent in machine learning (2006). He is co-Editor-in-Chief of the Evolving Systems journal (Springer) and a Member of the editorial boards of three other international scientific journals.

Dr. Angelov is a member of the Autonomous Systems Working Group of the Northwest Science Council of the U.K. and the Technical Committee (TC) on Fuzzy Systems, the Vice Chair of the TC on Standards, Computational Intelligence Society, and the founding Chair of the Task Force on Adaptive Fuzzy Systems. He serves regularly on the technical/program committees of leading international conferences on different aspects of computational intelligence.

Arthur Kordon (M'91) received the M.Eng. degree from Varna Technical University, Varna, Bulgaria, in 1974 and the Ph.D. degree from Sofia Technical University, Sofia, Bulgaria, in 1990, both in electrical engineering.

He spent more than 30 years in different positions in the industry and academe, developing and applying different methods of advanced control and computational intelligence. He is currently a Data Mining and Modeling Leader with the Data Mining and Modeling Group, The Dow Chemical Company, Freeport, TX. He is an internationally recognized expert in applying computational intelligence technologies in the industry. He has successfully introduced several novel technologies for improved manufacturing and new product design, such as robust inferential sensors, automated operating discipline, and accelerated fundamental model building. He has published more than 60 peer-reviewed papers, including the book *Applying Computational Intelligence: How to Create Value* (Springer-Verlag, 2009) and nine book chapters in the area of applied computational intelligence and advanced control. His research interests include application issues of computational intelligence, robust empirical modeling, intelligent process monitoring and control, and data mining.

Dr. Kordon is a member of the Technical Committee on Evolutionary Computation. He regularly serves on the program committees of leading international conferences related to computational intelligence.

Adaptive Inferential Sensors Based on Evolving Fuzzy Models

Plamen Angelov, *Senior Member, IEEE*, and Arthur Kordon, *Member, IEEE*

***Abstract*—A new technique to the design and use of inferential sensors in the process industry is proposed in this paper, which is based on the recently introduced concept of evolving fuzzy models (EFMs). They address the challenge that the modern process industry faces today, namely, to develop such adaptive and self-calibrating online inferential sensors that reduce the maintenance costs while keeping the high precision and interpretability/transparency. The proposed new methodology makes possible inferential sensors to recalibrate automatically, which reduces significantly the life-cycle efforts for their maintenance. This is achieved by the adaptive and flexible open-structure EFM used. The novelty of this paper lies in the following: 1) the overall concept of inferential sensors with evolving and self-developing structure from the data streams); 2) the new methodology for online automatic selection of input variables that are most relevant for the prediction; 3) the technique to detect automatically a *shift* in the data pattern using the *age* of the clusters (and fuzzy rules); 4) the online standardization technique used by the learning procedure of the evolving model; and 5) the application of this innovative approach to several real-life industrial processes from the chemical industry (evolving inferential sensors, namely, *eSensors*, were used for predicting the chemical properties of different products in The Dow Chemical Company, Freeport, TX). It should be noted, however, that the methodology and conclusions of this paper are valid for the broader area of chemical and process industries in general. The results demonstrate that well-interpretable and with-simple-structure inferential sensors can automatically be designed from the data stream in real time, which predict various process variables of interest. The proposed approach can be used as a basis for the development of a new generation of adaptive and evolving inferential sensors that can address the challenges of the modern advanced process industry.**

***Index Terms*—Concept shift in data streams, evolving fuzzy systems, fuzzy-rule aging, inferential sensors, learning and adaptation, Takagi–Sugeno (TS) fuzzy models.**

I. Introduction

Inferential sensors [1], [21], [23], [27], [28] are able to provide accurate real-time estimates of difficult-to-measure parameters or expensive measurements (like emissions, biomass, melt index, etc.) from the available cheap sensors (like temperatures, pressures, and flows). Different empirical methods have been used to develop inferential sensors, such as statistical models [2], neural networks (NNs) [3], support-vector machines [4], [22], and genetic programming [5], [13]. Model-based techniques for process-quality monitoring [1] often provide a valuable advantage over conventional approaches that rely on manual intervention and laboratory tests. Such models, however, are costly to build and maintain since the environment in which an industrial process takes place is dynamically changing, the equipment is getting older and contaminated or being replaced, raw materials usually alter in quality, and the complexity of processes leads to a number of aspects of the process being ignored by the models. A crucial weakness of model-based approaches is that they do not take into account the *shift* and *drift* in the data pattern that is related to the fact that these models are developed offline under certain conditions. Even minor process changes outside these conditions may lead to unacceptable performance deterioration that requires manual maintenance and recalibration.

The challenge is to develop inferential sensors with flexible yet interpretable structure [6] and adaptive parameters. The *gradual* evolution of the model structure (fuzzy rules) will mean that a retraining of the sensor when required will only modify (add, remove, or replace) one or few fuzzy rules [7]. Contrast this to a possible option of iteratively retraining an NN, which, in effect, will lead to a completely new NN and a loss of previous information [29]. Ideally, we would require inferential sensors that can automatically recalibrate and detect *shifts* and *drifts* in the data stream [4], [8]. One such methodological framework is presented by the evolving Takagi–Sugeno (ETS) fuzzy models [9], [10]. In this paper, we use this framework and build upon it a methodological concept for evolving inferential sensors, namely, *eSensors*, which is new and original. The main contributions of this paper include the following: 1) the overall concept of *eSensors*; 2) the new methodology for online automatic selection of input variables that are most relevant for the prediction; 3) the technique to detect automatically a *shift* in the data pattern using the *age* of the clusters (and fuzzy rules); 4) the online standardization technique used by the learning procedure of the evolving model; and 5) the application of this innovative approach to four real-life industrial processes from the chemical industries.

II. Adaptive Inferential Sensors Based on EFM

A. Principles of EFM

Evolving fuzzy models (EFMs) were first introduced as a technique for online adaptation of fuzzy-rule-based systems'

Manuscript received October 17, 2008; revised March 27, 2009 and June 26, 2009. This paper was recommended by Associate Editor T. H. Lee.

P. Angelov is with the Intelligent Systems Research Laboratory, Infolab21, Lancaster University, LA1 4WA Lancaster, U.K. (e-mail: p.angelov@lancaster.ac.uk).

A. Kordon is with The Dow Chemical Company, Freeport, TX 77541 USA (e-mail: AKKordon@Dow.com).

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/TSMCB.2009.2028315

structure (rule-based fuzzy sets), as well as their parameters [7], [14]. In that respect, they make a step further by comparing the aforementioned technique to the well-established adaptive-system theory [15], which is applicable to linear systems only and to a small circle of nonlinear systems. EFM systems are nonlinear, linguistically interpretable, yet adaptable online in a (local) least squares (LS) sense. The approach was further refined for the specific case of the so-called TS fuzzy models [16] by introducing a fully recursive algorithm called ETS [9], [10]. ETS fuzzy models are particularly suited as a framework for addressing the challenges that the process industry faces nowadays. They can provide the algorithmic backbone of systems that can be implemented as embedded autonomous intelligent sensors with self-calibration and self-maintenance capabilities.

The basic idea of ETS is to allow the TS fuzzy system structure to grow, shrink, adapt, and self-develop in an automatic fashion learned online from the data streams in a locally optimal way. TS fuzzy systems [16] are very attractive due to their dual nature—they combine the fuzzy linguistic antecedent part with a linear functional consequent part, thus being locally linear but nonlinear overall and being proven universal approximators [17]. The antecedent part is a linguistic representation of a partition of the measurable-variable space into fuzzily overlapping regions (see Fig. 14). The linguistic antecedent parts of TS fuzzy systems make them attractive for human operators (compared to NN, SVM, or polynomial models, for example). The architecture of an ETS fuzzy system is based on fuzzily weighted local linear models of the following form [9], [10]:

$$LM^i : y^i = \overline{x}^T \Theta \tag{1}$$

where LM^i denotes the ith local model, $i = 1, 2, \ldots, N$; $\overline{x} = [1, x_1, x_2, \ldots, x_n]^{\mathrm{T}}$ represents the $(n+1) \times 1$ extended vector of measurable variables; $y^i = [y_1^i, y_2^i, \ldots, y_m^i]^{\mathrm{T}}$ is the $m \times 1$ vector of estimated variables; and $\Theta^i = [\theta_0^i \quad \theta_1^i \quad \cdots \quad \theta_n^i]^{\mathrm{T}}$ denotes the matrix of consequent parameters.

All of the N local linear models describe the process in a local area defined by fuzzy rules and are blended in a fuzzy way to produce the overall output that is nonlinear in terms of measurable variables x's but is linear in terms of parameters Θ's

$$y = \psi^T \Theta \tag{2}$$

where $\psi = [\lambda^1 \overline{x}^T, \lambda^2 \overline{x}^T, \ldots, \lambda^N \overline{x}^T]^{\mathrm{T}}$ is a vector of measurable variables that are weighted by the normalized activation levels of the rules, λ^i, $i = 1, 2, \ldots, N$, with λ^i being the normalized firing level of the ith fuzzy rule that is a function of x, i.e., $\lambda^i(x)$.

The overall TS fuzzy model can then be described by a set of fuzzy rules of the following form:

$$\begin{aligned} R^i &: \text{IF } (x_1 \text{ is around } x_1^{i*}) \text{ AND}, \ldots \\ &\qquad \text{AND } \left(x_n \text{ is around } x_n^{i*}\right), \text{ THEN } (y^i = LM^i) \end{aligned} \tag{3}$$

where R^i denotes the ith fuzzy rule, with $i = [1, N]$; N is the number of fuzzy rules; $(x_j$ is around $x_j^{i*})$ denotes the jth fuzzy set of the ith fuzzy rule, with $j = 1, 2, \ldots, n$; and x^{i*} is the focal point of the ith-rule antecedent part.

The degree of membership of a certain data point (x) to any of the fuzzy rules can be described by a Gaussian centered at its focal point

$$\mu^i = e^{-\frac{\sum_{j=1}^n \left(x_j - x_j^{i*}\right)^2}{2\left(\sigma_j^i\right)^2}} \tag{4}$$

having a spread that is learned based on the data variance [10]

$$\begin{aligned} \left(v_{jk}^i\right)^2 &= \rho \left(v_{j(k-1)}^i\right)^2 + (1-\rho) \frac{1}{n_k^i} \sum_{l=1}^{n_k^i} \|z^{i*} - z_l\|_j^2, \\ v_{j1}^i &= 1, \quad \sigma_{jk}^i \leftarrow v_{jk}^i \end{aligned} \tag{5}$$

where v_{jk}^i denotes the variance of the data in the ith cluster in the jth dimension (jth variable) calculated at the kth time instant, σ_{jk}^i represents the spread of the Gaussian of the jth fuzzy set of the ith fuzzy rule calculated at the kth time instant, $z = [x, y]^{\mathrm{T}}$ depicts the overall data vector, and n_k^i denotes the *support* of the ith cluster/rule—the number of samples that are associated with it based on the distance to the focal point.

The firing strength of a fuzzy rule is determined by a *t-norm*, which can be represented as inner product [18]

$$\tau^i = \prod_{j=1}^n \mu_j^i(x_j) \tag{6}$$

and is normalized so that it sums to one

$$\lambda^i = \frac{\tau^i}{\sum_{j=1}^N \tau_j}. \tag{7}$$

B. *Monitoring the Quality of the Rule Base*

One can monitor and analyze online the quality of the clusters that are formed and the fuzzy rules, respectively—for example, the number of points that support them or their *age* [19]. The support of the rules is determined by a simple counting of the samples that are associated with the *nearest* focal point

$$n_{k+1}^i = n_k^i + 1, \quad i = \arg \min_{i=1}^N \|x_k - x^{i*}\|, \quad k = 2, 3, \ldots. \tag{8}$$

The support is initiated by one at the moment a rule is created

$$n_k^{N+1} \leftarrow 1, \quad k = 2, 3, \ldots. \tag{9}$$

In this paper, we introduce a recursive formula to calculate the *age* of the ith cluster/rule calculated at the kth moment in time (data sample)

$$A_k^i = k - \frac{1}{n_k^i} \left(k - A_{k-1}^i + k_{n_k^i}\right) \tag{10}$$

where k_l is the time index when the data sample was read.

This follows from

$$A_k^i = k - \frac{1}{n_k^i} \sum_{l=1}^{n_k^i} k_l \quad A_{k-1}^i = k - \frac{1}{n_{k-1}^i} \sum_{l=1}^{n_{k-1}^i} k_l.$$

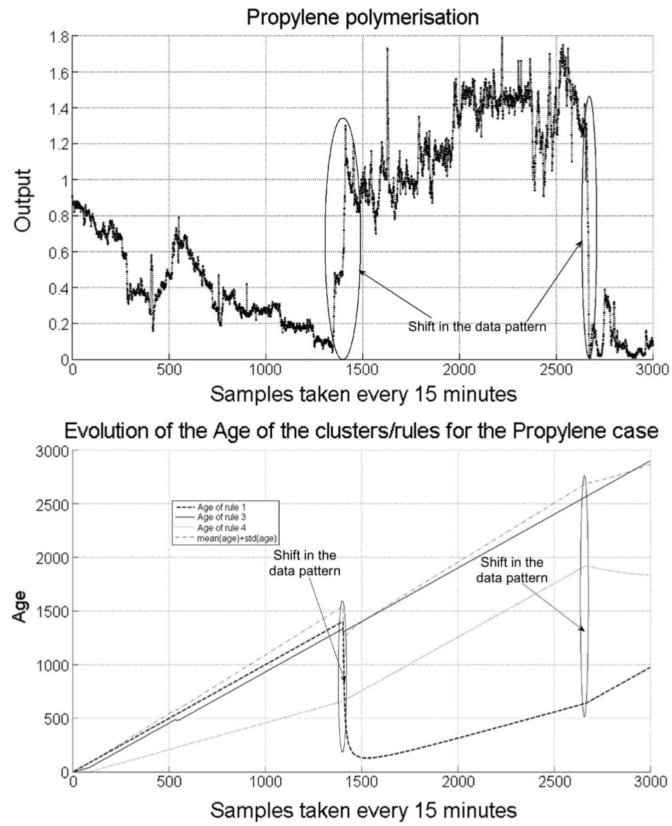

Fig. 1. (a) Top plot—output variable in case study 4—polymerization; (b) Bottom plot—*age* of the fuzzy rules describing the propylene-polymerization process. The two instants when a *shift* in the data pattern occurs are marked. This corresponds to a change in the *aging* rate seen from the bottom plot.

From there, we get

$$\sum_{l=1}^{n_{k-1}^i} k_l = \left(k - A_{k-1}^i\right) \quad A_k^i = k - \frac{1}{n_k^i}\left(\sum_{l=1}^{n_{k-1}^i} k_l + k_{n_k^i}\right).$$

Combining these two expressions, we arrive at (10).

Each time a new rule is created, its *age* is initiated by the index of the data sample that is used as a focal point of that rule. Each time a new data sample is associated to an existing rule (the distance from a sample to that focal point is smaller than that to any other focal points), the *age* of that rule gets smaller. If no sample is assigned to a rule, it gets older by one. Note that the *age* of a fuzzy rule can take values from the $[0; k]$ range. This is shown in Fig. 1 in the case of propylene estimation. From the top plot, one can see that there are three different stages of that process. The *aging* of three of the six fuzzy rules (rules ## 1, 3, and 4) are depicted in the bottom plot. One can see that precisely at the moment of a *shift* in the data pattern (a new phase), the *aging* of the rules is affected. By monitoring the derivative of A (i.e., *aging rate*), one can automatically detect such changes and respond by adapting the learning mechanism or rate.

Note that the *age rate* of rule #1 becomes ***negative*** before it increases again. This illustrates the so-called concept *shift* and is an indication of a transition from one operating state (which affects the data density in one local region, i.e., around the focal point of this rule) to another one (which affects the data density in another local region).

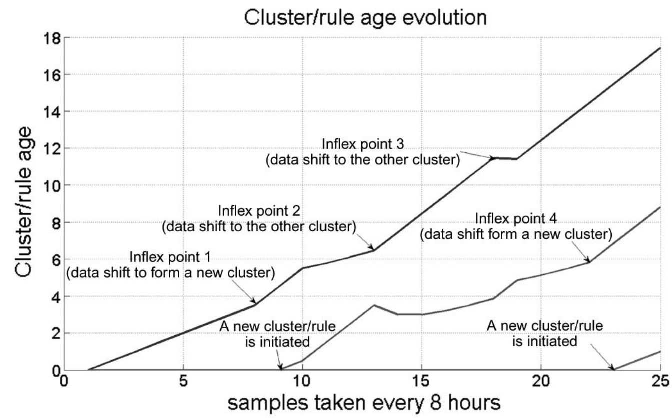

Fig. 2. Evolution of the *age* and *shift* in the data pattern, resulting in forming new clusters/rules for case study 2. The inflex points correspond to a shift of the data from one cluster to another existing cluster or to a newly formed cluster (as marked in the figure for each inflex point).

The *age* of the fuzzy rules (and the derivative of their *age* in terms of the sampling period (k), which represents the *aging rate*) can be very useful for online analysis of the concept *shift* in the data stream [12]. An *eSensor* can detect a concept *shift* [20] online by the rate of *aging* and the instances when it changes [the inflex points on the *age* evolution diagram that corresponds to the change of the sign of the *aging* rate indicate a *shift* (see Fig. 2)]. The aging corresponds to the first derivative of the *age* and is graphically represented by the slope of the age evolution lines in terms of the horizontal axis [see Fig. 1(b)].

In this paper, we use the following principle for the update of the rule base by removing the *older* rules (rules whose *age* exceeds the mean *age* for that rule by more than the standard-deviation [2] value calculated recursively up to that moment/sample):

$$\text{IF}\Big(A^i > \overline{A^i} + \text{std}(A^i)\Big), \text{THEN}\left(\text{remove } R^i; N \leftarrow N-1\right) \quad (11)$$

where $\overline{A^i}$ denotes the mean *age* (it is also denoted in Fig. 1(b) by a dash-dotted line) and $\text{std}(A^i)$ represents the standard deviation of the *age* of the ith rule.

C. Evolving the Structure of the Sensor From the Data Stream

The online design and learning of the *eSensor* are outlined here. Learning is based on decomposition of the identification problem into the following [7], [9], [10]: 1) fuzzy-rule-based structure design and 2) parameter identification. Both of these subproblems can be performed in online mode during one time step (per sample). The first subproblem, i.e., structure identification, can be approached using evolving clustering in the data space [9], [10], [12]. This partitioning leads to forming information granules, described linguistically by fuzzy sets. Thus, it serves the transformation of the data into primitive forms of knowledge. The basic notion of the partitioning algorithm is that of the data *density* [26], which is defined as a Cauchy function over the sum of distances d's between a certain data sample z_i and ***all*** other data samples in the feature space [10]

$$D_k(z_k) = \frac{1}{1 + \overline{v}_k^2} \quad (12)$$

where $\overline{v}_k^2 = (1/k-1)\sum_{i=1}^{k-1} d^2(z_k, z_i)$ is the variance of the data [2].

Data-space partitioning is based on the following principle: *The point with the highest density in the data space is chosen to be the focal point, and the antecedent of the first fuzzy rule is formed around it*. In this way, fuzzy rules with high descriptive power and generalization capabilities are generated. The density can be ***recursively*** calculated using the current data point (z_k^j) and $(n+1)$ memorized quantities only (β_k and χ_k^j, $j=[1,n]$) [10]

$$D_k(z_k) = (k-1)\left(\alpha_k(k-1) + \beta_k - 2\gamma_k + (k-1)\right)^{-1}, \qquad k = 2, 3, \ldots \quad (13)$$

where $\alpha_k = \sum_{j=1}^{n+1}(z_k^j)^2$; $\beta_k = \beta_{k-1} + \alpha_{k-1}$, with $\beta_1 = 0$; and $\gamma_k = \sum_{j=1}^{n+1} z_k^j \chi_k^j$, with $\chi_k^j = \chi_{k-1}^j + z_{k-1}^j$ and $\chi_1^j = 0$.

Each time a new data sample is read, it affects the data density of the existing focal points and can be updated by [10]

$$D_k(z^{i*}) = \frac{(k-1)D_{k-1}(z^{i*})}{k-2+D_{k-1}(z^{i*}) + D_{k-1}(z^{i*})d(z^{i*}, z_k)}, \qquad k = 2, 3, \ldots \quad (14)$$

where $d(z^{i*}, z_k)$ denotes the distance between the ith focal point and the current point.

Once the densities of the new coming data sample and of each of the previously existing focal points are recursively updated, they are compared. If the new coming data sample has a higher density than ***any*** of the previously existing focal points, then this means that it is a good candidate to become a focal point of a new rule (a new local linear model) because it has high descriptive power and generalization potential

$$D_k(z_k) > D_k(z^{i*}) \quad \forall i^* \in N. \quad (15a)$$

If the new coming data sample has a lower density than ***any*** of the previously existing focal points, then this means that it is also a good candidate to become a focal point of a new rule (a new local linear model) because it improves the coverage of the whole data space [12]

$$D_k(z_k) < D_k(z^{i*}) \quad \forall i^* \in N. \quad (15b)$$

Forming a new fuzzy rule around a newly added prototype leads to a *gradual* increase of the size of the rule base, which is why this approach is called "evolving"

$$z^{(N+1)*} \leftarrow z_k. \quad (16)$$

The density of the newly generated rule is set to one [10] temporarily (it will be updated to take into account later the influence of each new coming data sample on the generalization potential of this particular focal point)

$$D_k\left(z^{(N+1)*}\right) \leftarrow 1. \quad (17)$$

To increase the interpretability and update of the rule base, one needs also to remove the previously existing rules that become ambiguous after insertion of the new rule. Therefore, each time a new fuzzy rule is added, it is also checked whether any of the already existing prototypes in the rule base are described by this rule to a degree that is higher than 50%

$$\exists i, \quad i=[1,N]; \quad \mu_i^j(z^{N+1}) > 0.5 \quad \forall j, \quad j=[1,n]. \quad (18)$$

If any of the previously existing focal points satisfy this condition, the rules that correspond to them are being removed (replaced by the newly formed rule) [9], [19]. The spreads of the membership functions are also recursively updated by (5).

D. Self-Learning the eSensor

Once the antecedent part of the TS fuzzy model is formed, the consequent-parameter estimation (the second subproblem of the learning) is addressed as a fuzzily weighted recursive LS (RLS) estimation problem per rule [15]

$$\Theta_k^i = \Theta_{k-1}^i + C_k^i \lambda^i \overline{x}_k \left(y_k - \overline{x}_k^T \Theta_{k-1}^i\right), \quad \Theta_1^i = 0 \quad (19)$$

$$C_k^i = C_{k-1}^i - \frac{\lambda^i C_{k-1}^i \overline{x}_k \overline{x}_k^T C_{k-1}^i}{1 + \lambda^i \overline{x}_k^T C_{k-1}^i \overline{x}_k}, \quad C_1^i = \Omega I, \quad k = 2, 3, \ldots \quad (20)$$

where $C \in R^{N(n+L)xN(n+L)}$ denotes the covariance matrix, Ω is a large positive number, and I is the identity matrix.

As a result, the *eSensor* blends in a fuzzy way local linear predictors. Moreover, it is optimally (in an LS sense) [15] tuned in terms of consequent parameters Θ's. In terms of its antecedents and rule-based structure, it is based on the robust online partitioning approach. The procedure of the *eSensor* self-development and self-calibration is represented as a pseudo-code in the Appendix.

E. Online Normalization and Standardization of the Data in the eSensor

One specific issue related to this online algorithm is the normalization or standardization of the data. Both normalization and standardization are well-established techniques for the offline case when all the data are available [2]. An approach to update the normalization ranges of the data in a recursive manner is presented in [25], but in this paper, we use the recursive version of the standardization technique that can easily be inferred from the offline version [2] because it depends on the mean and variance of the data only. Let us remember that (offline) standardization is given by [2]

$$Z_{jk} = \frac{z_{jk} - \overline{z}_{jk}}{\zeta_{jk}}, \quad j=[1,n], \quad k = 2, 3, \ldots \quad (21)$$

where Z_{jk} denotes the standardized value of z_{jk}; $\overline{z}_{jk} = (1/k)\sum_{l=1}^{k} z_{jl}$, $j=[1,n]$, $k = 2, 3, \ldots$, represents the mean value of z_{jk}; and v_{jk} is the standard deviation of the jth input calculated based on k data samples.

Both the mean and standard deviation can be updated recursively

$$\begin{aligned}\overline{z}_{jk} &= \frac{k-1}{k}\overline{z}_{j(k-1)} + \frac{1}{k}z_{j(k-1)},\\ \overline{z}_{j1} &= 0, \quad j=[1,n+m], \quad k=2,3,\ldots \end{aligned} \tag{22a}$$

$$\begin{aligned}v_{jk}^2 &= \frac{k-1}{k}v_{j(k-1)}^2 + \frac{1}{k-1}\left(z_{jk} - \overline{z}_{j(k-1)}\right),\\ v_{j1} &= 0, \quad j=[1,n+m], \quad k=2,3,\ldots. \end{aligned} \tag{22b}$$

In order to return to the original scale, one should apply destandardization by

$$z_{jk} = Z_{jk}\nu_{jk} + \overline{z}_{jk}, \quad j=[1,n+m], \quad k=2,3,\ldots. \tag{23}$$

III. Online Input-Variable Selection in the eSensor

Inferential sensors, as well as other online models, traditionally assume the number of input variables to be known beforehand or to be preselected. In what follows, we propose an original[1] method to online "on-fly" ranking and selection of input variables, which was successfully approbated on the industrial case studies reported in this paper, as well as on other real applications [30]. The importance of this technique should not be underestimated because, very often in practice, there are large sets of candidate variables that may influence the monitored or measured output, but often, it is not clear how much. The idea is based on online ranking of the accumulated values formed by the consequent parameters Θ_{jk}^i, $j=[1,N]$,$i=[1,R]$.

The accumulated values π's indicate that the weight of a particular consequent parameter is determined by simply adding the absolute values (because the consequent parameters are unrestricted in sign and value, and their contribution is judged by the modulus)

$$\pi_{jk}^i = \sum_{l=1}^{k}\left|\Theta_{jl}^i\right|, \quad j=[1,n], \quad i=[1,R]. \tag{24}$$

One can also form a weight of a particular feature by the ratio of π values

$$\omega_{jk}^i = \frac{\pi_{jk}^i}{\sum_{r=1}^{n}\pi_{jk}^i}, \quad i=[1,R], \quad j=[1,n]. \tag{25}$$

It is important to note that (24) and (25) represent sums only and are thus easily performed online. The values of the weights ω's indicate the contribution of a particular input to the overall output and are thus a measure of the sensitivity of the outputs. Therefore, an intuitive technique to simplify the inferential sensor structure in terms of inputs can be proposed, which gradually removes the input variables for which the weight ω is negligibly small across the rules (i.e., the inputs that contribute little to the overall output)

$$\text{IF}\left(\exists j^*\middle|\omega_{j^*k}^i < \varepsilon \max_{j=1}^{n}\pi_{jk}^i\right), \text{ THEN (remove } j^*) \tag{26}$$

[1]This technique is part of a pending patent: P. Angelov, Machine Learning (Collaborative Systems), WO2008053161, priority date: November 1, 2006; intern. filing date: October 23, 2007; http://v3.espacenet.com/textdoc?DB=EPODOC&IDX=WO2008053161&F=0&QPN=WO2008053161

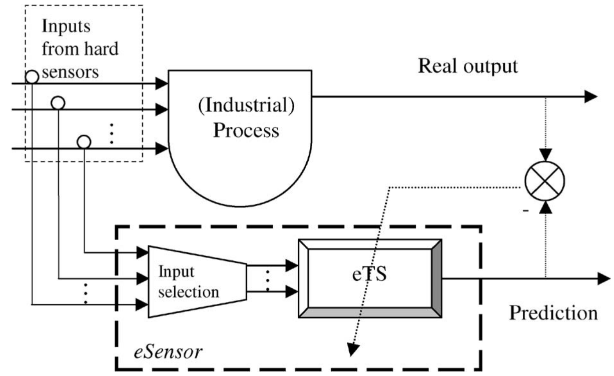

Fig. 3. Overall schematic representation of the *eSensor*.

where ε denotes a coefficient (the suggested values are [0.03; 0.1], which means that this input variable contributes 3%–10% to the overall output on average.

The rationale for the simplicity of this technique stems from the fact that the consequents represent locally linear combinations and can thus be analyzed. It should be noted that, when an input is removed (which does not usually occur very often), however, the dimension is reduced by one, which is reflected in the covariance matrices (a line and a column are removed), and the dimensions of the focal points are also updated, as well as the recursive variables in (13), i.e., α, β, γ, and χ.

The main advantages of the proposed *eSensor* approach that makes it suitable for implementation in the process industry are as follows.

1) It self-develops, *evolves*, and thus reduces the development and maintenance costs significantly.
2) It can provide high prediction rates.
3) It is one-pass and recursive and has low computational requirements; thus, it is suitable for hardware "on-chip" implementations [24].
4) It is useful for online analysis and monitoring of the concept *shift* using fuzzy-rule *aging* [see Figs. 1(b) and 2] and thus makes useful conclusions for possible faults and the quality of the process.
5) It can automatically select online a small subset of relevant inputs, thus fully automating the development process.
6) It can have a *multiple-input–multiple-output* structure and thus build a separate regression model for each output variable.

The procedure for adaptive and *evolving* inferential self-calibrating sensors, which we call *eSensor*, is presented by the pseudocode provided in the Appendix (see also Fig. 3).

IV. Case Study: Inferential Sensors for Chemical-Property Estimation

The capabilities of the proposed evolving inferential sensor are explored on four different industrial data sets for chemical-property estimation. All four cases include operating-regime changes with different impacts on specific chemical properties due to different levels of process change, various measurement methods with different accuracies, and a different number of potential process variables, related to the inferred chemical properties. However, all the changes create a challenge to

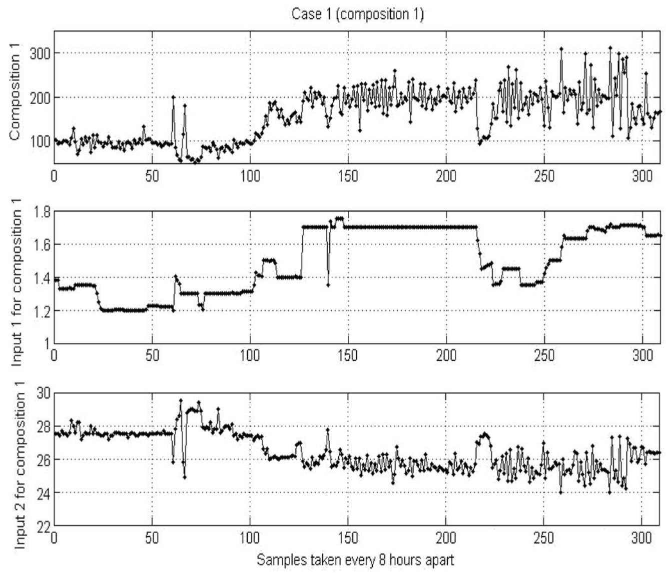

Fig. 4. Case study 1: Composition 1. Top plot—output variable (composition 1). Middle plot—input variable (x_1). Bottom plot—input variable (x_2).

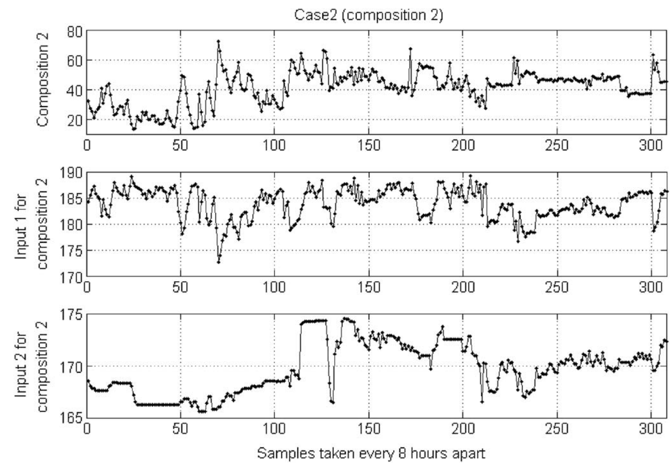

Fig. 5. Input and output variables for case study 2. Top plot—output variable (composition 2). Middle plot—input variable (x_1). Bottom plot—input variable (x_2).

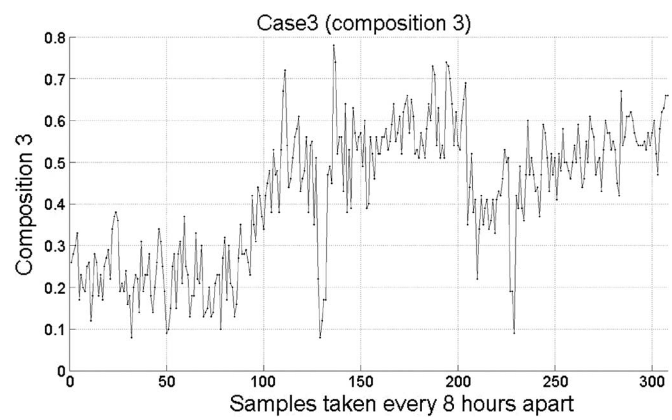

Fig. 6. Output variable for case study 3 (composition 3). There are seven selected inputs, and they are not shown for clarity purposes.

existing inferential sensors with a fixed structure. As a basis for comparison, inferential sensors based on the most widely used methods in commercial soft-sensor products, such as the feedforward NN of multilayer perceptron (MLP) type [3] and PLS [1], were used, as well as a recently introduced algorithm for adaptive online NN, namely, DENFIS [31].

In the chemical industry, inferential sensors are mostly used to estimate chemical properties, measured by two techniques: 1) offline laboratory analysis of grab samples of the properties and 2) pseudo real-time analysis with low frequencies by gas chromatographs. The sampling period for the properties, measured by laboratory analysis, is several hours, and accuracy depends on different measurement methods and varies substantially. The sampling period of gas-chromatograph-based properties is much shorter (usually 15–30 min), and accuracy is, on average, an order of magnitude higher than that from offline laboratory measurements. Three of the selected cases are based on offline laboratory measurements, and one is based on gas chromatographs. In the cases with laboratory measurements, two different levels of accuracies have been selected. The level of operating-condition change (which could be quantified by the percentage increase from the average level for 50 samples before the process change to the average level for 50 samples after the change), as well as the number of process inputs, is also different.

The first case, called Composition 1, is based on product-composition estimation in a distillation tower. The measurements are based on laboratory analysis, taken every 8 h, and the method accuracy is low (2.2% measurement error), which, by itself, introduced a measurement noise. Process data are the hourly averaged values around the time when the sample for the laboratory measurement has been taken. The output composition and the two-input data (Fig. 4) include 309 records (samples). As it is seen in the middle plot in Fig. 4, a significant change in operating conditions has been introduced after sample 127 by input 1. It is interesting to note that the two input variables that were selected online using the *eSensor* are the most statistically significant process variables related to this composition.

The second case, called Composition 2, is based on product-composition estimation *in the bottom* of a distillation tower, which is different from the tower in Composition 1. The composition measurements are based on laboratory analysis, taken every 8 h with a more accurate method of 1.3% measurement error, and are less noisy. Process data are the hourly averaged values for the time when the sample for the laboratory measurement has been taken. The output composition and the two-input data (Fig. 5) include 308 records (samples), where a significant change in operating conditions has been introduced after sample 113 by input 2. Forty-seven different input variables were measured using "*hard*" (conventional) sensors.

The third case, called Composition 3, is based on product-composition estimation *in the top* of the same distillation tower as that in Composition 2. The output composition is shown in Fig. 6, and it also includes 308 data samples with a significant change in operating conditions (catalyzing agent replacement) introduced after sample 113. The key differences of Case 3 relative to the other laboratory-measurement-based cases are as follows: 1) higher level of operating-condition changes (275% increase versus 220% increase for Case 1 and 232% increase

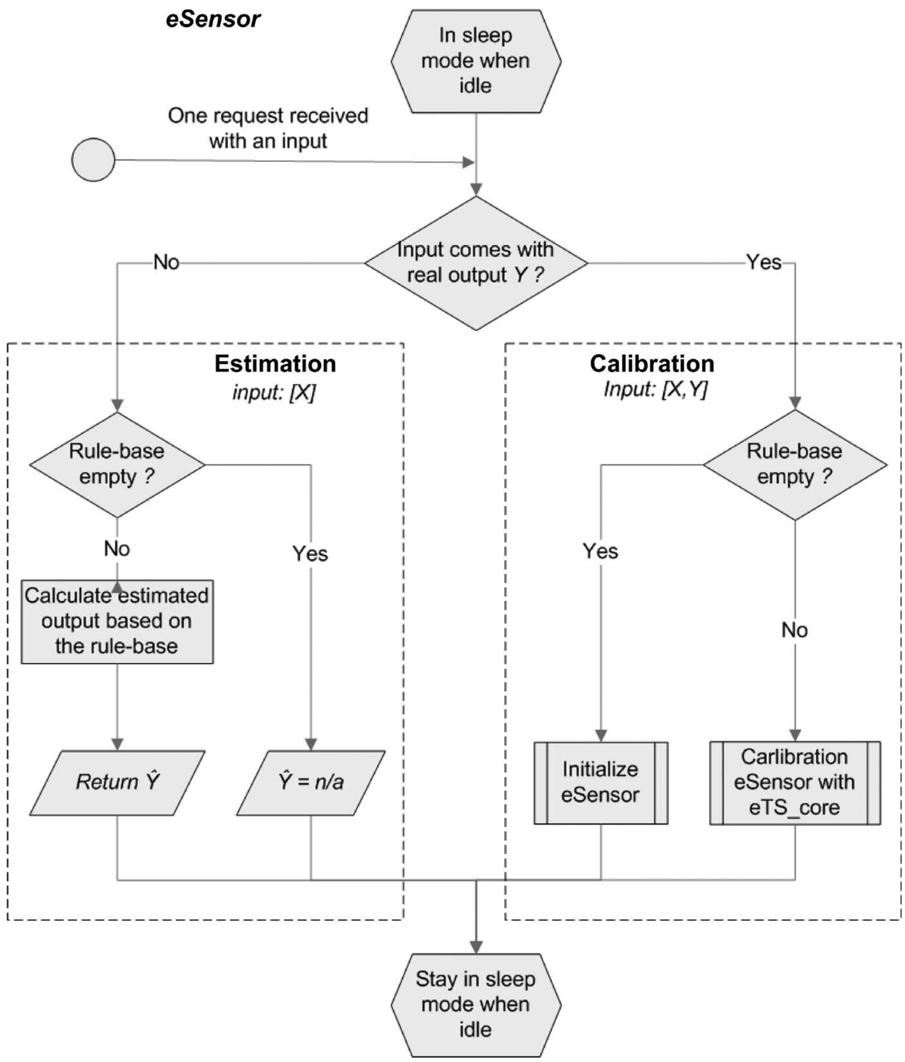

Fig. 7. Flowchart of the eSensor from the real-time software-realization point of view. Sleep mode means a default state expecting an external request. Note that all the stages of eSensor self-calibration are combined in one block on the right bottom part of the flowchart. This includes learning, the online input selection, as well as cluster/rule removal based on their *age*. The details of this procedure are provided in the Appendix.

for Case 2) and 2) larger number of process inputs (seven inputs versus two inputs for both Cases 1 and 2).

The fourth case is based on *propylene* estimation in the *top of a distillation tower*, which is different from the distillation towers in the previous cases. In this case 2, process variables that are related to propylene are used as inputs in the model development. The propylene measurements are based on gas-chromatograph analysis, taken every 15 min. Process data are the snapshot minute values for the time when the measurement has been taken. The data [Fig. 1(a)] include 3000 records (samples) with very broad range of operating conditions.

These four test cases (provided and used by The Dow Chemical Company, Freeport, TX) cover most of the real issues in applying inferential sensors in the advanced process industry, such as noisy data, changing operating conditions, a large number of correlated inputs, etc.

V. Experimental Results and Analysis

The main aim of the experimental study was to generate interpretable simple-to-understand models that are flexible and adaptive (evolving with time and following the dynamics of the data pattern) and are robust to noise and imprecise measurement data using the proposed technique *eSensor* and to compare these results with the available alternatives based on MLP-type NN, PLS, and a recently introduced evolving NN, i.e., DENFIS [31]. Precision was measured using root mean square errors (RMSE), as well as correlation [2]. The data in all experiments were standardized. The *eSensor* starts with an empty fuzzy-rule base (*no iniSensor*) and generates its rule-base "on fly" based on the data that are provided sample by sample and disregarded from the memory once processed. It also optimizes the parameters during retraining periods (it self-calibrates). The output prediction is calculated for every data sample and can be used at any time instant. Samples for recalibration are provided when they are available (see Fig. 7). DENFIS was also applied in an online mode.

The conventional inferential sensors (PLS and NN) that are not adaptive were trained initially using the first quarter of the available data samples, and afterward, they were retrained using samples from the third quarter of the available data stream. The error was only calculated on the second and fourth quarters of

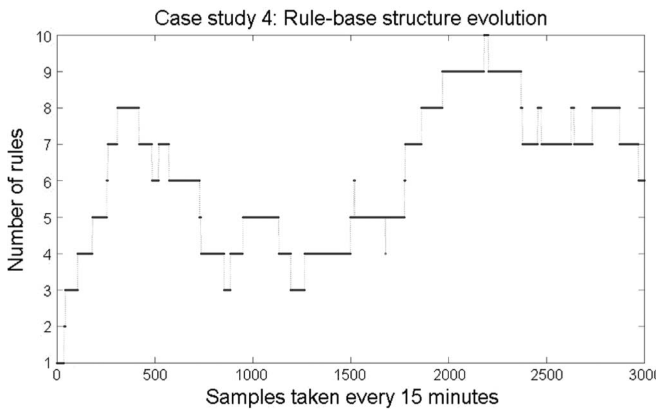

Fig. 8. Evolution of the rule base of the *eSensor* (number of rules vary starting from one—the first sample—finishing with six, and reaching at some stage 10).

TABLE I
VALIDATION RESULT USING ESENSOR AND REFERENCE APPROACHES

Case study		**1**	**2**	**3**	**4**
Total # of input variables		6	47	47	22
Total data samples		309	308	308	3000
Sampling period, *h*		1	8	8	1/4
eSensor* *(this paper)	# inputs	**2**	**2**	**7**	**2**
	RMSE	**18.533**	**2.3658**	**0.0716**	**0.0698**
	Correlation	**0.9477**	**0.9788**	**0.9026**	**0.9901**
	# rules	**4**	**3**	**4**	**6**
eTS [9,10] using all inputs	RMSE	18.918	3.786	0.075	0.116
	Correlation	0.884	0.834	0.782	0.976
	# rules	4	4	5	6
Feed-forward MLP	RMSE	23.12	2.87	0.098	0.628
	Correlation	0.890	0.91	0.802	0.85
PLS	RMSE	24.39	2.79	0.093	0.194
	Correlation	0.881	0.90	0.817	0.901
DENFIS [31] all inputs	RMSE	19.106	33.52	0.399	0.110
	# rules	19	32	32	235

the data stream in all cases (PLS, NN, and eSensor) to allow compatibility of the results. Note that the *eSensor* can also be retrained anytime when a training sample is available, and moreover, its structure (rule based) will be preserved and only gradually adapted/evolved.

The evolution of the fuzzy rule base is shown in Fig. 8, where the number of fuzzy rules generated is shown for the fourth case study (propylene). In retraining the NN and PLS, the parameters (weights) change completely and are not interpretable. Note that both PLS and NN require a separate training phase to build the model and, during this phase, use all training data, while the *eSensor* starts "from scratch" and uses each time the current data sample only plus the accumulated parameters β and χ^j [see (13)]. DENFIS also needs initialization and cannot start "from scratch" [31]. In addition, it is also noniterative. The fuzzy models that have automatically been extracted by the *eSensor* from the data streams are transparent and understandable by the operator of the process, yet they are robust and flexible. That means that the fuzzy-rule base that is extracted can be stored or directly presented to the operators without post-processing.

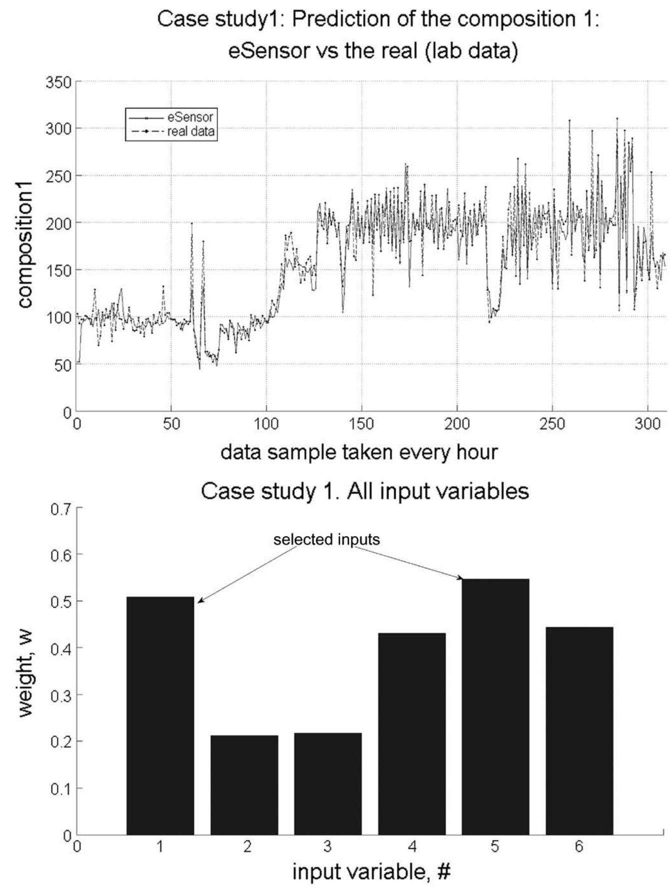

Fig. 9. Case study 1. (a) Top plot—prediction of composition 1 by the *eSensor* compared to the real data taken by laboratory samples every hour. (b) Bottom plot—selected input variables by the *eSensor*.

As seen from Table I, the *eSensor* significantly outperforms conventional inferential sensors, such as feedforward MLP and PLS-based approaches, as well as the adaptive DENFIS ones, in terms of precision. It also has significantly smaller number of rules as compared to DENFIS. The predicted versus the real (laboratory or chromatography) data are shown for all case studies in Figs. 9–12 in the top plots, together with input-variable selection in the bottom plots in Figs. 9–12.

One can see in Fig. 14 the local regions generated in another experiment (Composition 1), which are represented by dashed lines.

Additionally, the *eSensor* builds its entire structure, including input-variable online ranking and selection, fuzzy-rule generation, and self-recalibration, and is easily interpretable (linguistic). One example of the fuzzy-rule base generated automatically at the end of the training phase is given in the following for Case 2:

Final Rule Base for Composition 2:

R_1: *IF* (x_1 *is around 183.85*) *AND* (x_2 *is around 170.31*), *THEN* ($\overline{y} = 0.84 - 0.96\overline{x}_1 + 0.61\overline{x}_2$).

R_2: *IF* (x_1 *is around 178.09*) *AND* (x_2 *is around 166.84*), *THEN* ($\overline{y} = 0.87 - 0.98\overline{x}_1 + 0.54\overline{x}_2$).

R_3: *IF* (x_1 *is around 172.70*) *AND* (x_2 *is around 166.01*), *THEN* ($\overline{y} = 0.87 - 1.02\overline{x}_1 + 0.64\overline{x}_2$).

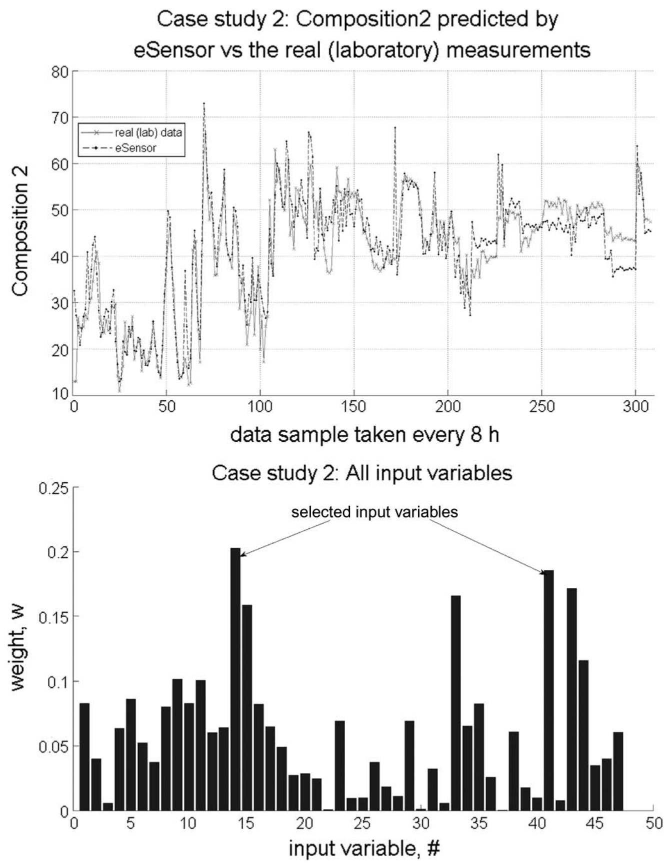

Fig. 10. Case study 2. (a) Top plot—prediction of composition 2 by the *eSensor* compared to the real data taken by laboratory samples every 8 h. (b) Bottom plot—selected input variables by the *eSensor*.

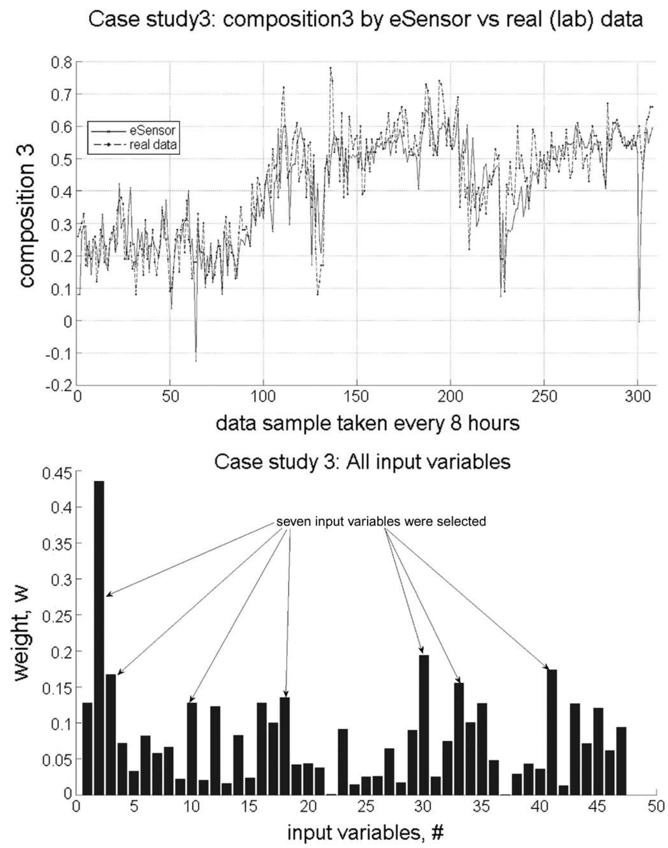

Fig. 11. Case study 3. (a) Top plot—prediction of composition 3 by the *eSensor* compared to the real data taken by laboratory samples every 8 h. (b) Bottom plot—selected input variables by the *eSensor*.

The interpretability of the fuzzy rules can be seen in Fig. 13, where the membership functions of the fuzzy sets that describe propylene polymerization are depicted. This illustrates for the input variable x_1 for the constant second input x_2 the rate with which the particular input (feature) affects the output in each of the local regions. Linear dependences are understandable for the human operators, and it is obvious from Fig. 13 that there are several linear dependences that are active for the values of x_1 (for example) around 25 and 40.

During the evolution of the rule base, the *age* of the clusters/rules is being monitored. Fig. 1(b) shows the *age* evolution of three rules from the rule base for propylene. Rule 1 is used extensively around sample 1400, and its *age* drops significantly around the same sample. At the same time, the *age rate* (first derivative of the *age*) for rule 4 is positive and increasing, which means that this particular fuzzy rule is getting older (*aging*). Such changes indicate that there is a *drift* in the data pattern, and *age rate* provides a mathematical tool to detect this automatically. A similar case occurs at around sample 2650, when a second significant *drift* is observed. Rule 3 is rarely used after its generation since its *age rate* is close to one during the whole process. This rule has been later removed automatically from the rule base.

VI. Conclusion

A new type of adaptive, self-calibrating, and self-developing inferential sensor that is based on the EFM of Takagi–Sugeno type (ETS) has been introduced in this paper and investigated on a range of case studies from the chemical and process industries. The proposed *eSensors* can be trained "on fly" starting either "from scratch" or being primed with an initial rule base. The results with data from real chemical processes demonstrate that the proposed adaptive and evolving inferential sensor is very flexible (it develops its model structure and adapts to sudden changes automatically, such as the introduced change of operating condition after sample 127 for *Composition 1* and after sample 113 for *Composition 2*). It does not need any pretraining and specific maintenance and thus reduces the life-cycle costs significantly. The structure of the proposed *eSensor* is transparent because it is composed of linguistic fuzzy rules that can be understood by an operator. The proposed evolving inferential sensor is also very robust. An illustration of this for the example of *Composition 3* was provided. Finally, due to the recursive calculations, the proposed technique is computationally very light (the computational complexity is on the order of $O(n \times R)$, where n is the number of inputs (in studied cases 2 or 7) and R is the number of fuzzy rules generated (usually a small number due to the very conservative requirement for generating new rules based on the data density (15); in the studied cases, the number of fuzzy rules generated was between two and six). It is important to note that the proposed *eSensor* is suitable for a range of process industries, including, but not limited to, chemical, biotechnology, oil refining, etc.

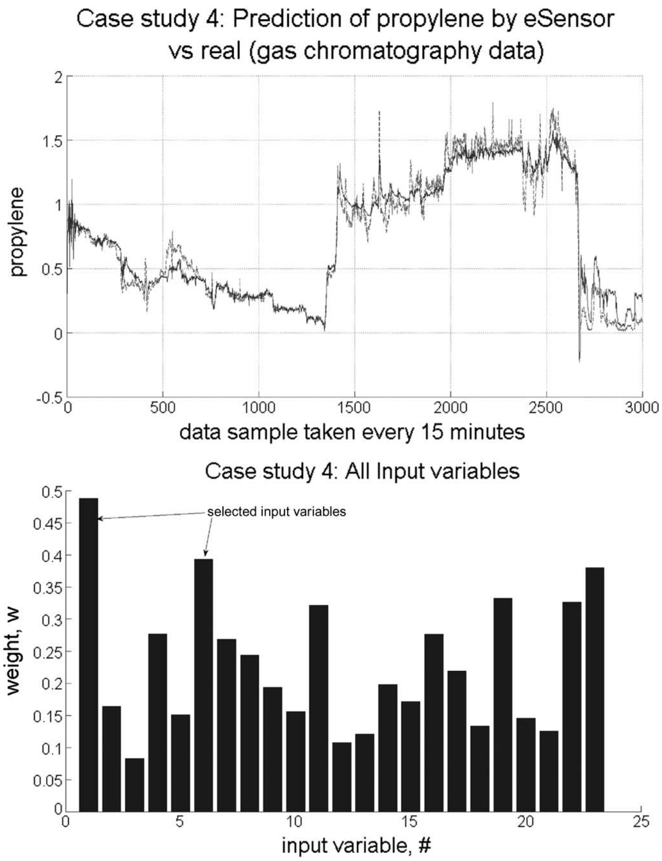

Fig. 12. Case study 4. (a) Top plot—prediction of propylene by the *eSensor* compared to the real data taken by the gas-chromatography test every 15 min. (b) Bottom plot—selected input variables by the *eSensor*.

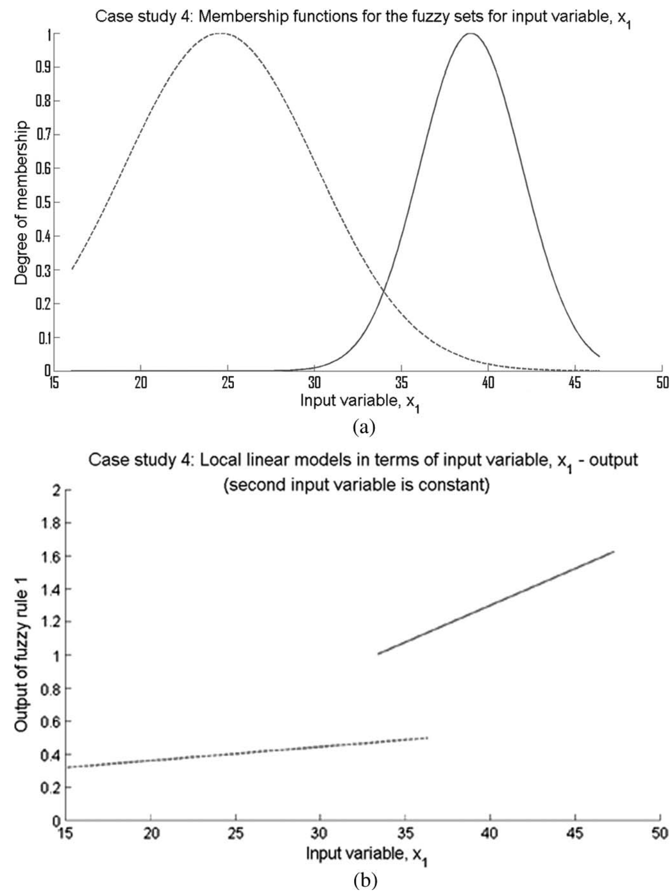

Fig. 13. (a) Membership functions of two of the fuzzy sets that form the antecedent part of the fuzzy rules of the *eSensor* at the end of the training for case study 4 (propylene). (b) Local linear models that form the consequent part of the fuzzy rules of the *eSensor* at the end of the training.

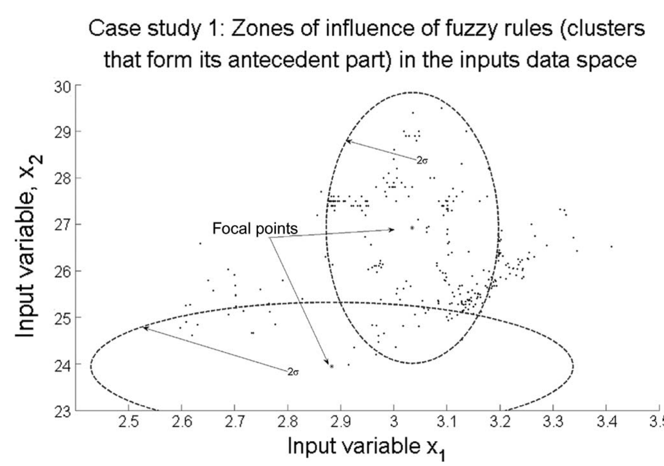

Fig. 14. Clusters that form the antecedent part of the fuzzy rules and illustrate the local areas of validity of the rules.

APPENDIX

Algorithm: *eSensor*
Begin eSensor
Initialize *eSensor* by the first data sample, $z_1 = [x_1, y_1]$; $(D_1)_1 \leftarrow 1$
(or by *iniSensor* if it exists)
DO for each data sample **WHILE** data are acquired
Read the measurable (by hard sensors) variables, x_k;
Calculate the membership to each of the fuzzy sets by (4);
Calculate the rule firing strength by (6) and (7);
Estimate the outputs, $\hat{y}_k$ by (1);
At the **next time step** $(k \leftarrow k+1)$
IF (mode = 'self-calibration')
Get the **real** value of the estimated variables, y_k;
Calculate the density of the data sample, $D_k(z_k)$ by (13);
Update the density of the existing focal points, $D_k(z^{i*})$, by (14);
IF (15) holds **THEN**
Add a new focal point based on the new data point, (16);
Initiate its density to one, (17);
Update spreads of membership functions by (5);
IF (18) holds **THEN** Remove the rules for which it holds;
ELSE (IF (15) holds)
Ignore (do not change the cluster structure);
Update spreads of membership functions by (5);
Update the *age* of the clusters by (10);
Update the input weights by (25)
Remove the old rules (rules for which (11) holds);
Remove the inputs with low weight (26).
END (IF THEN ELSE)
Update the consequent parameters by (19) and (20).
END (self-calibration)
END (DO . . . WHILE)
END *(eSensor)*

ACKNOWLEDGMENT

The authors would like to thank Dr. X. Zhou for performing the numerical experiments and producing some of the figures.

REFERENCES

[1] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*. London, U.K.: Springer-Verlag, 2007.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Heidelberg, Germany: Springer-Verlag, 2001.

[3] R. Rallo, J. Ferre-Gine, A. Arenas, and F. Giralt, "Neural virtual sensor for the inferential prediction of product quality from process variables," *Comput. Chem. Eng.*, vol. 26, no. 12, pp. 1735–1754, Dec. 2002.

[4] R. Klinkenberg and T. Joachims, "Detection concept drift with support vector machines," in *Proc. 7th ICML*, 2000, pp. 487–494.

[5] A. Kordon and G. Smits, "Soft sensor development using genetic programming," in *Proc. GECCO*, San Francisco, CA, 2001, pp. 1346–1351.

[6] F. Hopner and F. Klawonn, "Obtaining interpretable fuzzy models from fuzzy clustering and fuzzy regression," in *Proc. 4th Int. Conf. Knowl.-Based Intell. Eng. Syst.*, Brighton, U.K., 2000, pp. 162–165.

[7] P. Angelov, *Evolving Rule-Based Models: A Tool for Design of Flexible Adaptive Systems*. Berlin, Germany: Springer-Verlag, 2002.

[8] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, Apr. 1996.

[9] P. Angelov and D. Filev, "An approach to on-line identification of evolving Takagi–Sugeno models," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 484–498, Feb. 2004.

[10] P. Angelov and X. Zhou, "Evolving fuzzy systems from data streams in real-time," in *Proc. Int. Symp. Evolving Fuzzy Syst.*, Ambleside, U.K., Sep. 7–9, 2006, pp. 29–35.

[11] J. Macias, P. Angelov, and X. Zhou, "Predicting quality of the crude oil distillation using evolving Takagi–Sugeno fuzzy models," in *Proc. Int. Symp. Evolving Fuzzy Syst.*, Ambleside, U.K., Sep. 7–9, 2006, pp. 201–207.

[12] P. Angelov and X. Zhou, "On line learning fuzzy rule-based system structure from data streams," in *Proc. IEEE Int. Conf. Fuzzy Syst. Within IEEE World Congr. Comput. Intell.*, Hong Kong, Jun. 1–6, 2008, pp. 915–922.

[13] A. Kordon, G. Smits, A. Kalos, and E. Jordaan, "Robust soft sensor development using GP," in *Nature-Inspired Methods in Chemometrics*, R. Leardi, Ed. Amsterdam, The Netherlands: Elsevier, 2003, pp. 69–108.

[14] P. Angelov and R. Buswell, "Identification of evolving rule-based models," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 5, pp. 667–677, Oct. 2002.

[15] K. J. Astroem and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1989.

[16] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.

[17] L.-X. Wang, "Fuzzy systems are universal approximators," in *Proc. FUZZ-IEEE*, San Diego, CA, 1992, pp. 1163–1170.

[18] R. Yager and D. Filev, *Essentials of Fuzzy Modelling and Control*. New York: Wiley, 1994.

[19] P. Angelov and D. Filev, "Simpl_eTS: A simplified method for learning evolving Takagi–Sugeno fuzzy models," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Reno, NV, May 22–25, 2005, pp. 1068–1073.

[20] P. Domingos and G. Hulten, "Catching up with the data: Research issues in mining data streams," in *Proc. Workshop Res. Issues Data Mining Knowl. Discov.*, Santa Barbara, CA, 2001.

[21] E. Y. Nagai and L. V. R. Arruda, "Soft sensor based on fuzzy model identification," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, 2005.

[22] W. Yan, H. Shao, and X. Wang, "Soft sensing modeling based on support vector machine and Bayesian model selection," *Comput. Chem. Eng.*, vol. 28, no. 8, pp. 1489–1498, Jul. 2004.

[23] J. Liu, "On-line soft sensor for polyethylene process with multiple production grades," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, 2005.

[24] P. Angelov and M. Everett, "EvoMap: Evolving mapping—Implementation of evolving clustering on FPGA," Lancaster Univ., Lancaster, U.K., 2005.

[25] X. Zhou and P. Angelov, "An approach to autonomous self-localization of a mobile robot in completely unknown environment using evolving fuzzy rule-based classifier," in *Proc. 1st IEEE Int. Conf. Comput. Intell. Appl. Defense Security*, Honolulu, HI, Apr. 1–5, 2007, pp. 131–138.

[26] A. Ferreyra and J. J. Rubio, "A new on-line self-constructing neural fuzzy network," in *Proc. 45th IEEE Conf. Decision Control*, Dec. 13–15, 2006, pp. 3003–3009.

[27] P. S. Lee and A. L. Dexter, "A fuzzy sensor for measuring the mixed air temperature in air handling units," *Measurement*, vol. 37, no. 1, pp. 83–93, Jan. 2005.

[28] M. J. Arauzo-Bravo, J. M. Cano-Izquierdo, E. Gomez-Sanchez, M. J. Lopez-Nieto, Y. A. Dimitriadis, and J. Lopez-Coronado, "Automatization of a penicillin production process with soft sensors and an adaptive controller based on neuro-fuzzy systems," *Control Eng. Pract.*, vol. 12, no. 9, pp. 1073–1090, Sep. 2004.

[29] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends Cogn. Sci.*, vol. 3, no. 4, pp. 128–135, Apr. 1999.

[30] J. Kelly, P. Angelov, M. J. Walsh, H. M. Pollock, M. A. Pitt, P. L. Martin-Hirsch, and F. Martin, "A self-learning fuzzy classifier with feature selection for intelligent interrogation of mid-IR spectroscopy data derived from different categories of exfoliative cervical cytology," *Int. J. Comput. Intell. Res.—Special Issue on the Future of Fuzzy Systems Research*, vol. 4, no. 4, pp. 392–401, Dec. 2008, invited paper.

[31] N. K. 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.

Plamen Angelov (M'99–SM'04) received the M.Eng. degree in electronics and automation from Sofia Technical University, Sofia, Bulgaria, in 1989 and the Ph.D. degree in optimal control from the Bulgarian Academy of Sciences, Sofia, in 1993.

He spent over ten years as a Research Fellow working on computational intelligence and control. During 1995–1996, he was with Hans-Knoell Institute, Jena, Germany. In 1997, he was a Visiting Researcher with the Catholic University of Leuven, Leuven, Belgium. From 1998–2003, he was with Loughborough University, U.K. In 2003, he was appointed as a Lecturer with Lancaster University, U.K. In 2007, he was a Visiting Professor with University of Applied Sciences Braunschweig/Wolfenbüttel, Germany. He is currently a Senior Lecturer (Associate Professor) with Lancaster University, Lancaster, U.K., where he is with the Intelligent Systems Research Laboratory. He has authored or coauthored over 100 peer-reviewed publications, including the book entitled *Evolving Rule Based Models: A Tool for Design of Flexible Adaptive Systems* (Springer-Verlag, 2002), and over 30 journal papers, and he is a holder of a patent in machine learning (2006). He is co-Editor-in-Chief of the Evolving Systems journal (Springer) and a Member of the editorial boards of three other international scientific journals.

Dr. Angelov is a member of the Autonomous Systems Working Group of the Northwest Science Council of the U.K. and the Technical Committee (TC) on Fuzzy Systems, the Vice Chair of the TC on Standards, Computational Intelligence Society, and the founding Chair of the Task Force on Adaptive Fuzzy Systems. He serves regularly on the technical/program committees of leading international conferences on different aspects of computational intelligence.

Arthur Kordon (M'91) received the M.Eng. degree from Varna Technical University, Varna, Bulgaria, in 1974 and the Ph.D. degree from Sofia Technical University, Sofia, Bulgaria, in 1990, both in electrical engineering.

He spent more than 30 years in different positions in the industry and academe, developing and applying different methods of advanced control and computational intelligence. He is currently a Data Mining and Modeling Leader with the Data Mining and Modeling Group, The Dow Chemical Company, Freeport, TX. He is an internationally recognized expert in applying computational intelligence technologies in the industry. He has successfully introduced several novel technologies for improved manufacturing and new product design, such as robust inferential sensors, automated operating discipline, and accelerated fundamental model building. He has published more than 60 peer-reviewed papers, including the book *Applying Computational Intelligence: How to Create Value* (Springer-Verlag, 2009) and nine book chapters in the area of applied computational intelligence and advanced control. His research interests include application issues of computational intelligence, robust empirical modeling, intelligent process monitoring and control, and data mining.

Dr. Kordon is a member of the Technical Committee on Evolutionary Computation. He regularly serves on the program committees of leading international conferences related to computational intelligence.